



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Robust and Data-Efficient End-to-End Speech Translation

Segmentation, Cross-Modal Alignment, and Prosody-Aware Evaluation

Ioannis Tsiamas

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya

A thesis submitted for the degree of
Doctor of Philosophy

Advisors:

José A. R. Fonollosa

Marta R. Costa-jussà

Barcelona, August 2026

Abstract

End-to-end (E2E) Speech Translation (ST) offers a streamlined alternative to traditional cascaded systems, promising lower latency and reduced error propagation. However, its adoption in real-world scenarios is currently hindered by three critical bottlenecks: input processing challenges (mismatch between static data and continuous audio), training data scarcity, and evaluation limitations regarding paralinguistic information.

This dissertation addresses these limitations through a cohesive set of methodological and architectural innovations. First, we tackle the segmentation and input processing challenge. We introduce Supervised Hybrid Audio Segmentation (SHAS), a method that effectively bridges the gap between manual training segmentation and automatic inference segmentation, allowing E2E models to process continuous audio streams with minimal performance loss. Building on this, we propose SEGAUGMENT, a data augmentation strategy that utilizes training data re-segmentation to maximize the utility of existing datasets, significantly improving performance in low-resource settings.

Second, we tackle data scarcity by developing methods for zero-shot speech translation through cross-modal alignment. We introduce ZEROSWOT, which leverages Optimal Transport to align speech encoders with massively multilingual machine translation models at the subword level, enabling translation without paired ST data. We further refine this approach with CHARSONAR, demonstrating that character-level modeling significantly improves cross-lingual and cross-modal transfer, achieving state-of-the-art results, despite being a zero-shot method.

Finally, we investigate the semantic richness of E2E translations. We present a focused evaluation on prosody, revealing that while current E2E architectures possess the internal capacity to represent paralinguistic features like intonation and stress, they often fail to manifest these in the final translation.

Collectively, these contributions advance the state of the art by creating E2E ST systems that are more robust to unsegmented inputs and more data-efficient, while also providing insights into the limitations of current E2E systems in preserving the nuances of spoken communication.

Acknowledgments

I feel incredibly grateful to my academic supervisors, Marta and Adrian. Thank you for trusting me with this opportunity, for guiding my first steps in research, and for giving me the freedom to explore my own path.

I am also deeply thankful for the experiences I gained during my time in industry. My growth as a researcher was shaped significantly by the invaluable guidance I received from my managers and colleagues: Santi and Chunghsin at Dolby; Matthias at Apple; and Marta (once again!), David, Paul-Ambroise, Alex, João, Pere-Lluís, and the rest of the Omnilingual team at Meta.

To my PhD friends, with whom I have shared the same passion for this work: thank you for the countless hours of brainstorming, the late-night debugging, and the adrenaline of rushing for conference deadlines. To Gerard, Javi, Carlos, Belen, Eduardo, Gerard S., Christine, and Casimiro: this journey would not have been the same without you.

To my partner, Maria, thank you for being my constant. I am grateful for your unconditional support and for always being there for me through the highs and lows of this process.

Finally, the most sincere thank you to my family. None of what I have accomplished would have been possible without you.

Contents

Abstract	i
Acknowledgments	ii
I Introduction and Background	2
1 Introduction	3
1.1 Motivation	3
1.2 Main Contributions	4
1.3 PhD Outcomes	6
1.3.1 Publications	6
1.3.2 Industrial Collaborations	8
1.4 Thesis Outline	9
2 Background and State of the Art	11
2.1 Deep Learning and Natural Language Processing	11
2.1.1 Neural Networks and Optimization	11
2.1.2 Sequence Modeling	12
2.1.3 Tokenization and Input Representation	13
2.2 The Transformer Architecture	14
2.2.1 Self-Attention and Multi-Head Attention	14
2.2.2 The Encoder-Decoder Structure	16
2.2.3 Positional Encodings	17
2.3 Speech Processing Foundations	18
2.3.1 Digital Signal Processing (DSP)	18
2.3.2 Prosody and Paralinguistic Features	19
2.3.3 Self-Supervised Learning (SSL) in Speech	19
2.3.4 Audio Segmentation	21
2.4 Automatic Speech Recognition (ASR)	21
2.4.1 The ASR Task	22
2.4.2 Connectionist Temporal Classification (CTC)	22
2.4.3 Encoder-Decoder ASR	24

2.5	Machine Translation and Multilingual Representations	24
2.5.1	Neural Machine Translation (NMT)	25
2.5.2	Multilingual NMT	25
2.5.3	Multilingual Embedding Spaces	26
2.6	End-to-End Speech Translation	27
2.6.1	Cascade vs. End-to-End Paradigms	27
2.6.2	Data Scarcity	29
2.6.3	The Modality Gap	30
2.6.4	Speech Translation Architectures	31
2.6.5	Evaluation Metrics	34
II	Audio Segmentation	37
3	Bridging the Segmentation Gap	38
3.1	Introduction	38
3.1.1	The Segmentation Mismatch	38
3.1.2	Contribution: Supervised Hybrid Audio Segmentation	39
3.2	The Challenge of Audio Segmentation for Translation	40
3.3	Existing Approaches and Limitations	41
3.4	The SHAS Methodology	42
3.4.1	The Segmentation Frame Classifier	42
3.4.2	The Probabilistic Divide-and-Conquer Algorithm	45
3.5	Experimental Setup	47
3.5.1	Data	48
3.5.2	Segmentation Frame Classifiers	48
3.5.3	Speech Translation Models	49
3.5.4	Baselines and Evaluation	50
3.6	Results and Analysis	51
3.6.1	Approaching the Oracle	51
3.6.2	Multilingual Generalization	52
3.6.3	Zero-Shot Cross-Lingual Transfer	52
3.6.4	Domain Robustness	54
3.6.5	Sensitivity to Segment Length	54
3.6.6	Ablations	56
3.7	Summary and Conclusions	57

4	Turning Segmentation into Augmentation	60
4.1	Introduction	60
4.2	Related Work: Augmentation and Segmentation	62
4.3	Background: The Limits of Static Segmentation	64
4.4	The SEGAUGMENT Framework	65
4.4.1	Diversifying Inputs via Re-segmentation	66
4.4.2	Transcript Recovery via Forced Alignment	67
4.4.3	Translation Recovery via Text Alignment	68
4.5	Experimental Setup	69
4.5.1	Data	69
4.5.2	Method Configuration	71
4.5.3	Speech Translation	71
4.6	Maximizing Data Utility and Generalization	72
4.6.1	Augmenting Standard Baselines	73
4.6.2	Pushing State-of-the-Art Performance	74
4.6.3	Impact in Low-Resource Scenarios	75
4.6.4	Applicability to Sentence-Level Data	75
4.6.5	Independence from ASR Pre-training	76
4.7	Solving the Training-Inference Segmentation Mismatch	77
4.8	Why it Works: Mechanisms of Improvement	79
4.8.1	Contextual Diversity	80
4.8.2	Positional Diversity	81
4.8.3	Length Specialization	81
4.8.4	Knowledge Distillation vs. Data Augmentation	82
4.9	Summary and Conclusions	84
III	Cross-Modal Alignment	85
5	Zero-Shot Speech Translation via Cross-Modal Alignment	86
5.1	Introduction	86
5.2	Related Works: Advancements and Limitations	89
5.2.1	Strategies for Data Scarcity	89
5.2.2	Addressing the Modality Gap	90
5.2.3	Advances in Zero-Shot Speech Translation	91
5.3	Problem Formulation for Zero-shot Speech Translation	92
5.3.1	The Alignment Objective	92
5.3.2	Dimensionality Constraints	93

5.4	The ZEROSWOT Methodology	93
5.4.1	Model Architecture	94
5.4.2	Optimal Transport Alignment	98
5.4.3	Combined Training Objective	101
5.5	Experimental Setup	102
5.5.1	Datasets	102
5.5.2	Model Configuration	104
5.5.3	Training and Evaluation	105
5.6	Experimental Results	106
5.6.1	Comparison with Zero-Shot Baselines	106
5.6.2	Comparison with Supervised Baselines	107
5.6.3	Comparison with Cascade Systems	110
5.6.4	Supervised ST Fine-tuning	112
5.7	Analysis: The Modality Gap	113
5.7.1	Addressing the Length Gap	113
5.7.2	Closing the Representation Gap	114
5.8	Conclusions and Next Steps	116
6	Character-level Modeling for Enhanced Modality Transfer	117
6.1	Introduction	117
6.2	Related Work	120
6.2.1	Character-level Machine Translation	120
6.2.2	Cross-modal Transfer	120
6.2.3	SONAR and Fixed-size Embeddings	121
6.3	Methodology	122
6.3.1	The Foundation: SONAR	122
6.3.2	Problem Formulation and Preliminaries	124
6.3.3	Text Adaptation: Constructing CHARSONAR	125
6.3.4	Speech Adaptation: Bridging MMS and CHARSONAR	127
6.4	Experimental Setup	131
6.4.1	Datasets	131
6.4.2	Model Architecture	132
6.4.3	Training Details	132
6.4.4	Evaluation	133
6.5	Advancing Language Transfer: Text Adaptation Results	134
6.5.1	Initial Exploration and Objective Analysis	134
6.5.2	Scaling-up Language Coverage	136
6.5.3	Analysis	137

6.6	Closing the Modality Gap: Speech Adaptation Results	139
6.6.1	Adapter Design and Robustness	139
6.6.2	Zero-shot Speech Translation	140
6.6.3	Cross-modal Similarity Search	142
6.6.4	Impact of Text Surface	143
6.7	Conclusions	144
IV	Prosody-Aware Evaluation	146
7	Towards Prosody-aware Evaluation: Speech Is More Than Words	147
7.1	Introduction	147
7.2	Related Work	150
7.2.1	Prosody in Speech Synthesis	150
7.2.2	Prosody in Speech-to-Text Translation	150
7.3	The CONTRAPROST Benchmark	151
7.3.1	Sentence Stress	152
7.3.2	Prosodic Breaks	152
7.3.3	Intonation Patterns	153
7.3.4	Emotional Prosody	153
7.3.5	Politeness	153
7.4	Data Generation Methodology	154
7.4.1	Prosodic Example Generation	154
7.4.2	Oracle Translation	155
7.4.3	Controllable Speech Synthesis	156
7.4.4	Quality Assessment and Filtering	157
7.5	Contrastive Evaluation Framework	158
7.5.1	Contrastive Likelihood	158
7.5.2	Contrastive Quality	159
7.5.3	Evaluation Conditions	160
7.6	Experimental Setup	161
7.6.1	Data Generation Statistics	161
7.6.2	Speech Translation Models	162
7.6.3	Metrics and Implementation	163
7.7	Results and Analysis	163
7.7.1	General Prosody Awareness	165
7.7.2	Architecture Analysis: E2E vs. Cascade	165
7.7.3	Comparison with Standard Metrics	167

7.7.4	Analysis by Prosodic Category	168
7.7.5	Language Dependencies	169
7.8	Conclusions	169
V	Closing	173
8	Conclusions	174
8.1	Bridging the Segmentation Gap	174
8.2	Mitigating Data Scarcity via Cross-Modal Alignment	175
8.3	Evaluating Whether Speech Adds Meaning	176
8.4	Overall Takeaways	176
9	Retrospective: Speech Translation in the Era of Multimodal LLMs	177
10	Directions for Future Work	181
	Bibliography	184
A	Supplement to “Turning Segmentation into Augmentation”	220
A.1	Pseudocode for CTC-based Forced Alignment	220
A.2	Additional Experimental Details	221
A.3	Additional Experimental Results	223
A.4	SEGAUGMENT Data Statistics	225
B	Supplement to “Zero-Shot Speech Translation via Cross-Modal Alignment”	229
B.1	Dataset Details and Filtering	229
B.2	ZEROSWOT Model Configuration	231
B.2.1	Model Architecture	231
B.2.2	Training Details	232
B.2.3	Compression Ablation Configurations	233
B.2.4	Supervised ST Fine-tuning Details	234
B.3	Machine Translation Models	234
B.4	Cascade Speech Translation Systems	235
B.5	Evaluation	237
B.6	Additional Results	237
B.6.1	Massively Multilingual Evaluation on FLEURS	237
B.6.2	Extended Results in MuST-C and CoVoST	239

C	Supplement to “Character-level Modeling for Enhanced Modality Transfer”	242
C.1	Data and Languages	242
C.2	Additional Results	244
D	Supplement to “Towards Prosody-aware Evaluation: Speech Is More Than Words”	250
D.1	Examples for In-context Learning	250
D.2	Quality Assessment for TTS Candidates	252
D.3	Data	254
D.3.1	Data Statistics	254
D.3.2	Overly Explanatory Examples	256
D.4	Evaluated Speech Translation Models	256
D.5	Additional Results	258
E	Additional Main-Author Publications	262
E.1	Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation	262
E.2	Speech Translation with Foundation Models and Optimal Transport	263
E.3	Efficient Speech Translation with Dynamic Latent Perceivers	264
E.4	Sequential Contrastive Audio-Visual Learning	265

Part I

Introduction and Background

1 Introduction

1.1 Motivation

In an increasingly interconnected global society, overcoming linguistic barriers is a prerequisite for effective communication and collaboration. Text-based Machine Translation (MT) has achieved remarkable proficiency, approaching human parity for high-resource languages (Fan et al., 2021; Liu et al., 2020a; NLLB, 2024). Nevertheless, speech remains the primary and most natural mode of human communication. Consequently, the development of robust Speech Translation (ST) systems is essential for facilitating global interaction.

End-to-end Speech Translation (E2E ST) has emerged as a promising paradigm to translate speech signals directly into foreign text (Bérard et al., 2016; Weiss et al., 2017). By replacing the traditional cascade of Automatic Speech Recognition (ASR) and Machine Translation (MT) (Ney, 1999; Mathias and Byrne, 2006), E2E models theoretically avoid error propagation, reduce latency, and retain access to the full acoustic context of the source utterance (Sperber and Paulik, 2020; Bentivogli et al., 2021). However, despite these advantages, current E2E systems still lag behind cascades in many real-world scenarios (Anastasopoulos et al., 2021, 2022), hindering the widespread adoption.

First, there is a fundamental mismatch between training and inference conditions. E2E models are typically trained on manually segmented, sentence-level audio data. In contrast, real-world applications require processing continuous, unsegmented audio streams. When models trained on static sentences are applied to continuous inputs, performance degrades significantly (Ansari et al., 2020), highlighting a critical gap in the input processing pipeline.

Second, data scarcity is the most significant barrier to scaling ST. Unlike ASR or MT, which benefit from massive monolingual corpora (Ardila et al., 2020; Yang et al., 2025) or parallel text corpora (Tiedemann and Thottingal, 2020; NLLB, 2024), paired speech-to-translation data is extremely rare and usually includes only a handful of high-resource languages (Di Gangi et al., 2019a; Wang et al., 2020a). This scarcity creates a *modality*

gap (Liu et al., 2020b) between pre-trained text representations and acoustic features. Bridging this gap without relying on large amounts of supervised ST data is essential for expanding language coverage to the long tail of low-resource languages.

Finally, while E2E models have access to the source audio (Sperber and Paulik, 2020), there is limited evidence that they actually leverage the rich paralinguistic information contained within it. Features such as prosody (stress, intonation, rhythm) carry important semantic meaning (Bolinger, 1961, 1989) that is lost in text-based systems. Understanding whether current E2E architectures can capture and utilize this information is crucial for moving beyond simple text prediction toward semantically complete translation.

This dissertation addresses these challenges. We propose methods to bridge the segmentation gap, enable zero-shot translation through cross-modal alignment, and evaluate how well prosody is preserved in translation.

1.2 Main Contributions

The contributions of this thesis are organized around tackling three critical bottlenecks that currently limit the utility of E2E ST: the *Input Processing* bottleneck, the *Training Data* bottleneck, and the *Prosodic Evaluation* bottleneck. A visualization of the contributions inside the ST framework is provided in Figure 1.1.

Addressing the Input Processing Bottleneck: Segmentation. Real-world speech does not arrive in neat sentences. To address the mismatch between static training data and continuous audio streams, we propose SHAS (Supervised Hybrid Audio Segmentation). We demonstrate that by training a classifier on self-supervised speech representations to mimic manual segmentation, we can process continuous audio with minimal loss in translation quality. Furthermore, we introduce SEGAUGMENT, a data augmentation strategy that leverages this segmentation flexibility. By re-segmenting training data with different length constraints, we generate diverse views of the input, maximizing data utility in low-resource settings.

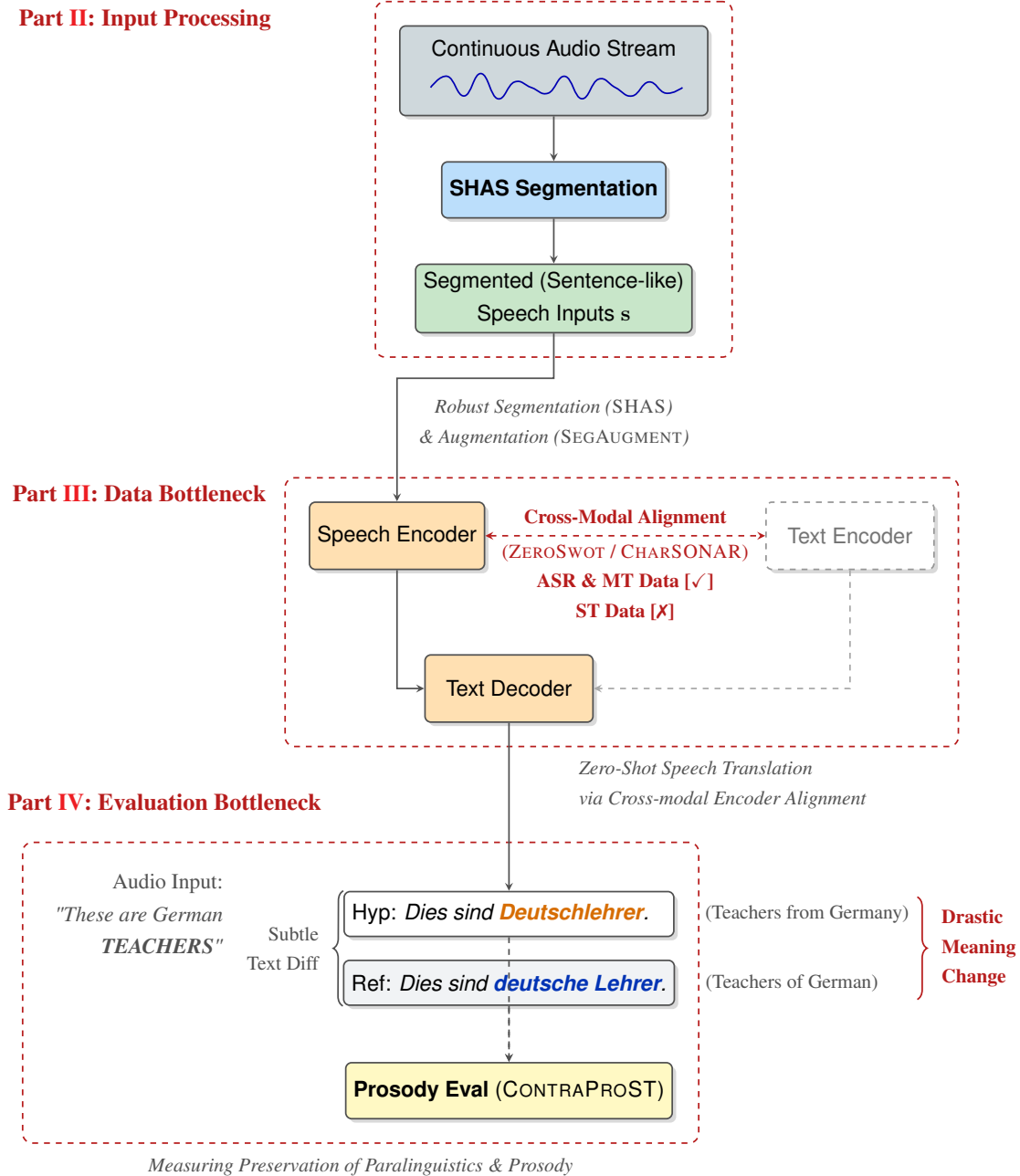


Figure 1.1: Thesis Overview. (Part II) We process continuous audio into sentence-like segments with SHAS, and furthermore use it to perform data augmentation with SEGAUGMENT. (Part III) We enable zero-shot ST by aligning a Speech Encoder to a pre-trained Text Encoder with the ZEROSWOT or CHARSONAR methodologies, allowing a connection to the Text Decoder. (Part IV) We evaluate if prosodic nuances (e.g., stress on "TEACHERS") are preserved in the translation with the CONTRAPROST benchmark.

Addressing the Training Data Bottleneck: Cross-Modal Alignment. The scarcity of paired speech-to-translation data hinders scaling to low-resource languages. We tackle

this by developing zero-shot methods that bridge the modality gap. We first propose ZEROSWOT, which aligns a speech encoder with a massively multilingual MT model using Optimal Transport and CTC compression, enabling translation without any paired ST data. We further improve this alignment with CHARSONAR, showing that character-level modeling significantly enhances not only transfer between modalities, but also transfer across languages.

Addressing the Evaluation Bottleneck: Prosody and Meaning. Current metrics focus on lexical correctness, often missing the nuances of spoken communication. We investigate the semantic richness of E2E translations by introducing CONTRAPROST, a benchmark and a methodology for evaluating prosody awareness. Our analysis reveals that while E2E models possess the internal capacity to represent features like contrastive stress, they often fail to manifest these cues in the final translation, suggesting a need for better inductive biases in training.

1.3 PhD Outcomes

The research presented in this dissertation was conducted at the Universitat Politècnica de Catalunya, with significant parts developed during four research internships at Meta AI (FAIR), Apple (AI/ML), and Dolby Laboratories.

1.3.1 Publications

The list of main author publications during the doctoral research period is summarized in Table 1.1.

	Paper	Venue
(Tsiamas et al., 2022b)	Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation	IWSLT 2022
(Tsiamas et al., 2022a)	SHAS: Approaching Optimal Segmentation for End-to-End Speech Translation	Interspeech 2022
(Tsiamas et al., 2023b)	Speech Translation with Foundation Models and Optimal Transport	IWSLT 2023
(Tsiamas et al., 2023c)	Efficient Speech Translation with Dynamic Latent Perceivers	ICASSP 2023
(Tsiamas et al., 2023a)	SegAugment: Maximizing the Utility of Speech Translation Data with Segmentation-based Augmentations	EMNLP 2023 (Findings)
(Tsiamas et al., 2024a)	Pushing the Limits of Zero-shot End-to-End Speech Translation	ACL 2024 (Findings)
(Tsiamas et al., 2024b)	Speech Is More Than Words: Do Speech-to-Text Translation Systems Leverage Prosody?	WMT 2024
(Tsiamas et al., 2025b)	Sequential Contrastive Audio-Visual Learning	ICASSP 2025
(Tsiamas et al., 2025a)	Improving Language and Modality Transfer in Translation by Character-level Modeling	ACL 2025

Table 1.1: Publications as main author during the doctoral research period. The publications included in this thesis are shown in bold.

The following publications are included in the thesis:

- (Tsiamas et al., 2022a) Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, Marta R. Costa-jussà (2022). *SHAS: Approaching Optimal Segmentation for End-to-End Speech Translation*. Proceedings of Interspeech 2022. (see Ch. 3)
- (Tsiamas et al., 2023a) Ioannis Tsiamas, José A. R. Fonollosa, Marta R. Costa-jussà (2023). *SegAugment: Maximizing the Utility of Speech Translation Data with Segmentation-based Augmentations*. Findings of the Association for Computational Linguistics (EMNLP 2023). (see Ch. 4)
- (Tsiamas et al., 2024a) Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, Marta R. Costa-jussà (2024). *Pushing the Limits of Zero-shot End-to-End Speech Translation*. Findings of the Association for Computational Linguistics (ACL 2024). (see Ch. 5)
- (Tsiamas et al., 2024b) Ioannis Tsiamas, Matthias Sperber, Andrew Finch, Sarthak Garg (2024). *Speech Is More Than Words: Do Speech-to-Text Translation Systems*

Leverage Prosody? Proceedings of the Ninth Conference on Machine Translation (WMT 2024). (see Ch. 7)

- (Tsiamas et al., 2025a) Ioannis Tsiamas, David Dale, Marta R. Costa-jussà (2025). *Improving Language and Modality Transfer in Translation by Character-level Modeling*. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025). (see Ch. 6)

To keep the thesis focused on the three main topics that we address in Speech Translation, we decided not to include the following main-author publications. See Appendix E for more information regarding these works:

- (Tsiamas et al., 2022b) Ioannis Tsiamas, Gerard I. Gállego, Carlos Escolano, José A. R. Fonollosa, Marta R. Costa-jussà (2022). *Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation*. Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022). (see §E.1)
- (Tsiamas et al., 2023b) Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, Marta R. Costa-jussà (2023). *Speech Translation with Foundation Models and Optimal Transport*. Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023). (see §E.2)
- (Tsiamas et al., 2023c) Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, Marta R. Costa-jussà (2023). *Efficient Speech Translation with Dynamic Latent Perceivers*. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023). (see §E.3)
- (Tsiamas et al., 2025b) Ioannis Tsiamas, Santiago Pascual, Chunghsin Yeh, Joan Serrà (2025). *Sequential Contrastive Audio-Visual Learning*. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2025). (see §E.4)

1.3.2 Industrial Collaborations

Several contributions in this thesis stem from collaborations with industrial research labs, which are summarized in Table 1.2.

Institution	Period	Location	Research Focus	Supervisor
Dolby Labs	Oct 2023 – Feb 2024 (4 months)	Barcelona	Sequential contrastive learning for audio-visual representations.	S. Pascual
Apple AI/ML	Apr 2024 – Jul 2024 (4 months)	Aachen	Evaluation of prosody and paralinguistics in speech translation.	M. Sperber
Meta FAIR	Aug 2024 – Feb 2025 (6 months)	Paris	Character-level modeling and knowledge transfer in translation.	M. Costa-jussà
Meta FAIR Omnilingual Team	Mar 2025 – Nov 2025 (9 months)	Paris & Berlin	Scaling multilingual embedding spaces and machine translation quality estimation to thousands of languages.	M. Costa-jussà

Table 1.2: Industrial Collaborations

1.4 Thesis Outline

The remainder of this thesis is structured as follows:

Part **I** closes with Ch. 2, which provides the *Technical Background* for the topics discussed in the thesis. It covers Deep Learning, NLP, Sequence-to-Sequence modeling, Transformers, and the foundations of ASR, Multilingual MT and Speech Translation.

Part **II** focuses on *Audio Segmentation*. Ch. 3 presents the SHAS segmentation method, while Ch. 4 describes SEGAUGMENT, a data augmentation method that leverages SHAS.

Part **III** addresses *Cross-Modal Alignment*. Ch. 5 details ZEROSWOT, where we enable zero-shot translation via cross-modal alignment using optimal transport. Ch. 6 introduces CHARSONAR, which refines the cross-modal alignment by character-level modeling.

Part **IV** is focused on *Prosody Evaluation*, providing a benchmark and an evaluation methodology to assess how well Speech Translation models utilize paralinguistic information.

Finally, Part **V** summarizes our *Conclusions* (Ch. **8**), provides a modern retrospective on the ideas of the thesis (Ch. **9**), and discusses future research directions (Ch. **10**).

2 Background and State of the Art

2.1 Deep Learning and Natural Language Processing

Deep Learning (DL) has fundamentally transformed Natural Language Processing (NLP) by replacing manual feature engineering with hierarchical representation learning. In the context of Speech Translation (ST), these methods allow us to map high-dimensional continuous signals (audio) or discrete sequences (text) into dense vector spaces where semantic and acoustic relationships can be modeled mathematically. This section introduces foundational concepts in neural networks, sequence modeling, and input representation that underpin the contributions of this thesis.

2.1.1 Neural Networks and Optimization

A neural network can be formalized as a parameterized function $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{V}|}$ that maps an input $\mathbf{x} \in \mathcal{X}$ to a vector of real-valued scores (*logits*) $\mathbf{z} = f_\theta(\mathbf{x})$. To interpret these scores as a probability distribution over a discrete output vocabulary $\mathcal{V}$ (e.g., the set of all possible tokens), we apply the softmax function $\sigma(\cdot)$:

$$P(y = i \mid \mathbf{x}; \theta) = \sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j \in \mathcal{V}} e^{z_j}} \quad (2.1)$$

The behavior of the network is determined by the set of parameters θ . In a supervised learning setting, given a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{D_{size}}$, the goal is to find the optimal parameters θ^* that minimize a loss function $\mathcal{L}$. For sequence generation tasks, such as predicting the next token, the standard objective is to minimize the Cross-Entropy (CE) loss summed over the sequence length K :

$$\mathcal{L}_{CE}(\theta) = - \sum_{k=1}^K \log P(y_k \mid \mathbf{y}_{<k}, \mathbf{x}; \theta) \quad (2.2)$$

Computing the gradient over the entire dataset $\mathcal{D}$ is computationally prohibitive. Therefore, optimization is typically performed using *mini-batches*. We approximate the true gradient using a small subset $\mathcal{B} \subset \mathcal{D}$ and iteratively update the parameters:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \left(\frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{B}} \mathcal{L}_{CE}(\mathbf{x}, \mathbf{y}; \theta_t) \right) \quad (2.3)$$

where η is the learning rate. The gradients ∇_{θ} are computed using the backpropagation algorithm (Rumelhart et al., 1986). While standard Stochastic Gradient Descent (SGD) is the theoretical foundation, modern methodologies rely on adaptive optimizers like AdamW (Loshchilov and Hutter, 2019), which adjust the learning rate for each parameter individually based on gradient moments.

2.1.2 Sequence Modeling

Natural language and speech are inherently sequential. In Machine Translation (MT) and Speech Translation (ST), the problem is framed as conditional sequence generation.

Given a source sequence $\mathbf{x} = (x_1, \dots, x_N)$ (or $\mathbf{s} = (s_1, \dots, s_T)$ in the case of speech) and a target sequence $\mathbf{y} = (y_1, \dots, y_K)$, we aim to model the conditional probability distribution $P(\mathbf{y} | \mathbf{x})$. Using the probability chain rule, this joint distribution is factorized autoregressively:

$$P(\mathbf{y} | \mathbf{x}; \theta) = \prod_{k=1}^K P(y_k | \mathbf{y}_{<k}, \mathbf{x}; \theta) \quad (2.4)$$

where $\mathbf{y}_{<k}$ denotes the history of generated tokens up to step k .

Training vs. Inference. During training, we typically employ *teacher forcing*, where the model predicts y_k conditioned on the ground-truth history $\mathbf{y}_{<k}$ rather than on its own previous predictions. This enables parallelization within a batch. At inference time, however, the ground truth is unavailable, so the model must generate the sequence step by step while conditioning on its own generated history. Since finding the globally optimal (argmax) sequence is intractable, we rely on approximate search strategies such as *greedy decoding* (selecting the highest-probability token at each step) or *beam search* (maintaining a set of B most probable hypotheses).

While early approaches utilized Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) to encode the history, these architectures suffered from sequential computation bottlenecks. These limitations necessitated the shift toward the Transformer architecture (Vaswani et al., 2017) (discussed in §2.2), which relies on attention mechanisms to model dependencies regardless of their distance in the sequence.

2.1.3 Tokenization and Input Representation

Neural networks operate on continuous vectors, yet textual data consists of discrete symbols. Transforming raw text into a format suitable for processing involves two distinct steps: *tokenization* and *embedding lookup*.

Tokenization. Tokenization is the process of segmenting a text sequence into a sequence of atomic discrete units, or tokens, $y \in \mathcal{V}$. The choice of granularity represents a critical trade-off between sequence length (N or K), vocabulary size $|\mathcal{V}|$, and semantic density.

- **Subword Segmentation:** Standard multilingual models, such as NLLB (NLLB, 2024) or SONAR (Duquenne et al., 2023), rely on subword algorithms like Byte Pair Encoding (BPE) (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018). Subwords offer a balance between the sparsity of word-level models and the length of character-level sequences, mitigating the out-of-vocabulary (OOV) problem by decomposing rare words into meaningful sub-units.
- **Character-level Modeling:** Alternatively, the vocabulary $\mathcal{V}_{char}$ can consist strictly of atomic characters. While this maximizes knowledge transfer across languages and robustness to noise (see Ch. 6), it results in significantly longer sequences ($K_{char} > K_{sub}$), exacerbating the computational complexity of the model.

Embedding Tables. Once tokenized, the discrete sequence must be projected into a continuous space. This is achieved via a learnable *Embedding Table* $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $|\mathcal{V}|$ is the vocabulary size and d is the model dimension. For each position k , the discrete token index y_k is used to look up the corresponding dense vector $\mathbf{e}_k = \mathbf{E}[y_k]$. These continuous vectors serve as the actual input to the neural network layers described in §2.1.2.

2.2 The Transformer Architecture

The Transformer (Vaswani et al., 2017), originally proposed for machine translation, has become the standard architecture for sequence modeling across modalities. Unlike Recurrent Neural Networks (RNNs), which process data sequentially, the Transformer relies entirely on attention mechanisms to model global dependencies.

The architecture consists of a stack of L identical layers. We denote the sequence of continuous representations (or *hidden states*) input to a given layer as $\mathbf{H} \in \mathbb{R}^{N \times d}$, where N is the sequence length. For the first layer, $\mathbf{H}$ corresponds to the sum of the input embeddings and positional encodings. For subsequent layers, it corresponds to the output of the previous layer.

2.2.1 Self-Attention and Multi-Head Attention

The fundamental building block of the Transformer is the *Scaled Dot-Product Attention*. This mechanism allows the model to weigh the importance of different elements in the input sequence $\mathbf{H}$. It operates on three matrices: Queries $\mathbf{Q}$, Keys $\mathbf{K}$, and Values $\mathbf{V}$, which are obtained via linear projections of $\mathbf{H}$:

$$\mathbf{Q} = \mathbf{H}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{H}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{H}\mathbf{W}^V \quad (2.5)$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d_k}$ are learnable parameter matrices.

The attention mechanism computes a weighted sum of the values, where the weights are determined by the compatibility of the query with the corresponding key. As shown in Figure 2.2, this involves a matrix multiplication of $\mathbf{Q}$ and $\mathbf{K}^T$, scaled by $\frac{1}{\sqrt{d_k}}$ to prevent vanishing gradients:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \right) \mathbf{V} \quad (2.6)$$

An optional masking step is applied during decoding to preserve the autoregressive property by setting attention scores to $-\infty$ for future positions.

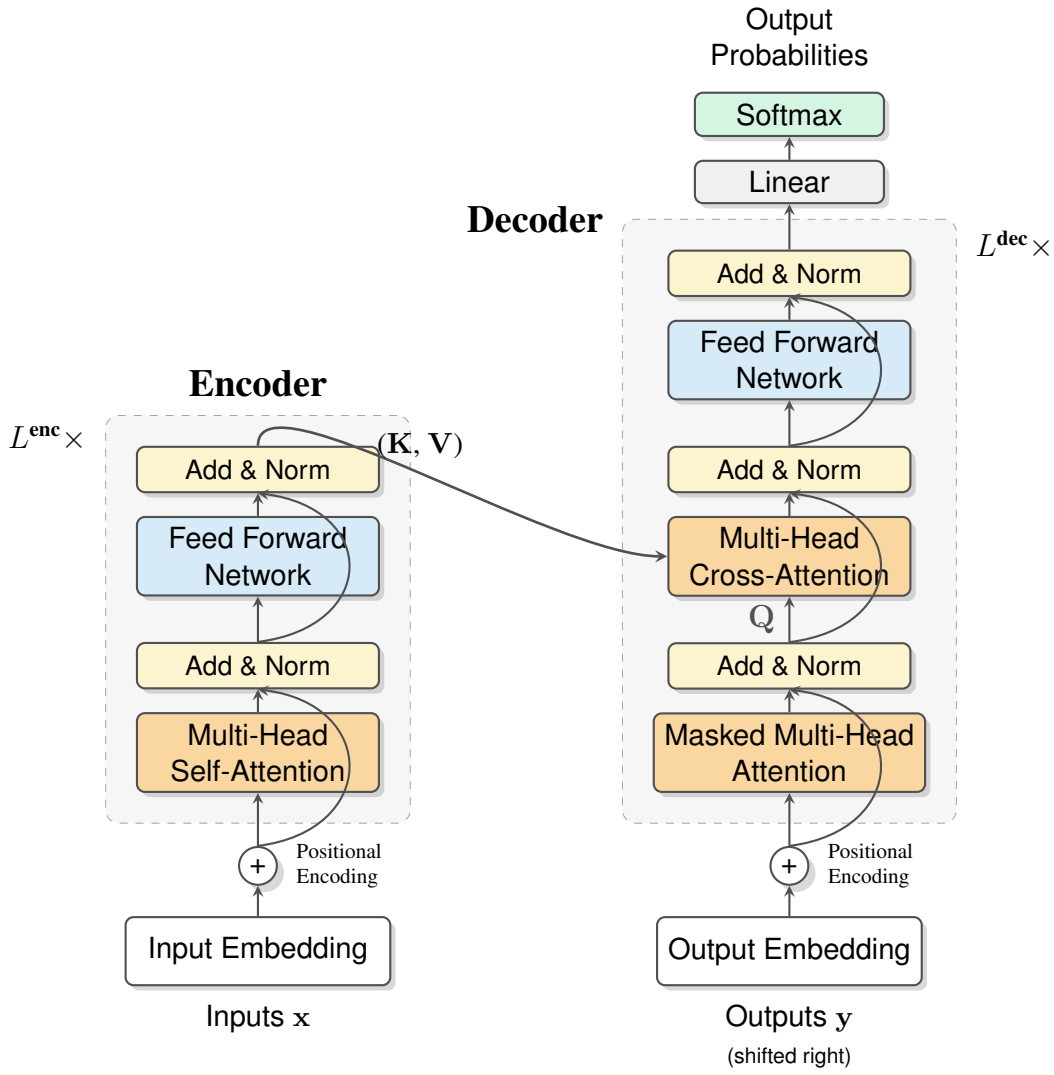


Figure 2.1: The Transformer Architecture. The Encoder (left) maps the input tokens into a sequence of continuous representations $\mathbf{H}_{\text{enc}}$. The Decoder (right) attends to $\mathbf{H}_{\text{enc}}$ via Cross-Attention while autoregressively generating the output. Adapted from Vaswani et al. (2017).

To capture diverse linguistic and acoustic features simultaneously, the Transformer extends this into *Multi-Head Attention* (MHA). As visualized in Figure 2.3, MHA projects the input $\mathbf{H}$ into l distinct subspaces through different heads. The attention function is applied in parallel to these projections, and the results are concatenated and projected back to the model dimension via $\mathbf{W}^O$:

$$\text{MHA}(\mathbf{H}) = \text{Concat}(\mathbf{U}_1, \dots, \mathbf{U}_l) \mathbf{W}^O \quad (2.7)$$

where $\mathbf{U}_i$, is the i -th head, and is obtained with $\text{Attention}(\mathbf{H}\mathbf{W}_i^Q, \mathbf{H}\mathbf{W}_i^K, \mathbf{H}\mathbf{W}_i^V)$.

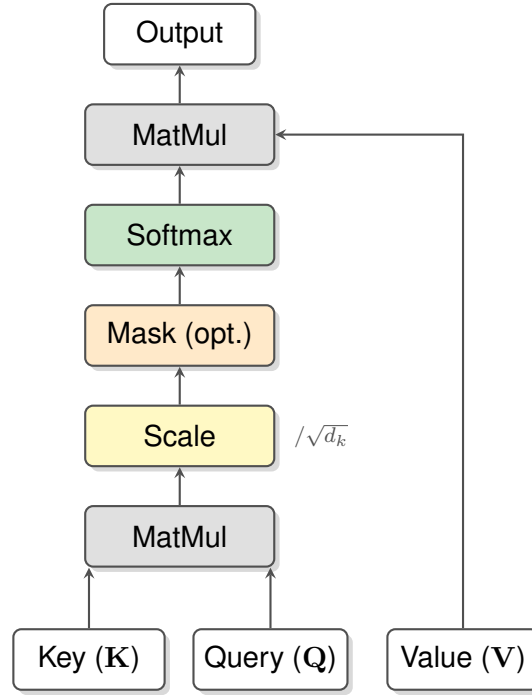


Figure 2.2: Scaled Dot-Product Attention. The mechanism computes compatibility scores between Queries (**Q**) and Keys (**K**), scales them by $1/\sqrt{d_k}$, and applies Softmax to generate weights for the Values (**V**). The optional masking step is utilized primarily in the Decoder.

2.2.2 The Encoder-Decoder Structure

Encoder. The encoder is a stack of layers that maps the input sequence into contextualized representations. Each layer consists of two sub-layers: Multi-Head Self-Attention and a Position-wise Feed-Forward Network (FFN).

The FFN is applied to each position i identically and independently. Given a position vector $\mathbf{h}_i \in \mathbb{R}^d$ (a row of $\mathbf{H}$), the FFN consists of two linear transformations with a ReLU activation:

$$\text{FFN}(\mathbf{h}_i) = \max(0, \mathbf{h}_i \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2 \quad (2.8)$$

To facilitate training, a Residual Connection (He et al., 2016) is employed around each sub-layer, followed by Layer Normalization (Ba et al., 2016). The output of any sub-layer is $\mathbf{H}' = \text{LayerNorm}(\mathbf{H} + \text{Sublayer}(\mathbf{H}))$.

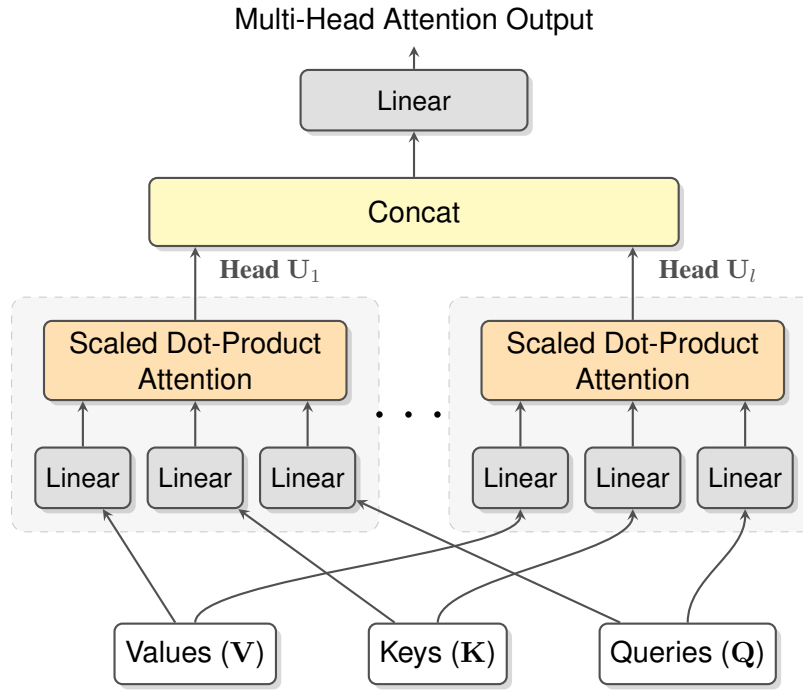


Figure 2.3: Multi-Head Attention (MHA). The input is projected into l parallel subspaces. Each head performs Scaled Dot-Product Attention independently, and their outputs are concatenated and linearly projected to form the final output.

Decoder. The decoder generates the output sequence autoregressively. In addition to the two sub-layers present in the encoder, the decoder inserts a third sub-layer: *Cross-Attention*. In this module, the Queries Q arise from the decoder’s previous layer (representing the target history), while the Keys K and Values V are projected from the final encoder output H_{enc} . This mechanism allows the decoder to attend to relevant parts of the source input (e.g., specific speech frames) during generation.

2.2.3 Positional Encodings

Since the self-attention mechanism is permutation invariant, it possesses no inherent notion of order. To provide position information, Positional Encodings (PE) are added to the input embeddings E before the first layer: $H^{(0)} = E + PE$.

In the original Transformer, fixed sinusoidal functions are used:

$$PE_{(pos, 2j)} = \sin(pos/10000^{2j/d}) \quad (2.9)$$

$$PE_{(pos, 2j+1)} = \cos(pos/10000^{2j/d}) \quad (2.10)$$

For speech applications, alternative approaches such as learned positional embeddings or relative positional encoding schemes (Su et al., 2024) are often employed to better capture the continuous nature of the audio signal.

2.3 Speech Processing Foundations

While text processing begins with discrete symbols, speech processing must handle continuous, high-dimensional physical signals. Before a neural network can process speech, the raw acoustic waveform must be digitized and transformed into a meaningful representation. This section outlines the transition from signal processing to modern representation learning.

2.3.1 Digital Signal Processing (DSP)

A speech signal is a continuous variation in air pressure over time. To process it digitally, we discretize it via *sampling*, capturing the amplitude at regular intervals (e.g., 16,000 times per second for 16kHz audio). Although end-to-end models can theoretically process raw waveforms directly, the standard practice is to extract features that align with human auditory perception.

Spectrograms and Mel-Filterbanks. The raw waveform is transformed from the time domain to the frequency domain using the Short-Time Fourier Transform (STFT). This applies a sliding window (e.g., 25ms) over the signal, computing the Discrete Fourier Transform (DFT) for each frame. The magnitude of the result yields the *spectrogram*, a visual representation of the spectrum of frequencies as they vary with time.

Since human hearing is non-linear, and we perceive frequency differences better at lower pitches than at higher ones, we apply the *Mel-scale* transformation. The frequency axis is warped (quasi-)logarithmically to match human pitch perception, and the energy is summed into frequency bands (filterbanks). The resulting *Log Mel-filterbank* features are the standard input for most acoustic models, typically consisting of 80-dimensional vectors computed every 10ms.

2.3.2 Prosody and Paralinguistic Features

Speech carries significantly more information than just the linguistic content (words). *Prosody* refers to the suprasegmental features of speech, which apply to syllables, words, or phrases rather than individual phonemes. It is primarily composed of three acoustic correlates:

- **Pitch (F0):** The fundamental frequency of the voice, perceived as intonation. It distinguishes questions from statements and, in tonal languages, determines lexical meaning.
- **Duration/Rhythm:** The timing and length of phonemes and pauses. It signals stress, emphasis, and speaking rate.
- **Intensity/Loudness:** The energy of the signal, used to highlight important information (contrastive stress).

In Speech Translation, these features are critical for preserving the correct semantics and pragmatic intent of the speaker (e.g., sarcasm or urgency). As we discuss in Ch. 7, text-based translation systems inherently lose this information, motivating the use of end-to-end speech-to-text architectures that can theoretically capture and translate these paralinguistic cues.

2.3.3 Self-Supervised Learning (SSL) in Speech

Similar to the “BERT revolution” in NLP (Devlin et al., 2019), speech processing has shifted from supervised training on Mel-filterbanks to *Self-Supervised Learning* (SSL) on raw waveforms. These models learn rich contextual representations from massive amounts of unlabeled audio.

WAV2VEC 2.0. The foundational model for much of this thesis is WAV2VEC 2.0 (Baevski et al., 2020), illustrated in Figure 2.4. The architecture processes the raw input waveform *s* through two primary components:

1. **Feature Encoder:** A multi-layer convolutional neural network (CNN) processes the

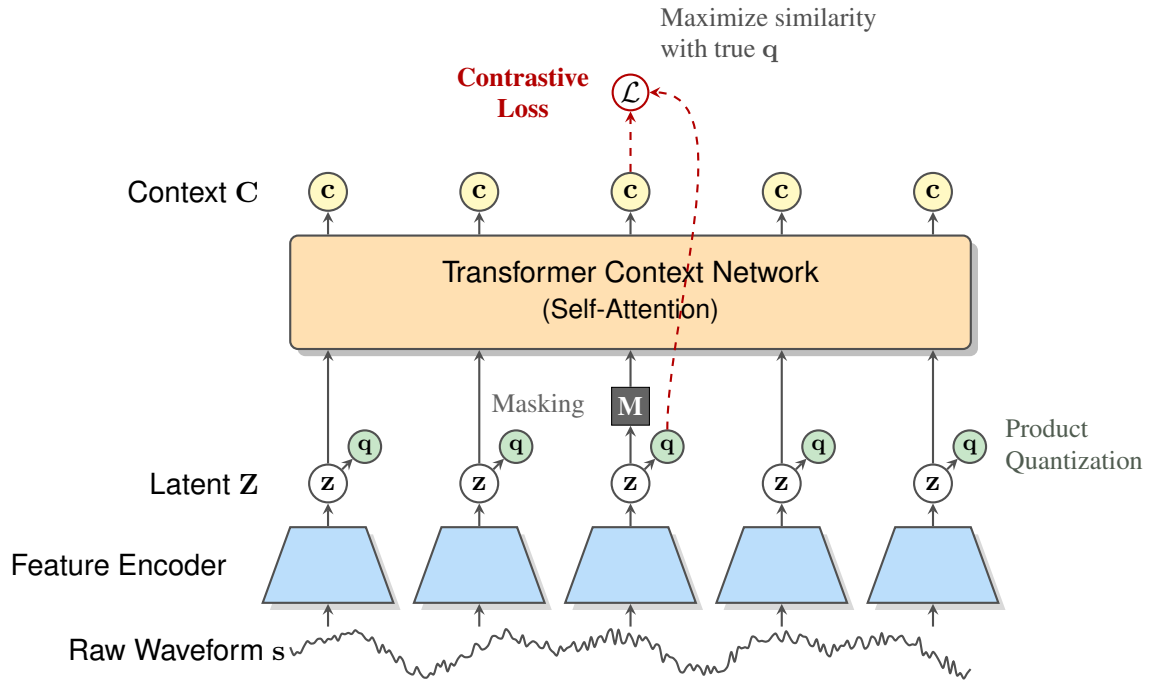


Figure 2.4: The WAV2VEC 2.0 Architecture. A convolutional Feature Encoder maps raw audio s to a sequence of latent representations $\mathbf{Z}$. These are discretized via Product Quantization into targets $\mathbf{Q}$. Simultaneously, a masked version $\tilde{\mathbf{Z}}$ is fed into the Transformer (Context Network) to produce context representations $\mathbf{C}$. The model is trained to minimize the contrastive loss $\mathcal{L}$ by identifying the correct quantized target $\mathbf{q}_t$ corresponding to the masked context $\mathbf{c}_t$.

raw audio s to generate a sequence of latent speech representations $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_T)$ (depicted as blue trapezoids in the figure).

2. **Context Network:** A Transformer stack builds context representations $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_T)$ from the latent inputs.

The model is trained via a *Contrastive Task*. As shown in the diagram, the latent vectors $\mathbf{Z}$ are processed via two pathways. In the first, they are discretized via a Product Quantization module to form discrete targets $\mathbf{Q}$. In the second, a proportion of the time-steps in $\mathbf{Z}$ are *masked* (replaced with a learned vector) to form $\tilde{\mathbf{Z}}$, which is then fed into the Transformer.

The objective function forces the model to predict the correct quantized representation $\mathbf{q}_t$ for a masked time-step t using the context vector $\mathbf{c}_t$, distinguishing it from distinct “distractor” samples sampled from other time steps. This encourages the model to learn high-level, phoneme-like acoustic units rather than low-level signal details.

HUBERT. Hidden Unit BERT (HUBERT) (Hsu et al., 2021) simplifies this by using an offline clustering step (k-means) on the acoustic features to generate pseudo-labels, rather than learning the quantization online. The model is then trained with a masked prediction loss, similar to BERT (Devlin et al., 2019), where it directly predicts the cluster index for masked frames.

These pre-trained encoders serve as powerful feature extractors, significantly reducing the amount of labeled data required for downstream tasks like ASR and ST.

2.3.4 Audio Segmentation

In real-world scenarios, speech translation models must process continuous audio streams rather than pre-segmented sentences. Since Transformer-based models have a limited context window and quadratic memory complexity, processing arbitrarily long streams is infeasible. Therefore, the continuous signal must be split into manageable segments.

Traditional approaches rely on Voice Activity Detection (VAD) (Sohn et al., 1999), which detects pauses based on energy thresholds or statistical models. However, VAD-based segmentation often splits audio at acoustically silent but linguistically inappropriate points (e.g., in the middle of a sentence), leading to a significant degradation in translation quality known as the *segmentation mismatch* (Ansari et al., 2020; Gaido et al., 2021b). This limitation motivates the need for semantic-aware segmentation methods, such as the SHAS method proposed in Ch. 3.

2.4 Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) is the task of transcribing a continuous speech signal into a discrete sequence of words or characters. It serves as the bridge between the acoustic foundations discussed in §2.3 and the symbolic text processing methods of §2.1.

2.4.1 The ASR Task

Formally, given a sequence of acoustic features $\mathbf{s} = (s_1, \dots, s_T)$ (e.g., Log Mel-filterbanks, WAV2VEC 2.0 representations, or raw signal), the goal of ASR is to find the most probable sequence of distinct linguistic units $\mathbf{x} = (x_1, \dots, x_N)$, where N is the length of the transcription:

$$\hat{\mathbf{x}} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{V}^*} P(\mathbf{x}|\mathbf{s}) \quad (2.11)$$

A fundamental challenge in ASR is the *length mismatch*: the acoustic sequence length T is typically much larger than the transcription length N (e.g., 1000 frames for a 10-word sentence). Furthermore, the alignment between speech frames and text tokens is unknown and not necessarily monotonic (i.e., non-monotonic) due to pauses, varying speaking rates, and noise.

2.4.2 Connectionist Temporal Classification (CTC)

To address the length mismatch and alignment issues without relying on complex HMM-based alignment pipelines, [Graves et al. \(2006\)](#) introduced the *Connectionist Temporal Classification* (CTC) loss. This algorithm is central to the methodologies proposed in this thesis (see Chs. 5 and 6) as it provides a mechanism to map acoustics to text in a non-autoregressive manner.

CTC introduces a special *blank token*, denoted as ϵ , extending the vocabulary to $\mathcal{V}' = \mathcal{V} \cup \{\epsilon\}$. The network predicts a probability distribution over $\mathcal{V}'$ for every input frame t , independently of other frames (conditional independence assumption).

Let $\pi = (\pi_1, \dots, \pi_T)$ be a path of tokens chosen from $\mathcal{V}'$ at each step. CTC defines a many-to-one mapping function $\mathcal{B}$ that transforms a path π into a label sequence $\mathbf{x}$ by:

1. Merging consecutive repeated characters.
2. Removing all blank tokens ϵ .

For example, $\mathcal{B}(\text{h, h, e, l, } \epsilon, \text{l, o}) = \text{hello}$. See Figure 2.5.

The probability of a label sequence $\mathbf{x}$ is the sum of the probabilities of all valid paths π

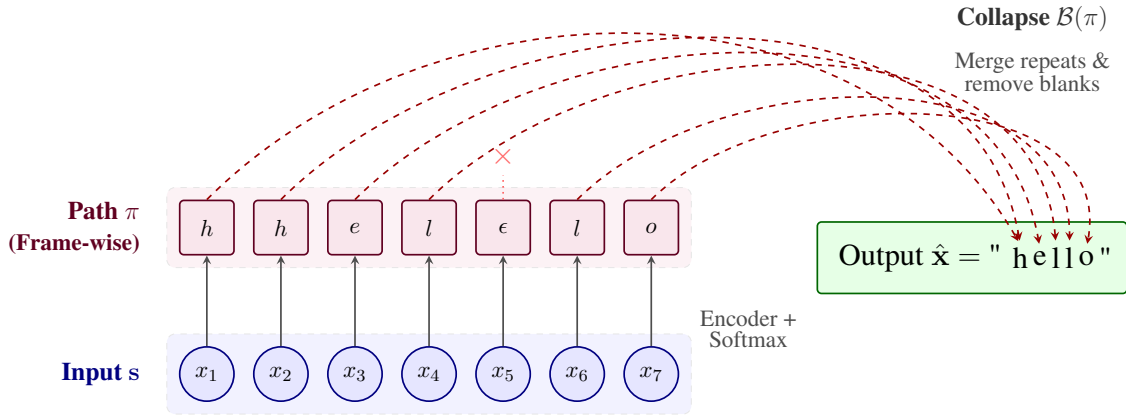


Figure 2.5: The CTC Collapse Process. An example frame alignment path π is shown predicting h, h, e, l, ϵ , l, o. The repeated ‘h’s are merged, and the blank token ϵ between the ‘l’s is removed but also prevents them from merging. The final sequence $\mathbf{x}$ is obtained by merging consecutive repeats and removing the blanks.

that map to it:

$$P(\mathbf{x}|\mathbf{s}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{x})} P(\pi|\mathbf{s}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{x})} \prod_{t=1}^T P(\pi_t|s_t) \quad (2.12)$$

The loss function is simply the negative log-likelihood of the ground truth transcription $\mathbf{x}^*$:

$$\mathcal{L}_{\text{CTC}} = -\log P(\mathbf{x}^*|\mathbf{s}) \quad (2.13)$$

Although the summation over all paths appears computationally expensive, it is efficiently computed using a dynamic programming algorithm similar to the Forward-Backward algorithm.

Crucially, CTC provides a frame-level alignment in which regions of the input are explicitly associated with output characters or the blank state. In Ch. 5, we leverage this property to perform *CTC compression* (Gaido et al., 2021a). By collapsing consecutive repeated tokens and subsequently removing blank frames, we can compress the acoustic representations $\mathbf{H}$ into a sequence $\mathbf{H}'$ whose length T' is closer to that of the text sequence N , facilitating better alignment with text embeddings. In Chs. 5 and 6, we explore two ways to use these compressed representations to bridge the modality gap.

2.4.3 Encoder-Decoder ASR

While CTC is computationally efficient, its conditional independence assumption prevents it from explicitly modeling dependencies between output tokens (e.g., learning that “q” is almost always followed by “u”). To overcome this, modern ASR systems employ the *Sequence-to-Sequence* (Seq2Seq) or *Encoder-Decoder* architecture (Chorowski et al., 2015; Chan et al., 2016).

In this framework, the problem is modeled directly as a conditional generation task $P(\mathbf{x}|\mathbf{s})$. The input $\mathbf{s} = (s_1, \dots, s_T)$ represents the acoustic sequence, which may be a sequence of handcrafted features (e.g., Log Mel-filterbanks) or the raw waveform signal itself.

The architecture consists of two main components:

1. **Encoder:** Maps the acoustic input $\mathbf{s}$ to a high-level representation $\mathbf{H}^{\text{enc}}$. In modern systems, this is typically a Transformer (Dong et al., 2018) or Conformer (Gulati et al., 2020) encoder.
2. **Decoder:** Generates the transcription $\mathbf{x}$ autoregressively, one token at a time. At each step n , the decoder attends to the entire encoder output $\mathbf{H}^{\text{enc}}$ via Cross-Attention and conditions on its own previous predictions $\mathbf{x}_{<n}$:

$$P(x_n|\mathbf{x}_{<n}, \mathbf{s}) = \text{Decoder}(x_{n-1}, \mathbf{h}_{n-1}^{\text{dec}}, \text{CrossAttn}(\mathbf{H}^{\text{enc}})) \quad (2.14)$$

Unlike CTC, the Encoder-Decoder approach implicitly learns the alignment between speech and text through the attention mechanism, allowing it to capture strong linguistic context and dependencies. However, this comes at the cost of auto-regressive inference, which is generally slower than non-autoregressive CTC decoding.

2.5 Machine Translation and Multilingual Representations

While the previous sections focused on general sequence modeling, this section narrows the focus to Machine Translation (MT). We discuss how modern systems scale from bilingual mappings to massively multilingual embedding spaces, where text and speech can theoretically be aligned in a language-agnostic “interlingua”.

2.5.1 Neural Machine Translation (NMT)

Neural Machine Translation (NMT) redefines translation not as a rule-based or statistical phrase-matching process, but as a conditional probability problem modeled by a single end-to-end neural network. Given a source sentence $\mathbf{x} = (x_1, \dots, x_N)$ in a source language ℓ_s , the model aims to maximize the probability of the target sentence $\mathbf{y} = (y_1, \dots, y_K)$ in a target language ℓ_t . This joint probability is decomposed via the chain rule:

$$P(\mathbf{y}|\mathbf{x}; \theta) = \prod_{k=1}^K P(y_k | \mathbf{y}_{<k}, \mathbf{x}; \theta) \quad (2.15)$$

Early NMT systems (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015) relied on Recurrent Neural Networks (RNNs) to encode the source into a fixed-size context vector. However, this created an information bottleneck that degraded performance on long sentences. The introduction of the Transformer (Vaswani et al., 2017) resolved this by allowing the decoder to attend to *all* source positions simultaneously.

In a standard bilingual setting, the parameters θ are optimized by minimizing the negative log-likelihood over a parallel corpus $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}$. While effective, training separate models for every language pair ($O(\Lambda^2)$ models for Λ languages) is computationally intractable and fails to leverage linguistic similarities between related languages (e.g., Spanish and Catalan).

2.5.2 Multilingual NMT

Multilingual NMT (MNMT) extends the NMT paradigm to handle multiple language pairs (many-to-many) within a single model. One of the earliest explorations of this direction was by Ha et al. (2016), who demonstrated the feasibility of a single shared model for multiple language pairs. The standard approach, later popularized and scaled by Johnson et al. (2017), utilizes a *shared encoder-decoder* architecture with a shared vocabulary (typically built via Joint BPE).

Language Control. To indicate the desired target language, a special language token (e.g., <spa> for Spanish) is prepended to the source sequence or the decoder input. Sur-

prisingly, this simple signal is sufficient for the model to generate the correct language, implying that the shared encoder learns a form of neural *interlingua*. In this representation, sentences with the same meaning but different source languages are mapped to similar regions in the vector space. Subsequent work has explored more sophisticated language control mechanisms and extensions of language tags (Liu et al., 2021; Wicks and Duh, 2022).

Massively Multilingual Scaling. Recent years have seen a shift toward “massively” multilingual models covering hundreds of languages, such as M2M-100 (Fan et al., 2021), MBART-50 (Liu et al., 2020a), and NLLB (NLLB, 2024). These models demonstrate two key phenomena:

1. **Positive Transfer:** Low-resource languages benefit significantly from training alongside high-resource languages by sharing lexical and syntactic representations.
2. **The Curse of Multilinguality:** As the number of languages increases for a fixed model capacity, the per-language performance can eventually degrade due to parameter contention (Conneau et al., 2020). This necessitates larger models (scaling laws) or specialized techniques like Mixture-of-Experts (MoE) (NLLB, 2024) to maintain quality across all directions.

2.5.3 Multilingual Embedding Spaces

While standard NMT models map variable-length sequences to variable-length sequences, discriminative tasks (e.g., semantic search) and zero-shot alignment methods often require **fixed-size** vector representations.

LASER. The Language-Agnostic SEntence Representations (LASER) (Artetxe and Schwenk, 2019) was a pioneering approach that used a Bi-LSTM encoder to map sentences into a shared, fixed-size vector space. The encoder was trained via a translation objective, where the sentence embedding served as the initial state for the decoder.

SONAR. More recently, the SONAR architecture (Duquenne et al., 2023) (depicted in Figure 2.6) has established a new state-of-the-art for multimodal and multilingual em-

beddings. Unlike LASER, SONAR relies on a Transformer-based architecture, initialized from the NLLB-1.3B model (NLLB, 2024).

The key innovation in SONAR is the use of a *fixed-size bottleneck* within a Transformer Encoder-Decoder structure:

1. **Encoding:** The Transformer encoder (initialized with NLLB-1.3B weights) processes the input text $\mathbf{x}$. The resulting sequence of hidden states is then aggregated into a single vector $\mathbf{h} \in \mathbb{R}^d$ (typically $d = 1024$) via a pooling operation (e.g., Mean Pooling).
2. **Bottleneck Decoding:** Crucially, the decoder (also initialized from NLLB-1.3B) reconstructs the target text $\mathbf{y}$ by attending *only* to this single fixed vector $\mathbf{h}$. This differs from standard NMT, where the decoder uses cross-attention over the entire variable-length source sequence.
3. **Objective:** The model is trained with a multi-task objective that includes (a) translation (predicting a translation $\mathbf{y}$ from $\mathbf{x}$), (b) auto-encoding (reconstructing $\mathbf{x}$ from itself) to prevent catastrophic forgetting, and (c) embedding similarity (minimizing distance between $\mathbf{h}$ and a target embedding) to align the space.

2.6 End-to-End Speech Translation

Speech Translation (ST) is the task of translating an acoustic speech signal in a source language directly into text in a target language. It was historically treated as a niche application of ASR and MT, but it has evolved into a distinct field with its own architectural paradigms and challenges.

2.6.1 Cascade vs. End-to-End Paradigms

Traditionally, ST was approached via *Cascade* systems (Ney, 1999; Mathias and Byrne, 2006), which concatenate an Automatic Speech Recognition (ASR) model and a Machine Translation (MT) model. This implies having two separate systems, running two distinct

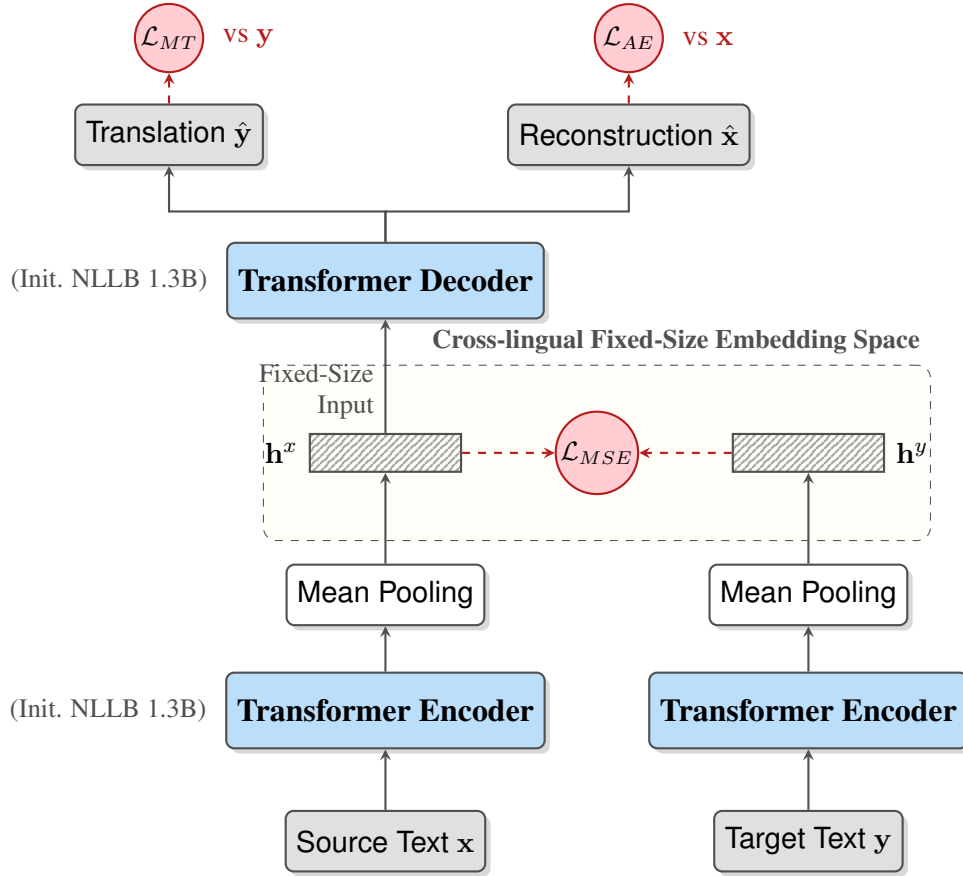


Figure 2.6: The SONAR architecture. The input sequence is encoded and pooled into a single fixed-size sentence embedding h . The decoder conditions solely on h to reconstruct the text (Auto-Encoding) or generate a translation (MT), creating a bottleneck that forces semantic density.

generation processes, one after the other.

$$\hat{x} = \underset{x}{\operatorname{argmax}} P_{\text{ASR}}(x|s), \quad \hat{y} = \underset{y}{\operatorname{argmax}} P_{\text{MT}}(y|\hat{x}) \quad (2.16)$$

where s is the source speech, $\hat{x}$ is the intermediate source transcription, and $\hat{y}$ is the target translation.

In contrast, *End-to-End* (E2E) ST (Bérard et al., 2016) employs a single model to map speech directly to the translation, maximizing $P(y|s)$ without explicit intermediate representations (see Figure 2.7).

The E2E paradigm offers several theoretical advantages (Sperber and Paulik, 2020):

1. **No Error Propagation:** In cascade systems, ASR errors (e.g., homophone con-

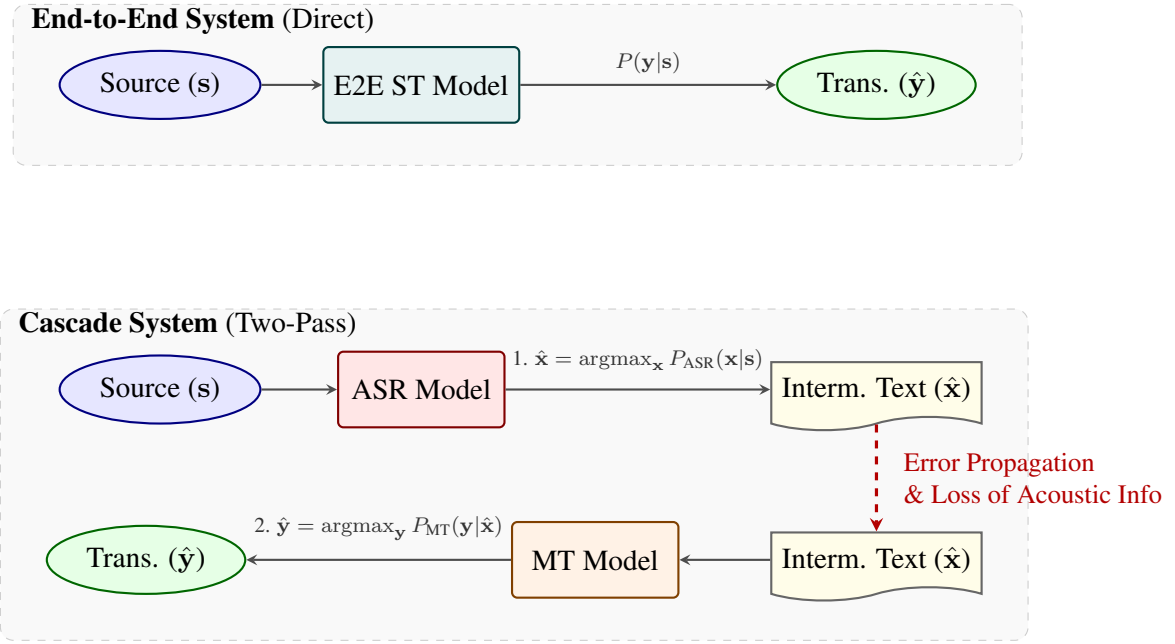


Figure 2.7: Comparison of End-to-End vs Cascade approaches. The Cascade system (bottom) flows from left to right then right to left, illustrating the two distinct generation steps separated by the information bottleneck.

fusion) are irreversible inputs to the MT system. E2E models can recover from acoustic ambiguity using translation context.

2. **Lower Latency:** By removing the intermediate decoding step, E2E models are theoretically faster and more suitable for real-time applications.
3. **Access to Paralinguistics:** As discussed in §2.3.2, the speech signal contains prosodic information (intonation, emotion) that is stripped away in the text transcription. E2E models preserve access to these cues.

2.6.2 Data Scarcity

Unlike Automatic Speech Recognition (ASR) and Machine Translation (MT), which benefit from massive monolingual or parallel text corpora, Speech Translation (ST) relies on triplets of (audio, transcription, translation). Acquiring such triplet data is expensive and difficult, making *data scarcity* one of the most significant barriers to scaling ST systems.

While several benchmarks exist, they represent a fraction of the data available for text-based tasks. The standard datasets used in this thesis include:

- **MuST-C** (Di Gangi et al., 2019a): A multilingual corpus built from TED talks, containing approximately 400 hours of speech data aligned with English transcriptions and translations into 8 European languages. It represents a high-resource scenario by ST standards, yet it is orders of magnitude smaller than standard MT benchmarks (e.g., WMT).
- **CoVoST & CoVoST-2** (Wang et al., 2020a,c): A large-scale multilingual ST corpus based on the Common Voice dataset (Ardila et al., 2020). Although it covers a diverse set of languages, many low-resource directions contain only a few hours of paired data, highlighting the severe scarcity in the “long tail” of languages.
- **FLEURS** (Conneau et al., 2023): A highly multilingual benchmark covering 102 languages derived from FLoRes-101 (Goyal et al., 2022). It is specifically designed to evaluate n-way parallel speech-to-text translation, which is crucial for assessing zero-shot capabilities in massively multilingual settings where training data is absent.

Despite these resources, the lack of large-scale paired data—and especially for non-English source languages—remains a primary bottleneck for End-to-End models.

2.6.3 The Modality Gap

The scarcity of paired data discussed above fundamentally exacerbates the *Modality Gap* (Liu et al., 2020b). This gap refers to the representational distance between the continuous, high-variance acoustic space and the discrete, semantic text space. When paired data is abundant, a model can learn to map between these spaces via supervision. However, in low-resource or zero-shot scenarios, the model fails to align the acoustic input with the correct semantic subspace of the decoder.

The modality gap manifests in two primary forms:

1. **Length Mismatch:** Speech sequences are significantly longer than their corresponding textual descriptions. A speech encoder typically outputs high-frequency frame sequences (e.g., 50Hz), resulting in thousands of acoustic frames (T) for a sentence that contains only a few dozen text tokens (N), i.e., $T \gg N$.
2. **Representation Mismatch:** Speech encoders capture acoustic features (e.g., pitch, speaker identity, noise), whereas text encoders operate on sparse token sequences that encode high-level semantic meaning. Even when powerful pre-trained encoders

exist for both modalities (e.g., WAV2VEC 2.0 and MBART-50), their latent spaces are not naturally aligned.

Bridging this gap without relying on scarce ST data is the primary motivation for the ZEROSWOT method presented in Ch. 5, which utilizes Optimal Transport to align these disparate modalities using only ASR data.

2.6.4 Speech Translation Architectures

Speech-to-Text Transformer. Speech sequences are inherently longer than their text counterparts. For instance, a 10-second utterance may correspond to only 15 text tokens ($N \approx 15$) but result in 1,000 acoustic frames ($T \approx 1000$ at 10ms/frame). Feeding such raw frame sequences directly into a standard Transformer is computationally intractable due to the quadratic $O(T^2)$ complexity of self-attention. To mitigate this, standard End-to-End ST architectures, such as the S2T-Transformer (Wang et al., 2020b), employ a dedicated *subsampling module* prior to the encoder. As illustrated in Figure 2.8, the input features pass through a stack of strided 1D Convolutional layers (typically two layers with kernel size 5 and stride 2). Each layer halves the sequence length, resulting in a total reduction factor of 4. The downsampled sequence is then projected to the model dimension d , combined with sinusoidal positional encodings, and processed by a standard Transformer Encoder-Decoder stack to generate the target translation.

Joint Speech-to-Text. To mitigate the scarcity of paired speech translation data, Tang et al. (2021a) proposed a multi-task learning framework where the ST model is trained jointly with an auxiliary Machine Translation (MT) task.

As illustrated in Figure 2.9, this architecture processes two modalities simultaneously: an acoustic encoder (similar to the encoder of the S2T-Transformer with down-sampling) processes the speech input s , while a shared semantic text encoder processes the output of the acoustic encoder or the source transcript x . The shared encoder feeds into a shared decoder to generate the translation y for either the speech or the text input. This parameter sharing strategy allows the data-scarce ST task to benefit from the richer linguistic representations and stronger language modeling capabilities learned from the abundant text translation data.

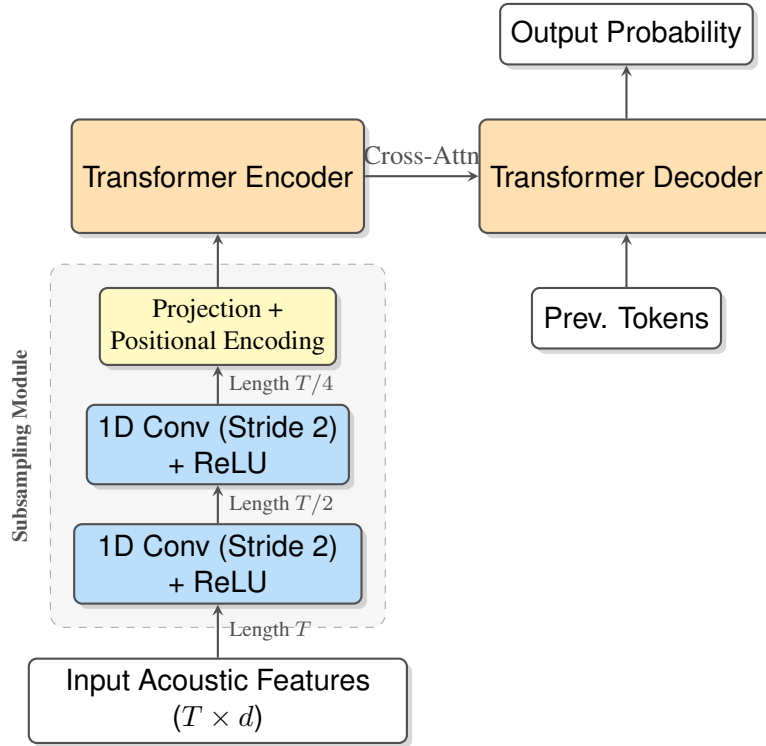


Figure 2.8: The Speech-to-Text (S2T) Transformer. To handle long acoustic sequences, the input passes through a convolutional subsampling module (typically two layers with stride 2) before entering the Encoder. This reduces the sequence length T to $T/4$, significantly lowering the computational cost of self-attention.

To further bridge the gap between modalities, two auxiliary objectives are often employed:

1. **Cross-Modal Regularization ($\mathcal{L}_{Reg}$):** This loss term minimizes the distance (e.g., L2 norm or Cosine distance) between the speech and text encoder outputs for parallel data. It forces the speech encoder to learn representations that are semantically aligned with the textual ones.
2. **Knowledge Distillation ($\mathcal{L}_{KD}$):** A knowledge distillation loss (Hinton et al., 2015) is applied where the output probability distribution of the auxiliary MT task acts as a “teacher” for the ST task. This guides the speech translation model to mimic the generally more accurate predictions of the text translation model.

The final training objective is a weighted sum of the standard Cross-Entropy ($\mathcal{L}_{CE}$) losses for both tasks, combined with the regularization and distillation terms.

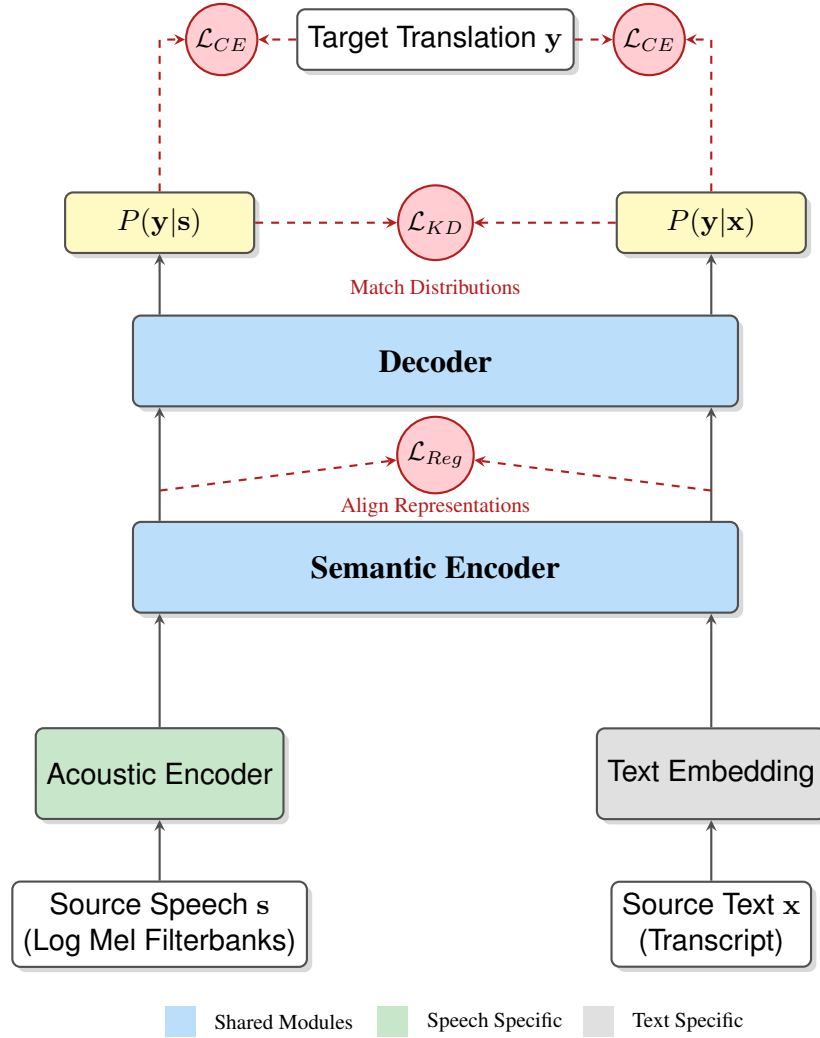


Figure 2.9: Joint Speech-to-Text Architecture. The model uses modality-specific modules for initial feature extraction, but shares the semantic encoder and the decoder between tasks. The training is regularized by aligning the Semantic Encoder outputs ($\mathcal{L}_{Reg}$) and distilling knowledge from the MT output distribution to the ST output ($\mathcal{L}_{KD}$).

Combining Foundation Models. Recent advancements in self-supervised learning and large scale training have enabled the development of powerful foundation models for both speech and text modalities. To leverage these rich representations, several works have adopted a composite architecture that initializes the model components with pre-trained weights, as depicted in Figure 2.10. Specifically, the acoustic encoder is initialized from a self-supervised speech model, such as WAV2VEC 2.0 (Baevski et al., 2020) or HUBERT (Hsu et al., 2021), or their CTC-finetuned versions, which provides robust acoustic feature extraction. The semantic encoder and the decoder are initialized from a large multilingual Machine Translation (MT) model, such as mBART-50 (Liu et al., 2020a) or NLLB (NLLB, 2024), inheriting its strong cross-lingual understanding and generation capabilities. To minimize the modality gap (§2.6.3), usually a Length Adaptor is used on

the output of the acoustic encoder. It consists of a stack of 1D strided convolutional layers that progressively down-sample the acoustic representation along the temporal dimension. Note that some works do not utilize a semantic encoder, and directly feed the acoustic representations from the length adaptor into the decoder (Li et al., 2021; Tsiamas et al., 2022b).

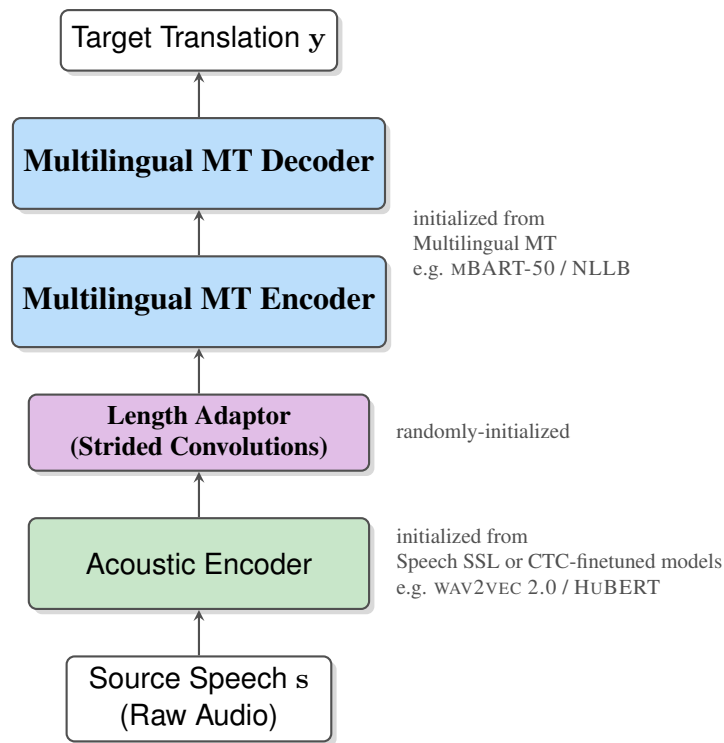


Figure 2.10: Architecture for Combining Foundation Models. The acoustic encoder is initialized from a self-supervised speech model (green), while the semantic encoder and decoder are initialized from a multilingual Machine Translation model (blue). A Length Adaptor (purple) bridges the modality gap by down-sampling the acoustic features to match the sequence length expected by the MT encoder.

2.6.5 Evaluation Metrics

Evaluating ST systems requires comparing the generated hypothesis $\hat{y}$ against a human reference y^* . The standard metrics include the following, which are directly borrowed from the field of Machine Translation.

BLEU. The *Bilingual Evaluation Understudy* (BLEU) (Papineni et al., 2002) is the most ubiquitous metric in translation. It computes the geometric mean of n-gram precisions

(typically up to $n = 4$) between the hypothesis and the reference, multiplied by a Brevity Penalty (BP) to discourage overly short translations:

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^4 w_n \log p_n \right) \quad (2.17)$$

While efficient and widely comparable, BLEU relies on exact surface-form matching. It fails to recognize synonyms or paraphrases and correlates poorly with human judgment on semantic correctness, especially for free-word-order languages. sacreBLEU (Post, 2018) offers a standardized way to report BLEU scores, making comparison across different methods more fair.

chrF. *Character n -gram F -score* metrics, chrF and chrF++ (Popović, 2015, 2017), operate at the character level rather than the word level. They compute the F -score (the harmonic mean of precision and recall) of character n -grams (typically up to $n = 6$).

$$\text{chrF}_\beta = (1 + \beta^2) \frac{\text{chrP} \cdot \text{chrR}}{\beta^2 \cdot \text{chrP} + \text{chrR}} \quad (2.18)$$

By bypassing word boundaries, chrF is significantly more robust to morphological variations and tokenization errors. It is particularly effective for agglutinative languages and has been shown to correlate better with human rankings than BLEU (Freitag et al., 2024).

COMET. *Cross-lingual Optimized Metric for Evaluation of Translation* (COMET) (Rei et al., 2020) represents a paradigm shift from overlapping-based metrics to learned, model-based metrics. It utilizes a pre-trained multilingual encoder (e.g., XLM-RoBERTa (Conneau et al., 2020)) to map the source x , hypothesis $\hat{y}$, and reference y^* into a shared embedding space. A regression head is then trained on human quality annotations (such as DA or MQM scores) to predict a quality score. COMET captures semantic similarity beyond lexical overlap, effectively penalizing critical errors while rewarding valid paraphrases. Extended variants such as XCOMET-XL (Guerreiro et al., 2023) additionally provide fine-grained error detection, and are used in our evaluations in Ch. 6.

Learned vs Surface-level Metrics. Overall, learned metrics like COMET and MetricX (Juraska et al., 2024) correlate significantly better with human annotations compared to surface-level metrics like BLEU and chrF (Freitag et al., 2024). Nevertheless, the preva-

lence of learned metrics was less evident in the early stages of this research (2022). As a result, in several evaluations (particularly in Chs. 3 and 4), we relied solely on surface-level metrics. Furthermore, aside from its well-documented disadvantages (Mathur et al., 2020a), BLEU is widely used and provides a straightforward way to compare methods via reported scores. An additional consideration is that automatic or re-segmented audio poses specific challenges for learned metrics, as their reference-based scoring can be sensitive to segment boundary mismatches (Sperber et al., 2024). Finally, learned metrics do not yet support all language pairs; consequently, in later chapters (e.g., Ch. 6), surface-level metrics are used to supplement COMET where necessary.

The Prosody Blind-Spot. Crucially, practically all standard metrics (BLEU, chrF, COMET) are *text-only*. They evaluate the lexical content of the translation but place little importance on suprasegmental features. This limitation motivates the creation of the CONTRAPROST benchmark (see Ch. 7), which specifically evaluates the preservation of paralinguistic cues in the translated text.

Part II

Audio Segmentation

3 Bridging the Segmentation Gap

This chapter is based on our work “SHAS: Approaching Optimal Segmentation for End-to-End Speech Translation”, published in **Interspeech 2022** (Tsiamas et al., 2022a).

3.1 Introduction

End-to-end (E2E) Speech Translation (ST) offers a streamlined alternative to the traditional cascade of Automatic Speech Recognition (ASR) and Machine Translation (MT) (Bérard et al., 2016; Weiss et al., 2017). By optimizing a single differentiable path from acoustic signals to foreign text, E2E models avoid the error propagation inherent in cascaded systems and reduce inference latency (Sperber and Paulik, 2020; Bentivogli et al., 2021). Yet, transitioning from controlled research environments to real-world deployment uncovers a significant challenge: processing continuous audio inputs.

3.1.1 The Segmentation Mismatch

Most E2E ST models are trained on curated datasets such as MuST-C (Di Gangi et al., 2019a), mTEDx (Salesky et al., 2021), or Europarl-ST (Iranzo-Sánchez et al., 2020). In these corpora, the source audio is manually segmented into sentence-level utterances based on transcript punctuation. Consequently, the model learns to translate well-formed acoustic sentences with clear boundaries. In contrast, real-world scenarios like live lectures, parliamentary debates, or streaming videos present continuous audio streams without explicit boundaries.

This difference creates a *segmentation mismatch* (Ansari et al., 2020; Papi et al., 2021). The model expects sentence-like units but receives an arbitrary stream that must be split before processing. The quality of this segmentation is paramount, as incorrect splits can sever linguistic dependencies (e.g., separating a subject from its verb), causing failures in the translation output.

A common solution employs Voice Activity Detection (VAD) (Sohn et al., 1999) to segment audio based on acoustic silence. While effective for ASR, which relies on local acoustic context, VAD is often insufficient for ST. Translation requires global semantic context to resolve ambiguities and reorder syntactic structures across languages. A pause in speech does not necessarily signal a semantic boundary. It often indicates hesitation, emphasis, or a breath within a linguistic unit. As a result, VAD-based segmentation tends to fragment sentences, degrading translation quality compared to the “oracle” performance of manual segmentation (Anastasopoulos et al., 2021).

3.1.2 Contribution: Supervised Hybrid Audio Segmentation

To address this challenge, we introduce Supervised Hybrid Audio Segmentation (SHAS), a method designed to align the segmentation used at inference with the manual segmentation seen during training.

We hypothesize that optimal segmentation boundaries for translation are semantic rather than purely acoustic. Therefore, we replace heuristic VADs with a classifier trained to mimic human segmentation decisions. SHAS leverages the contextualized speech representations of a pre-trained WAV2VEC 2.0 model (Baevski et al., 2020) to predict the probability that a given audio frame belongs to a valid segment. These probabilities feed into a Probabilistic Divide-and-Conquer (PDAC) algorithm, which recursively splits the audio at points of lowest probability to identify sentence boundaries while respecting user-defined length constraints.

Our main contributions are:

- **Segmentation Model.** We propose the Segmentation Frame Classifier (SFC), a lightweight module that takes as input self-supervised features from WAV2VEC 2.0 and learns to identify segmentation boundaries from labeled speech corpora, effectively transferring “oracle” segmentation knowledge to the inference pipeline.
- **Segmentation Algorithm.** We introduce the Probabilistic Divide-and-Conquer (PDAC) algorithm, a decoding strategy that utilizes the continuous probability outputs of the SFC to produce variable-length segments optimized for translation.
- **State-of-the-Art Results.** We demonstrate that SHAS outperforms standard length-

based and pause-based methods. On the MuST-C and mTEDx benchmarks, SHAS retains 95–98% of the translation quality achieved with manual segmentation, compared to 87–93% for competing methods.

- **Zero-shot capabilities.** We show that the method exhibits strong zero-shot generalization. A classifier trained on English data successfully segments unseen languages (e.g., German, Spanish) by leveraging the cross-lingual representations of the underlying WAV2VEC 2.0 model.
- **Open-sourcing.** Our trained models and methodology are available on a public repository.¹

The remainder of this chapter discusses the challenges of audio segmentation for speech translation (§3.2), details the current research landscape (§3.3), introduces the proposed SHAS methodology (§3.4), provides the experimental setup (§3.5), and finally presents and analyses our results (§3.6).

3.2 The Challenge of Audio Segmentation for Translation

To understand why audio segmentation is a critical bottleneck for Speech Translation, we must first distinguish the requirements of ST from those of Automatic Speech Recognition (ASR). ASR is largely a monotonic task, where the sequence of output characters corresponds roughly to the sequence of input acoustic frames (Sakoe and Chiba, 1978). While context helps resolve ambiguities (e.g., homophones), the correspondence is fundamentally local (Jelinek, 1976). Consequently, ASR systems are relatively robust to arbitrary segmentation. Overlapping windows or fixed-length chunks often suffice because the model can transcribe the current window with minimal dependence on the distant future.

Speech Translation, however, is inherently non-monotonic and requires global context. The translation of a word often depends on parts of the sentence that appear much later in the audio stream. For instance, translating a verb-final German sentence into English requires the model to wait until the end of the source utterance to generate the correct verb in the target. If an arbitrary segmentation point falls between the subject and the verb, the model loses the necessary context to generate a grammatically correct translation. This

¹github.com/mt-upc/SHAS/

sensitivity makes segmentation not merely a preprocessing step, but a determining factor in final translation quality.

3.3 Existing Approaches and Limitations

In the absence of a manual (or oracle) segmentation, several automatic strategies have been proposed. These can generally be categorized into length-based, pause-based, and hybrid approaches.

Length-based Segmentation. The simplest approach splits the audio stream into fixed-length chunks (e.g., every 30 seconds). While this ensures that the input fits within the memory constraints of the Transformer architecture, it completely disregards the linguistic content of the signal. Segments are almost guaranteed to cut through sentences or words, resulting in fragmented inputs that are difficult to translate.

Pause-based Segmentation. A more sophisticated approach aims to split the audio at natural pauses, typically using Voice Activity Detection (VAD) tools ([Sohn et al., 1999](#)). These systems classify audio frames as speech or non-speech based on energy thresholds or statistical models. The underlying assumption is that acoustic silence corresponds to linguistic sentence boundaries.

However, this assumption is often flawed in spontaneous speech. Speakers frequently pause for breath, hesitation, or emphasis in the middle of a sentence. A VAD-based system will eagerly split at these silences, fracturing the sentence into incomplete fragments. Conversely, in fast-paced speech, speakers may connect distinct sentences without a perceptible pause, causing the VAD to merge them into overly long sequences that exceed the model’s capacity. The issue is further reinforced by the absence of syntactic information during the segmentation. This mismatch between acoustic pauses and semantic boundaries is a primary cause of the performance degradation observed in real-world ST deployments ([Gaido et al., 2021b](#)).

Hybrid Segmentation. To mitigate the limitations of the aforementioned methods, hybrid approaches attempt to balance acoustic features with length constraints. The most prominent algorithm in this category is the Divide-and-Conquer (DAC) strategy proposed by Potapczyk and Przybysz (2020). This method recursively splits the audio at the longest detected pauses until all segments satisfy a maximum length constraint. Gaido et al. (2021b) further refined this with a streaming-compatible version.

While hybrid methods offer a significant improvement over naive VAD segmentation by avoiding splits at short hesitation pauses, they remain fundamentally limited by their reliance on energy-based features. They are essentially heuristic algorithms operating on the output of a standard VAD. If the acoustic signal does not contain a distinct energy drop at the sentence boundary, or if a hesitation pause is acoustically indistinguishable from a terminal pause, the heuristic fails. As we demonstrate in §3.6, even these sophisticated hybrid methods lag significantly behind the oracle performance of manual segmentation, highlighting the need for a method that can learn the semantic structure of speech.

3.4 The SHAS Methodology

Our proposed method, Supervised Hybrid Audio Segmentation (SHAS), approaches the segmentation problem as a two-stage process. First, we learn a probability map over the audio signal using a supervised classifier that mimics the manual segmentation of the training data. Second, we employ a decoding algorithm that utilizes these probabilities to generate variable-length segments satisfying specific length constraints. The training architecture is depicted in Figure 3.1, while the complete inference pipeline is illustrated in Figure 3.2.

3.4.1 The Segmentation Frame Classifier

The core of our approach is the Segmentation Frame Classifier (SFC), a neural network designed to act as a learnable boundary detector.

Problem Formulation and Data. Unlike standard ASR datasets which are often pre-split, ST datasets such as MuST-C (Di Gangi et al., 2019a) or mTEDx (Salesky et al., 2021)

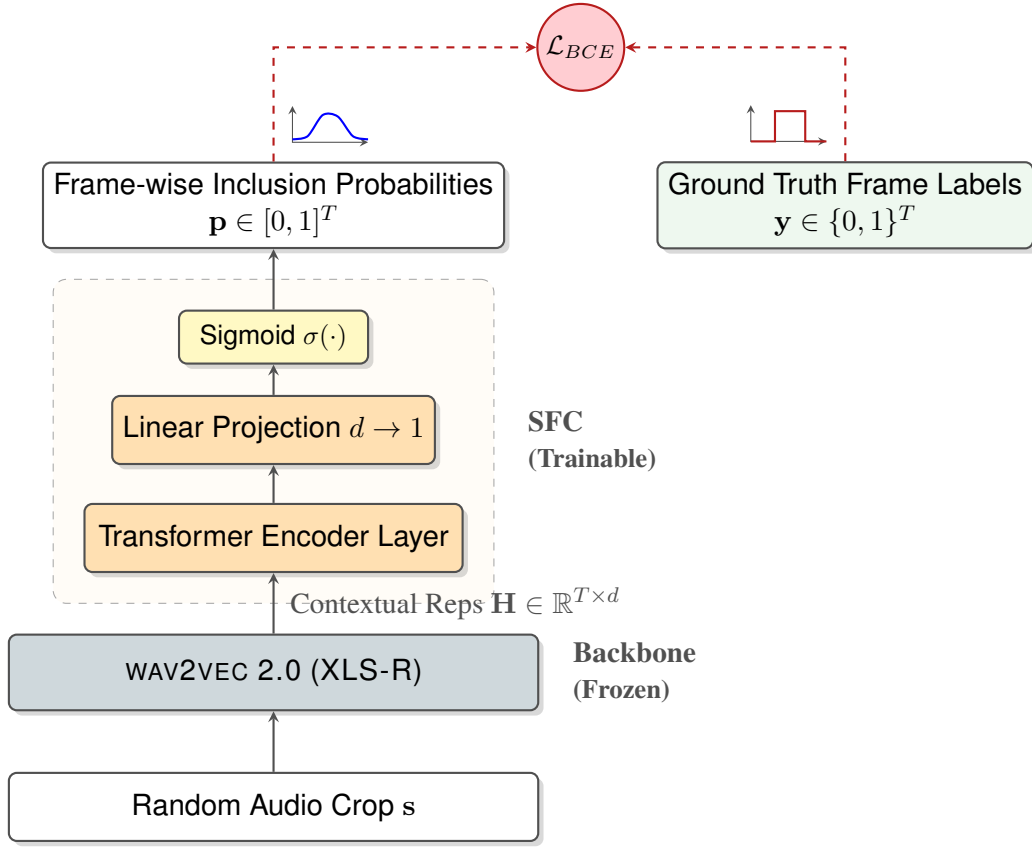


Figure 3.1: The training architecture of the Segmentation Frame Classifier (SFC). A frozen WAV2VEC 2.0 backbone extracts contextual representations, which are processed by a trainable adapter to predict frame-wise inclusion probabilities.

consist of long audio recordings (e.g., 10–20 minute talks). Each recording is associated with a set of manual segments $\mathcal{S} = \{(s_1, e_1), \dots, (s_M, e_M)\}$, where each tuple (s_i, e_i) defines the start and end timestamps of a sentence-level translation unit.²

These segments do not cover the entire duration of the recording; there are “gaps” between segments corresponding to silence, applause, or non-linguistic noise. We leverage this structure to formulate segmentation as a binary classification task. For any given frame t in the audio, we define the ground truth label y_t as:

$$y_t = \begin{cases} 1 & \text{if } \exists i : s_i \leq t \leq e_i \quad (\text{Positive: inside a segment}) \\ 0 & \text{otherwise} \quad (\text{Negative: boundary/gap}) \end{cases} \quad (3.1)$$

The goal of the SFC is to predict the probability $P(y_t = 1 | \mathbf{x})$, effectively learning to distinguish semantic speech content from the boundaries that separate them.

²The segments are mapped to the output frame resolutions of the backbone model.

Architecture. The SFC leverages the acoustic modeling capabilities of self-supervised speech representations (Figure 3.1). As input, it accepts a sequence of raw audio waveform samples $\mathbf{x}$. These are processed by a pre-trained WAV2VEC 2.0 model (Baeovski et al., 2020), specifically its multilingual variant, XLS-R (Babu et al., 2022). To maintain computational efficiency and preserve general acoustic features, we keep the parameters of the WAV2VEC 2.0 backbone frozen.

Let $f_{w2v}(\cdot)$ denote the frozen backbone. The extraction process yields a sequence of contextual representations $\mathbf{H}$:

$$\mathbf{H} = f_{w2v}(\mathbf{s}) \in \mathbb{R}^{T \times d} \quad (3.2)$$

where d is the hidden dimension of the model (e.g., 1024 for XLS-R). It is important to note that WAV2VEC 2.0 operates with a convolutional stride (typically 20ms or 320 samples at 16kHz). Therefore, T represents the number of acoustic frames, which is proportional to the number of raw samples $|\mathbf{s}|$ divided by the stride factor.

These representations are then passed to a trainable adapter module. This module consists of a Transformer encoder layer (Vaswani et al., 2017) to capture local dependencies within the window, followed by a linear projection $d \rightarrow 1$ and a sigmoid activation $\sigma(\cdot)$:

$$\mathbf{H}' = \text{TransformerEncoder}(\mathbf{H}) \quad (3.3)$$

$$\mathbf{p} = \sigma(\mathbf{H}'\mathbf{W}) \in [0, 1]^T \quad (3.4)$$

Here, $\mathbf{W} \in \mathbb{R}^{d \times 1}$ are learnable parameters, and $\mathbf{p} = (p_1, \dots, p_T)$ represents the predicted inclusion probabilities for each frame t .

Training Strategy. Training on full-length talks is computationally infeasible. Conversely, training only on pre-segmented sentences would deprive the model of negative samples (boundaries). Therefore, we employ a *random segmentation* training scheme. At each training step, we sample a random fixed-length window of duration N (e.g., 20 seconds) from the continuous recording. This window may contain partial sentences, multiple short sentences, or silence.

The model is trained to minimize the Binary Cross-Entropy (BCE) loss between the

predicted probabilities p_t and the ground truth labels y_t :

$$\mathcal{L} = -\frac{1}{T} \sum_{t=1}^T [w \cdot y_t \log p_t + (1 - y_t) \log(1 - p_t)] \quad (3.5)$$

where w is a weighting factor introduced to handle the class imbalance, as positive frames (speech) typically outnumber negative frames (boundaries) in standard talks.

3.4.2 The Probabilistic Divide-and-Conquer Algorithm

The output of the SFC is a continuous probability curve where low values indicate potential split points. To transform this curve into a valid segmentation, we introduce the Probabilistic Divide-and-Conquer (PDAC) algorithm.

Standard hybrid approaches typically apply a binary decision rule (e.g., VAD speech/non-speech) and then split at the longest detected pause (Potapczyk and Przybysz, 2020). However, this binary quantization discards valuable information about the *quality* of the potential split. A long pause might be a hesitation, whereas a shorter pause might be a clear sentence boundary.

PDAC operates directly on the continuous probability map generated by the SFC. As illustrated in Figure 3.2 (Top), we first handle arbitrary input durations via a rolling window strategy. The audio stream is processed in overlapping fixed-length windows, and predictions in overlapping regions are averaged to ensure smooth transitions.

Once the full probability curve p is obtained, the algorithm recursively identifies the optimal split point within a given segment (Figure 3.2, Bottom). We define the optimal split point not as the longest silence, but as the frame t with the global minimum probability p_t .

The Algorithm. The process, formalized in Algorithm 1, proceeds as follows:

1. Given a segment of audio, which initially corresponds to the full recording, the algorithm checks if its duration exceeds a user-defined *max-segment-length* (ℓ_{max}).

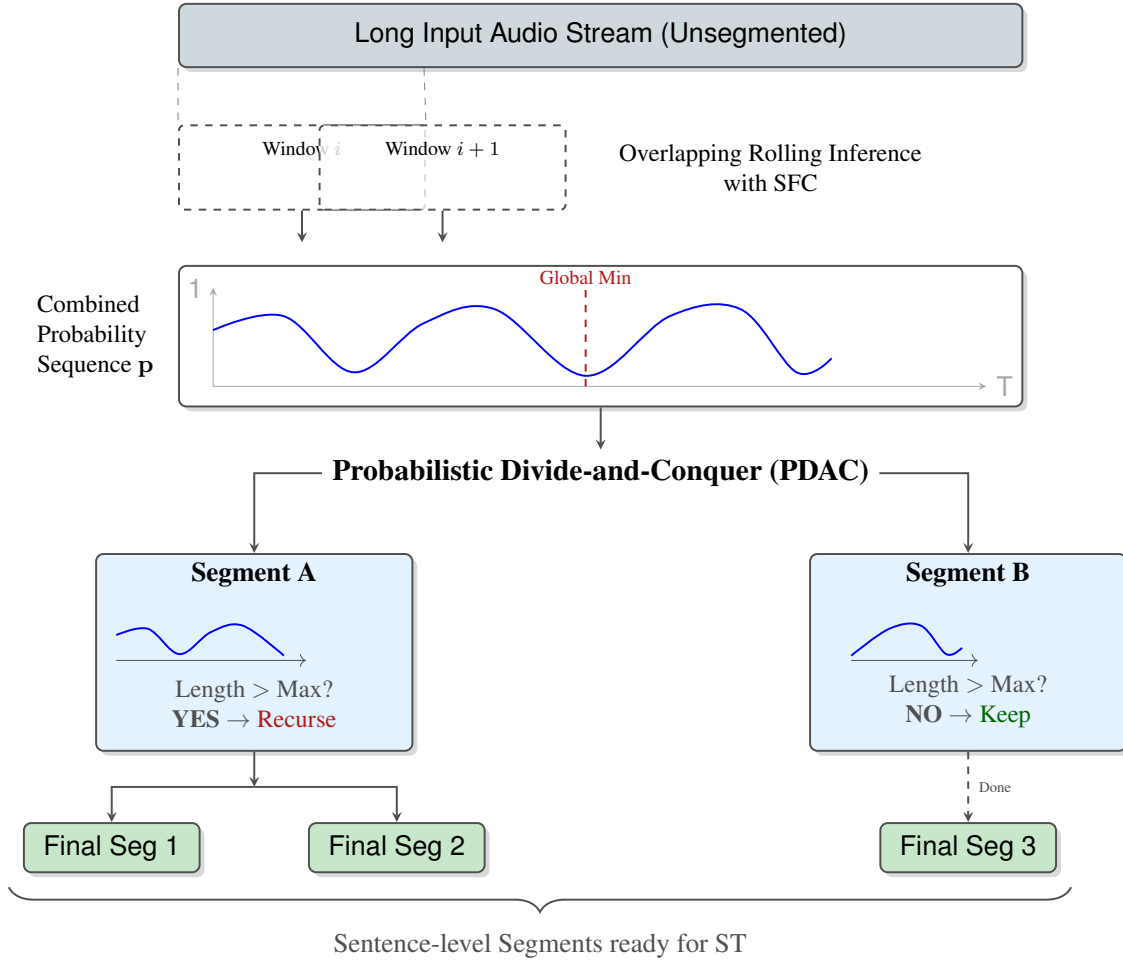


Figure 3.2: The SHAS Inference Pipeline. **Top:** The Rolling Window strategy merges probabilities from overlapping windows. **Bottom:** The PDAC algorithm recursively splits the probability curve at global minima to form valid segments.

2. If the segment is too long, the algorithm searches for the frame with the lowest probability ($\text{argmin } P$).
3. To prevent creating unusable fragments, we enforce a *min-segment-length* (ℓ_{min}) constraint. The split point must result in two new segments that are both longer than ℓ_{min} .
4. If a valid split point is found, the segment is divided, and the algorithm recursively processes the left and right sub-segments.
5. Finally, the boundaries of the resulting segments are trimmed using a probability threshold (p_{thr}) to remove non-speech frames at the edges.

This approach allows the segmentation policy to be dynamic. In regions of high uncertainty (all probabilities are relatively high), the algorithm effectively looks for the “least bad”

place to split, minimizing the potential damage to the translation, whereas standard VAD might fail to trigger a split entirely.

Algorithm 1 Probabilistic Divide-and-Conquer (PDAC)

Require: Frame probabilities P , max length ℓ_{max} , min length ℓ_{min} , threshold p_{thr}

```

1: procedure RECURSIVESPLIT( $start, end$ )
2:   if ( $end - start$ ) <  $\ell_{max}$  then
3:     return [( $start, end$ )]
4:   end if
5:    $candidates \leftarrow \text{SortIndicesByProbability}(P[start : end])$ 
6:   for  $idx$  in  $candidates$  do
7:      $split \leftarrow start + idx$ 
8:      $left \leftarrow (start, split)$ 
9:      $right \leftarrow (split, end)$ 
10:    if  $\text{Length}(left) > \ell_{min}$  and  $\text{Length}(right) > \ell_{min}$  then
11:       $L_{segs} \leftarrow \text{RECURSIVESPLIT}(left)$ 
12:       $R_{segs} \leftarrow \text{RECURSIVESPLIT}(right)$ 
13:      return  $L_{segs} + R_{segs}$ 
14:    end if
15:  end for
16:  return [( $start, end$ )] ▷ No valid split found
17: end procedure
18: procedure PDAC( $P, \ell_{max}, \ell_{min}, p_{thr}$ )
19:    $Segments \leftarrow \text{RECURSIVESPLIT}(0, \text{Length}(P))$ 
20:    $FinalSegments \leftarrow []$ 
21:   for ( $s, e$ ) in  $Segments$  do
22:     ( $s', e'$ )  $\leftarrow \text{Trim}(P, s, e, p_{thr})$  ▷ Trim edges below threshold
23:     Append ( $s', e'$ ) to  $FinalSegments$ 
24:   end for
25:   return  $FinalSegments$ 
26: end procedure

```

3.5 Experimental Setup

We evaluate the proposed method on five language directions, covering both high-resource and medium-resource scenarios, and assess its robustness across different domains.

3.5.1 Data

Our experiments cover translations from English (eng) to German (deu) and from Spanish (spa), French (fra), Italian (ita), and Portuguese (por) into English. We utilize the following datasets:

- **MuST-C** (Di Gangi et al., 2019a): We use the eng-deu part of this corpus. The English audio is used to train the English Segmentation Frame Classifier (SFC). We use the dev split for development and the tst-COMMON split for testing.
- **mTEDx** (Salesky et al., 2021): For the non-English source languages (spa, fra, ita, por), we utilize the ASR splits of the mTEDx corpus to train the respective SFCs. The ST valid and test splits are used for development and testing.
- **Europarl-ST** (Iranzo-Sánchez et al., 2020): We use the eng-deu test set of this corpus to evaluate cross-domain generalization, applying the classifier trained on MuST-C (TED talks) to parliamentary debates without any fine-tuning.

All audio data were converted to mono and sampled at 16kHz.

3.5.2 Segmentation Frame Classifiers

We trained five monolingual SFC models (eng, spa, fra, ita, por) and one multilingual model covering all source languages. To handle the lack of explicit negative examples (boundaries) in standard datasets, we trained the models using random audio crops of $N = 320,000$ samples (20 seconds). This random cropping strategy ensures the model encounters both valid speech segments and boundaries during training.

Architecture. We utilized the pre-trained XLS-R model (Babu et al., 2022) as the backbone, and specifically the 300M-parameter version available on the Transformers library.³ Based on preliminary experiments on the MuST-C devset (see ablation §3.6.6), we extract representations from the 14th layer of XLS-R, as this layer was found to contain the most optimal features for the segmentation task, outperforming the final layer. The extracted features are processed by a lightweight adapter consisting of a single Transformer encoder layer with a model dimension of 1024, a feed-forward dimension of 2048, 8

³huggingface.co/facebook/wav2vec2-xls-r-300m

attention heads, pre-layer normalization, and GELU activation (Hendrycks and Gimpel, 2023). A final linear layer with a sigmoid activation maps the representations to frame-level probabilities. Crucially, the parameters of the XLS-R backbone are kept frozen, and only the adapter (approximately 8.4M parameters) is trained.

Optimization. Models were trained for 8 epochs using the AdamW optimizer (Loshchilov and Hutter, 2019) with a batch size of 14. We used a dropout of 0.1 in the transformer adapter layer. We employed a learning rate of $2.5 \cdot 10^{-4}$ with cosine annealing decay. To address class imbalance that stems from frames containing speech significantly outnumber boundary frames, we weighted the binary cross-entropy loss towards the negative class. For inference, we perform a rolling window analysis with a stride of 10 seconds. To minimize context loss at the window edges, we average the probabilities from two overlapping windows.

3.5.3 Speech Translation Models

To assess the impact of segmentation on translation quality, we used the pretrained Joint Speech-to-Text models (Figure 2.9) from FAIRSEQ (Tang et al., 2021a; Wang et al., 2020b). These models represent strong, standard baselines for E2E ST. We used two different configurations:

1. **English-German Model:** Trained on MuST-C eng-deu. It features a Transformer encoder-decoder structure (12 encoder layers, 6 decoder layers) with a model dimension of 512. The input consists of 80-dimensional log mel-filterbank features with a 10ms stride. The vocabulary size is 10,000 unigram subwords.⁴
2. **Multilingual Model:** Trained on the mTEDx dataset, including the directions spa, fra, ita, and por to English. This is a larger model (24 encoder layers, 12 decoder layers) with a dimension of 768. Unlike the eng-deu model, it processes raw audio waveforms directly. The vocabulary consists of 64,000 subwords.⁵

During inference, we strictly use the speech modality (no auxiliary text input) and decode with a beam search of size 5.

⁴github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/ende-mustc.md

⁵github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/iwslt2021.md

3.5.4 Baselines and Evaluation

We compare SHAS against the following segmentation strategies:

- **Length-based:** Splits audio into fixed-length segments determined by a max-length parameter.
- **Pause-based (VAD):** Uses WebRTC VAD⁶ to split audio at detected silences. We tune the frame length (10, 20, 30ms) and aggressiveness mode (1, 2, 3).
- **Hybrid Approaches:** We compare against the Hybrid VAD-DAC and Hybrid VAD-STRM methods proposed by [Gaido et al. \(2021b\)](#), as well as the Hybrid W2V-DAC ([Gállego et al., 2021](#)), which uses a fine-tuned WAV2VEC 2.0 model as a pause detector.

Hyperparameter Tuning. For all methods, parameters were tuned on the respective development sets to maximize BLEU scores. For SHAS and the hybrid/length-based baselines, the *max-segment-length* was tuned in the range of $[4, 40]$ seconds. For SHAS, we set the *min-segment-length* to 0.2 seconds and the trimming probability threshold to 0.5.

Metrics. Since different segmentation methods produce different sentence boundaries, the resulting translations cannot be compared directly line-by-line with the reference. To address this, we realign the generated translations with the reference text using `mwerSegmenter` ([Matusov et al., 2005](#)), which is standard practice in IWSLT evaluation campaigns ([Ansari et al., 2020](#); [Anastasopoulos et al., 2021](#)). We then report BLEU scores ([Papineni et al., 2002](#)) computed with `sacreBLEU` ([Post, 2018](#)). To quantify the effectiveness of the segmentation, we also report the percentage of the “oracle” BLEU score (obtained using manual segmentation) that is retained by the automatic method. Overall, when comparing different segmentation methods, we use the same Speech Translation model, thus any differences in translation quality are directly stemming from the differences in segmentation quality. As this chapter was published in 2022, we report BLEU as the primary evaluation metric, following the standard practice at the time; learned metrics such as COMET were less prevalent then. See §2.6.5 for a broader discussion of metric choices and their limitations.

⁶github.com/wiseman/py-webrtcvad

3.6 Results and Analysis

3.6.1 Approaching the Oracle

We first evaluate the effectiveness of SHAS in bridging the gap between automatic and manual segmentation on the MuST-C and mTEDx benchmarks. Table 3.1 presents the BLEU scores obtained by translating segments generated by SHAS compared to existing baselines. To quantify the quality of the segmentation independently of the underlying translation model, we report the “Recovery Rate”, which is the percentage of the BLEU score obtained with the manual (oracle) segmentation that is retained by the automatic method.

The results demonstrate that SHAS consistently produces high-quality segmentations, allowing the ST models to retain more than 95% of the manual BLEU score across all tested language directions, with an average retention of 96%. In comparison to standard baselines, we observe substantial improvements. SHAS outperforms the widely used pause-based (VAD) approach by 4.5 BLEU points on average, and surpasses the Hybrid VAD methods (Gaido et al., 2021b) by approximately 2 BLEU points. Notably, even when compared to the strongest baseline, the Hybrid W2V-DAC (Gállego et al., 2021) which also utilizes WAV2VEC 2.0 features for pause detection, SHAS achieves an improvement of 1.2 BLEU points, effectively closing the gap to the oracle performance by an additional 4%.

We also observe that the traditional pause-based method significantly underperforms in this multilingual setting, retaining only 81% of the manual BLEU on average. Surprisingly, a naive length-based segmentation often yields better results (85%) than VAD. This underperformance is likely due to the VAD’s reliance on energy thresholds which may not generalize well across different languages and recording conditions, whereas the length-based approach guarantees segments that fit the model’s memory constraints. These findings highlight that semantic continuity, which SHAS aims to preserve, is more critical for translation quality than merely splitting at acoustic silences.

Segm. Method	eng-deu	esp-eng	fra-eng	ita-eng	por-eng	Average
Manual (Oracle)	27.0	31.9	36.7	27.2	34.9	31.5
Length-based (fixed)	22.3 (83%)	27.7 (87%)	30.6 (83%)	23.7 (87%)	30.2 (87%)	26.9 (85%)
Pause-based (VAD)	22.8 (84%)	27.0 (85%)	30.0 (82%)	21.8 (80%)	26.6 (76%)	25.6 (81%)
Hybrid VAD-DAC	24.2 (90%)	29.4 (92%)	31.9 (87%)	24.4 (90%)	30.9 (89%)	28.1 (89%)
Hybrid VAD-STRM	23.8 (88%)	29.5 (93%)	31.8 (87%)	24.7 (91%)	31.2 (89%)	28.2 (89%)
Hybrid W2V-DAC	24.5 (91%)	29.7 (93%)	33.5 (91%)	24.8 (91%)	32.5 (93%)	29.0 (92%)
SHAS (Monolingual)	25.7 (95%)	30.5 (96%)	35.1 (96%)	26.4 (97%)	33.2 (95%)	30.2 (96%)
SHAS (Multilingual)	25.6 (95%)	30.8 (97%)	35.3 (96%)	26.6 (98%)	33.5 (96%)	30.4 (96%)

Table 3.1: BLEU scores of SHAS compared to manual segmentation and other methods on MuST-C and mTEDx test sets. The numbers in parentheses indicate the Recovery Rate (percentage of Manual BLEU retained). The translations are carried out for each language pair by the same model, thus any differences in translation quality are stemming directly from the segmentation method.

3.6.2 Multilingual Generalization

While the results discussed above utilize monolingual Segmentation Frame Classifiers (SFC) trained specifically for each source language, we also investigated the efficacy of a single Multilingual SFC. This model was trained on the concatenated data of all source languages. As shown in the last row of Table 3.1, this unified model achieves additional improvements for most language pairs. For instance, on the Italian-to-English direction (*ita-eng*), the multilingual SHAS reduces the gap with the manual segmentation to just 0.5 BLEU. This indicates that the segmentation boundaries learned by the model share common acoustic-semantic characteristics across languages, allowing a single model to effectively handle multiple source languages without performance degradation. The only exception is the high-resource *eng-deu* pair, where the specialized monolingual model performs marginally better.

3.6.3 Zero-Shot Cross-Lingual Transfer

Given that the SFC is built on top of the cross-lingual XLS-R representations, we hypothesized that it might exhibit zero-shot generalization capabilities. To test this, we applied the monolingual classifiers to languages they were not trained on (e.g., using an English-trained classifier to segment Spanish audio). Table 3.2 presents the matrix of zero-shot performance.

The results reveal a remarkably high degree of cross-lingual transfer. For most languages,

Trained / Tested	eng-deu	esp-eng	fra-eng	ita-eng	por-eng
eng	<u>25.7</u>	22.3	33.6	14.3	26.1
esp	21.7	<u>30.5</u>	32.8	25.8	33.0
fra	23.7	30.3	35.1	26.4	32.6
ita	22.2	30.4	34.4	<u>26.4</u>	32.8
por	23.0	30.8	34.1	26.2	<u>33.2</u>

Table 3.2: Zero-shot BLEU scores on MuST-C and mTEDx test sets. Rows represent the language the SFC was trained on; columns represent the language which is evaluated. Underlined is the supervised performance (diagonal). **Bold** is the best performance for each evaluated language (columns).

the zero-shot performance is comparable to the supervised performance. For example, the Portuguese (por) classifier achieves the best result on the Spanish test set (30.8 BLEU), slightly outperforming the supervised Spanish classifier itself. This phenomenon is a direct consequence of the underlying XLS-R vector space, which aligns acoustic representations across languages. Training a simple linear adapter on top of these frozen representations preserves this alignment, enabling the segmentation policy to transfer to unseen languages.

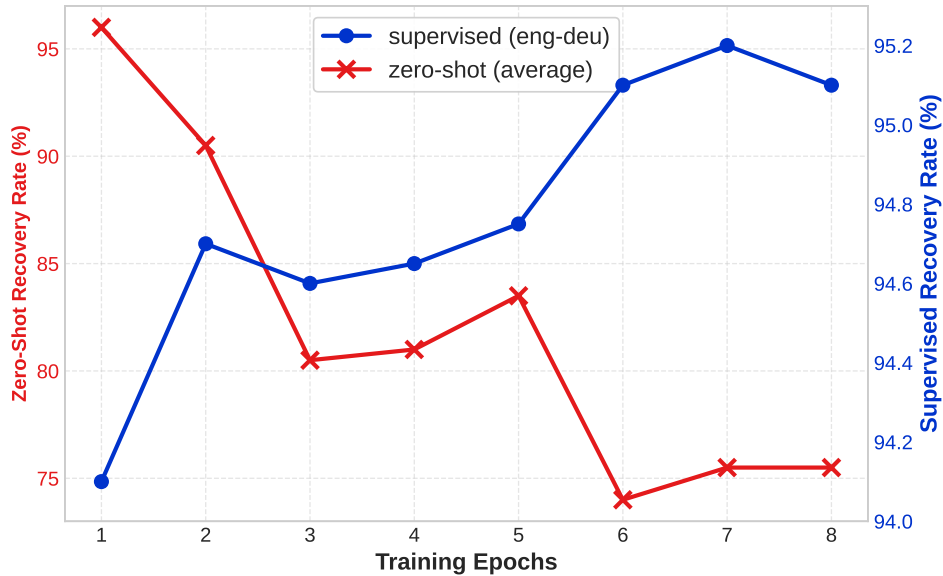


Figure 3.3: The trade-off between specialization and generalization. As the English SFC trains (x-axis: epochs), its performance on the supervised task (eng-deu) improves, while its zero-shot performance on other languages (average) degrades.

However, we observe a distinct behavior with the English classifier. While it performs optimally on English data, it generalizes poorly to other languages (e.g., achieving only 14.3 BLEU on Italian). This is likely due to the significantly larger size of the English training corpus (450 hours vs. 100-200 hours for others), which leads the model to over-specialize

and suffer from catastrophic forgetting of cross-lingual alignment (Pfeiffer et al., 2020; Kumar et al., 2022). Figure 3.3 illustrates this trade-off: in the early epochs of training, the English classifier retains high zero-shot performance, but as training progresses and the supervised performance peaks, the zero-shot capability sharply declines. This suggests that for zero-shot applications, early stopping or training on less data might be beneficial to preserve the cross-lingual generalization of the pre-trained embeddings.

3.6.4 Domain Robustness

Finally, we assess the robustness of SHAS to domain shifts. We applied the SFC trained on MuST-C (TED talks) to the Europarl-ST benchmark (parliamentary debates) without any fine-tuning. Table 3.3 presents these cross-domain results.

Method	BLEU	Recovery Rate
Manual (Oracle)	28.8	100%
Length-based (fixed)	22.4	80%
Pause-based (VAD)	18.6	67%
Hybrid VAD-STRM	24.1	87%
Hybrid W2V-DAC	24.9	90%
SHAS (Monolingual)	26.4	95%
SHAS (Multilingual)	26.2	94%

Table 3.3: Cross-domain robustness on Europarl-ST eng-deu test. The SHAS model was trained on MuST-C (TED talks) and applied zero-shot to Europarl-ST.

SHAS demonstrates strong cross-domain transfer, achieving a BLEU score of 26.4 and retaining 95% of the manual segmentation’s quality. In contrast, the best existing baseline achieves only 90% retention, and the standard VAD approach drops significantly to 67%. This indicates that the semantic boundaries learned by SHAS are not specific to the prosodic style of TED talks but capture fundamental properties of speech structure that hold across different speaking styles and domains.

3.6.5 Sensitivity to Segment Length

A critical hyperparameter in most segmentation algorithms is the *max-segment-length* (max), which dictates the maximum duration of any generated audio segment. In standard

hybrid approaches, this parameter requires careful tuning, as the optimal length often depends on the capacity of the downstream ST model.

We analyzed the sensitivity of SHAS and baseline methods to this parameter by evaluating translation quality across a range of ℓ_{max} values (from 8 to 40 seconds). We tested this using four different ST models of varying quality⁷ to simulate different resource conditions.

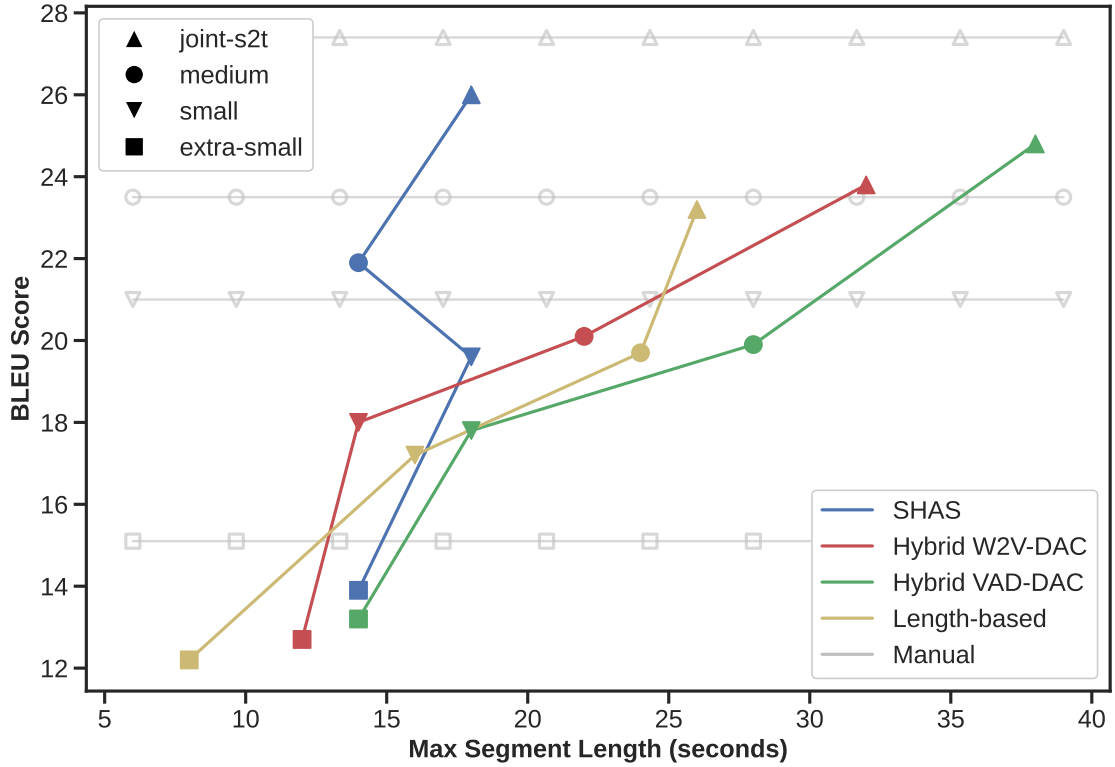


Figure 3.4: Impact of the *max-segment-length* parameter (ℓ_{max}) on translation quality. Different segmentation methods evaluated on a range of ℓ_{max} values (x-axis), with four different Speech Translation models, of varying quality (weakest: extra-small, strongest: joint-s2t). BLEU scores on MuST-C eng-deu tst-COMMON.

The results, visualized in Figure 3.4, reveal a divergence in behavior. For existing hybrid and length-based methods, there is a clear trend: higher-quality ST models require increasingly longer segments to reach their peak performance. This implies that these methods frequently create improper splits, and stronger models need larger context windows to recover from these errors.

In contrast, SHAS exhibits remarkable stability. The optimal value for the ℓ_{max} parameter consistently falls in the range of 14–18 seconds, regardless of whether the downstream

⁷The extra-small, small and medium standard S2T-TRANSFORMER (§2.6.4) and the Joint S2T models used in the main experiments (§3.5)

ST model is small or large. This indicates that SHAS effectively identifies natural semantic boundaries that match the expected sentence granularity of the translation models. Consequently, SHAS eliminates the need for extensive hyperparameter tuning on held-out development sets, making it a robust solution for diverse deployment scenarios.

3.6.6 Ablations

We investigate the importance of the Segmentation Frame Classifier’s (SFC) depth and the trade-offs between offline and online processing. Table 3.4 summarizes these experiments.

Method	Full Audio	Train Params	Infer Params	BLEU
SHAS (Standard Configuration)	✓	8.4 M	208.5 M	25.7 (95.1%)
w/ PSTRM Algorithm	✗	8.4 M	208.5 M	25.6 (94.7%)
w/ Finetuned XLS-R Feature Extractor	✓	25.7 M	202.1 M	25.3 (93.6%)
w/ 0 Transformer layers in SFC	✓	0.0 M	218.9 M	24.9 (92.2%)
w/ 2 Transformer layers in SFC	✓	16.8 M	218.9 M	25.7 (95.2%)

Table 3.4: Ablation study on architectural choices and inference constraints for MuST-C engine. We report BLEU and the percentage of manual BLEU retained. The standard configuration of SHAS uses the PDAC algorithm, takes as input the representations of the 14th of a frozen XLS-R, and utilizes 1 Transformer layer in the SFC.

Transformer Layers in SFC. Our standard configuration (Full SHAS) uses a single Transformer layer in the SFC and operates in an offline mode, processing the full audio file. We observe that removing the Transformer layer and relying solely on a linear projection of the WAV2VEC 2.0 features results in a performance drop of 0.8 BLEU (from 25.7 to 24.9). This confirms that the local context aggregation provided by the self-attention mechanism is crucial for identifying semantic boundaries. Interestingly, increasing the complexity by adding a second Transformer layer yields negligible gains, suggesting that a single layer is sufficient for this task.

Segmentation Algorithm. We also evaluate a streaming-compatible version of our method (PSTRM). Instead of the Divide-and-Conquer algorithm which requires the full audio duration, this version uses the Probabilistic Streaming algorithm (Gaido et al., 2021b) on the probabilities predicted by SHAS. The results show a minimal degradation of only

0.1 BLEU compared to the offline approach. This demonstrates that SHAS is highly suitable for online applications where low latency is required.

Compact Backbone. Finally, we explored a “compact” version of the model where we use the output of the convolutional feature extractor rather than the deep Transformer layers of XLS-R, and additionally fine-tune these convolutional parameters. While this reduces the inference parameter count significantly (from 200M frozen parameters to essentially just the feature extractor), it lags by 0.4 BLEU behind the full model. Nevertheless, it remains competitive with other hybrid methods, offering a viable alternative for highly resource-constrained environments.

Optimal Feature Selection. The standard SHAS configuration utilizes representations from the 14th layer of the frozen XLS-R model. To arrive at this choice, we conducted an experiment to determine which layer of the backbone provides the most informative features for segmentation. We trained separate classifiers on top of every layer of the backbone (from the feature extractor (0th layer) output up to the last 24th layer) and evaluated their performance on the MuST-C eng-deu development set. As illustrated in Figure 3.5, we observe a distinct bell-shaped trend. The performance is suboptimal when using the very early layers (low contextual information). However, the performance also degrades significantly when using the final layers, with the last layer retaining only about 89% of the manual BLEU. This suggests that the final layers of self-supervised models are overly specialized towards the pre-training objective (masked token prediction) and may lose some of the structural acoustic information necessary for boundary detection. The “middle” layers (11–20) consistently perform best, with the 14th layer achieving the global maximum. This aligns with findings in other downstream tasks, such as unsupervised speech recognition, where middle layers often yield superior representations (Baevski et al., 2021).

3.7 Summary and Conclusions

In this chapter, we have addressed the critical challenge of processing continuous audio streams in End-to-End Speech Translation. We argued that the *segmentation mismatch*—the discrepancy between the well-formed sentence-level utterances seen during training and

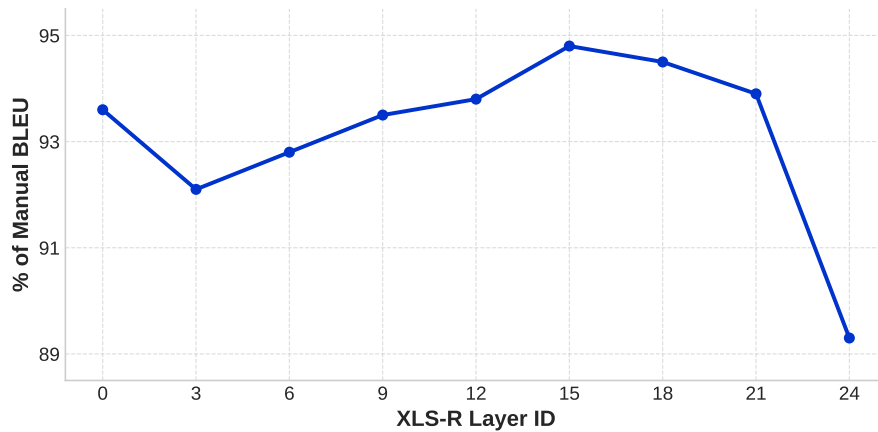


Figure 3.5: Performance of SHAS using input representations from different layers of the pre-trained XLS-R model. Measured in percentage of manual BLEU score retained in MuST-C eng-deu development set. Layer ID 0 is the output of the feature extractor.

the unsegmented streams encountered at inference—is a primary cause of performance degradation in real-world deployments. Standard approaches relying on acoustic pauses (VAD) often fail to capture the semantic structure of speech, leading to fragmented inputs that disrupt the translation process.

We proposed Supervised Hybrid Audio Segmentation (SHAS), a learnable segmentation model that bridges this gap. By training a lightweight classifier on top of self-supervised speech representations, SHAS learns to mimic the semantic segmentation decisions made by humans. Our experimental results on MuST-C and mTEDx demonstrate that this approach is highly effective, retaining nearly 96% of the translation quality achieved with manual segmentation and significantly outperforming traditional heuristic baselines. Furthermore, we showed that the method exhibits strong robustness: it generalizes to unseen domains without fine-tuning and transfers effectively to new languages in a zero-shot manner, leveraging the cross-lingual alignment of the underlying WAV2VEC 2.0 encoder.

While SHAS effectively solves the segmentation problem at *inference* time by forcing the input to resemble the training data, the dependency on a fixed segmentation strategy suggests a complementary question: can we instead modify the *training* data to be robust to segmentation variance? The probabilistic nature of the SHAS classifier provides not just a single optimal segmentation, but a distribution of valid boundaries. In Ch. 4 we explore how this flexibility can be utilized during training. We introduce SEG AUGMENT, a method that leverages dynamic re-segmentation as a data augmentation strategy, creating

diverse training views that further close the gap between training and inference conditions while maximizing data utility in low-resource settings.

4 Turning Segmentation into Augmentation

This chapter is based on our work “SEGAUGMENT: Maximizing the Utility of Speech Translation Data with Segmentation-based Augmentations”, published in **EMNLP Findings 2023** (Tsiamas et al., 2023a).

4.1 Introduction

In the previous chapter (Ch. 3), we addressed the segmentation problem at *inference* time. We showed that by using a Supervised Hybrid Audio Segmentation (SHAS) model, we could process continuous audio streams with minimal loss in translation quality. However, relying solely on inference-time solutions leaves a fundamental inefficiency in how we train End-to-End (E2E) Speech Translation (ST) models.

A major bottleneck for E2E ST is the scarcity of paired audio-text data (Sperber and Paulik, 2020). To train these models, we typically rely on corpora such as MuST-C (Di Gangi et al., 2019a), mTEDx (Salesky et al., 2021), or Europarl-ST (Iranzo-Sánchez et al., 2020). These datasets are created by manually segmenting long audio recordings (like TED talks) into sentence-level utterances based on transcript punctuation. While this provides clean training data, it results in a “static” view of the speech, in which a given sentence is always presented with the exact same acoustic boundaries and context.

This static segmentation creates two major limitations. First, it restricts the amount of available training examples, as well as their diversity, which is critical given the data scarcity issue in ST (Sperber and Paulik, 2020). Second, it creates a mismatch between training and deployment (Ansari et al., 2020; Anastasopoulos et al., 2021; Papi et al., 2021). During training, the model sees perfectly segmented sentences. During inference, even with a strong segmentation model like SHAS, the boundaries are automatic and may not match the linguistic pauses perfectly. As we observed in §3.6, this mismatch can still lead

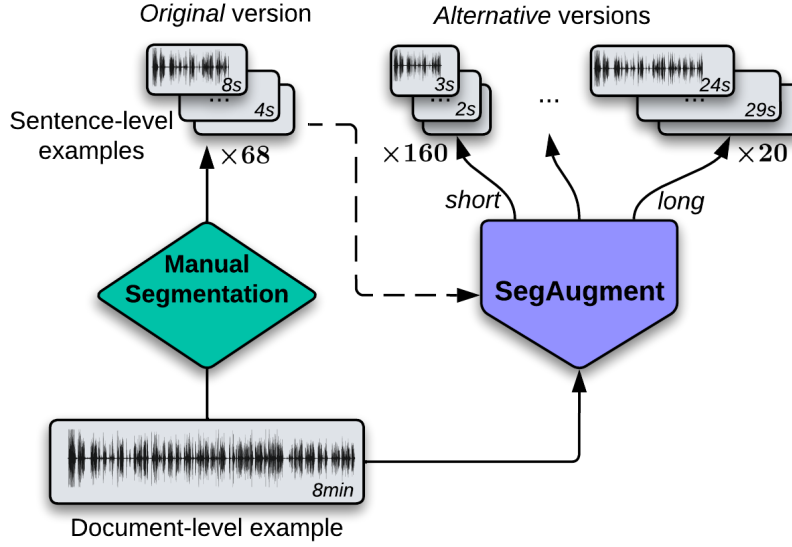


Figure 4.1: Conceptual comparison between standard data training and SEGAUGMENT. While standard training relies on a single, fixed segmentation of the document, SEGAUGMENT generates multiple alternative versions with varying length constraints, effectively multiplying the available training data and diversity.

to performance degradation, where ST inference on the automatically segmented test set cannot match the performance of the ST inference on the manually segmented one.

In this chapter, we propose to turn the segmentation problem into a data augmentation opportunity. We introduce SEGAUGMENT, a strategy that moves the segmentation process from a fixed preprocessing step to a dynamic augmentation pipeline (Figure 4.1). Instead of training on a single version of the data, we use the SHAS method developed in Ch. 3 to re-segment the original document-level speech with varying length constraints (e.g., short, medium, and long segments). We then recover the corresponding text through audio-text forced alignment (McAuliffe et al., 2017) and machine translation, thus creating a diverse dataset in which segments appear in multiple contexts.

The contributions of this chapter are as follows:

- We propose SEGAUGMENT, a novel data augmentation method for ST that leverages audio re-segmentation to multiply and diversify the training data.
- We demonstrate that this augmentation method yields consistent improvements across multiple benchmarks and scenarios. Specifically, it improves translation quality for both standard and state-of-the-art methods in MuST-C, with gains being even larger when applied in the low-resource pairs of mTEDx, and even observing

minor improvements in CoVoST-2, even though the original data are sentence-like and not based on documents.

- We provide concrete evidence that our method solves the training-inference segmentation mismatch, enabling ST models to retain 100% of the manual segmentation translation quality when evaluated on an automatic segmentation.
- We analyze the potential reasons behind the effectiveness of the proposed method, focusing on (a) contextual diversity, (b) positional diversity, (c) length specialization, and (d) knowledge distillation.
- We open-source our code for implementing the SEGAUGMENT pipeline, and all synthetic data generated with it.¹

The rest of this chapter is structured as follows. §4.2 presents the existing approaches and their limitations. §4.3 provides the background on the manual segmentation process. §4.4 details the three-step pipeline used to generate the augmented data. §4.5 describes the experimental setup, including the datasets and baselines. §4.6 presents the main results of our method showing its broad benefits and applicability. §4.7 focuses on how our method solves the training-inference segmentation mismatch. Finally, §4.8 analyzes the mechanisms behind the effectiveness of SEGAUGMENT.

4.2 Related Work: Augmentation and Segmentation

Signal-level Augmentation. Data augmentation is a standard technique for improving the generalization of speech models. The most prominent approach in this category is SpecAugment (Park et al., 2019), which operates directly on the acoustic features (spectrograms) by applying masking to blocks of frequency and time channels. WavAugment (Kharitonov et al., 2021) is a similar method, but operates on the time-domain of waveforms, by introducing effects like pitch, tempo, and echo. These methods have been successfully adapted for ST (Bahar et al., 2019; Di Gangi et al., 2019b; Gállego et al., 2021), demonstrating consistent improvements. However, they are limited to signal-level modifications and do not alter the linguistic content or structural composition of the training examples. Our proposed SEGAUGMENT is orthogonal and complementary to such tech-

¹github.com/mt-upc/SegAugment

niques, where instead of altering the speech features, we generate new training instances by modifying the segmentation points, effectively providing the model with diverse structural views of the same underlying data.

Synthetic Data from External Sources. A common strategy to address the data scarcity in ST is to leverage large amounts of external data to generate synthetic ST pairs. For instance, creating pseudo-labels by translating the transcriptions of ASR datasets (Jia et al., 2019; Pino et al., 2019) or by performing Text-to-Speech (TTS) synthesis on MT corpora (Jia et al., 2019). While effective, these methods rely heavily on the availability of large-scale external datasets. In contrast, SEGAUGMENT generates synthetic data *internally*, maximizing the utility of the existing ST dataset without requiring any additional external speech or text resources.

Re-combining Training Data. A similar line of work to ours aims to generate synthetic examples by cropping or merging the original ones, with sub-sequence sampling for ASR (Nguyen et al., 2020), and concatenation for MT (Nguyen et al., 2021; Wu et al., 2021; Kondo et al., 2021), and for speech tasks (Lam et al., 2023). Our approach is more general than the aforementioned ones, since it segments documents in different contextually valid points, thus providing access to a greater number of synthetic examples. Furthermore, Lam et al. (2022) proposed to recombine training data in a linguistically-motivated way. They first sample pivot tokens, retrieve possible continuations from a suffix memory, combine them to obtain new speech-transcription pairs, and finally use an MT model to generate the translations. Our method is similar since it also leverages audio alignments and MT, but instead of mixing speech, it segments at alternative points.

Robustness to Inference Segmentation. The mismatch between training on manual segments and inferring on automatic segments is a well-known challenge (Ansari et al., 2020). Previous research has addressed this by re-segmenting the training data in random points (Gaido et al., 2020b; Papi et al., 2021). Our method provides a more robust solution, by re-segmenting not randomly, but by using a sophisticated segmentation model. A different line of works aims to reduce the training-inference segmentation gap by utilizing context-aware ST models (Zhang et al., 2021a; Amrhein and Haddow, 2022), which are inherently more robust to segmentation errors because they can access information beyond the current sentence boundary. While effective, these approaches require specialized

architectures or training regimes to encode context. Our approach aims to solve it within classic sentence-level ST models, without requiring expensive context-aware modeling.

4.3 Background: The Limits of Static Segmentation

Most ST corpora, such as MuST-C (Di Gangi et al., 2019a), Europarl-ST (Iranzo-Sánchez et al., 2020), and mTEDx (Salesky et al., 2021), are natively document-level, consisting of full audio documents (e.g., TED talks, parliamentary sessions) rather than collections of isolated sentences. To be utilized for standard ST model training, these documents are converted into sentence-level examples via a process of *manual segmentation* (Figure 4.2).

A document-level speech translation corpus $\mathcal{D}^{\text{doc}}$ is comprised of n triplets representing the speech wave $\mathbf{S}$, the transcription $\mathbf{X}$, and the translation $\mathbf{Y}$ for each document:

$$\mathcal{D}^{\text{doc}} = \{\mathbf{S}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Y}^{(i)}\}_{i=1}^n \quad (4.1)$$

To create a useful resource for sentence-level ST, the document-level corpus $\mathcal{D}^{\text{doc}}$ is processed into a sentence-level corpus $\mathcal{D}^{\text{sent}}$ containing $m = \sum_{i=1}^n m_i$ examples, where m_i is the number of sentence-level examples extracted from the i -th document. Specifically:

$$\mathcal{D}^{\text{sent}} = \left\{ \left\{ \mathbf{s}^{(i,j)}, \mathbf{x}^{(i,j)}, \mathbf{y}^{(i,j)} \right\}_{j=1}^{m_i} \right\}_{i=1}^n \quad (4.2)$$

Here, $\mathbf{s}^{(i,j)}$ denotes the sentence-level source speech for the j -th segment of the i -th document, while $\mathbf{x}^{(i,j)}$ and $\mathbf{y}^{(i,j)}$ denote the corresponding transcriptions and translations. The corpus $\mathcal{D}^{\text{sent}}$ is obtained via the manual segmentation process, where the document-level transcription and translation are split based on strong punctuation and aligned using cross-lingual sentence alignment (Braune and Fraser, 2010). Finally, the corresponding sentence-level speech segments are obtained via audio-text alignment (McAuliffe et al., 2017).

Since speech is continuous, rather than defining several independent speech sequences $\mathbf{s}^{(i,j)}$, it is standard to define start and end timestamps $\text{start}^{(i,j)}, \text{end}^{(i,j)} \in \mathbb{R}$ relative to the document-level speech $\mathbf{S}^{(i)}$. Thus, $\mathcal{D}^{\text{sent}}$ can be re-defined as:

$$\mathcal{D}^{\text{sent}} = \left\{ \mathbf{X}^{(i)}, \left\{ (b^{(i,j)}, \mathbf{x}^{(i,j)}, \mathbf{y}^{(i,j)}) \right\}_{j=1}^{m_i} \right\}_{i=1}^n \quad (4.3)$$

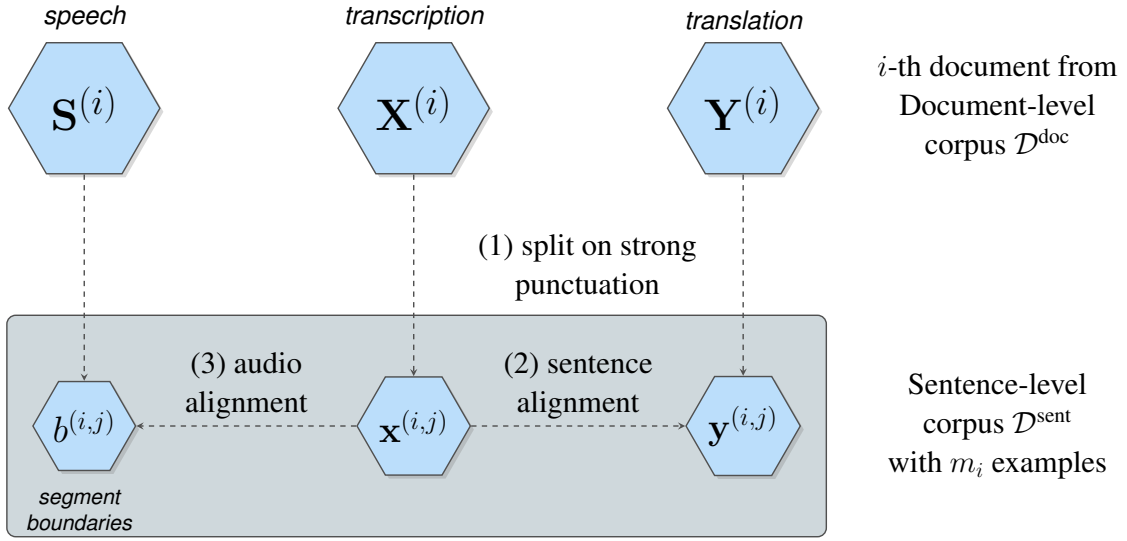


Figure 4.2: Manual Segmentation of a document-level example i from a document-level corpus $\mathcal{D}^{\text{doc}}$ into m_i examples of a sentence-level corpus $\mathcal{D}^{\text{sent}}$. Process: (1) the document-level transcription $\mathbf{X}$ and translation $\mathbf{Y}$ are split on strong punctuation into a set of sentence-level transcriptions and translations and then in (2) they are aligned with a sentence alignment tool to produce $\{\mathbf{x}, \mathbf{y}\}_j^{m_i}$. Finally, in (3) are obtained the segment boundaries that correspond to each sentence-level transcription via audio alignment.

where $b^{(i,j)} = (\text{start}^{(i,j)}, \text{end}^{(i,j)})$ is the tuple of segment boundaries for the j -th segment, such that it can slice the document-level speech to obtain the sentence-level segment $\mathbf{s}^{(i,j)} = \mathbf{S}^{(i)}_{\text{start}^{(i,j)}:\text{end}^{(i,j)}}$. Note that gaps often exist between consecutive segments due to silent periods or unaligned content.

4.4 The SEGAUGMENT Framework

The proposed data augmentation method, SEGAUGMENT, aims to maximize the utility of the manually segmented training data $\mathcal{D}^{\text{sent}}$ by generating an alternative synthetic sentence-level dataset $\hat{\mathcal{D}}^{\text{sent}}$. The synthetic dataset is derived from different segmentations of the original speech documents in $\mathcal{D}^{\text{doc}}$.

While the manual segmentation relies on grammatical features and punctuation (§4.3), we propose to split based on acoustic features by utilizing the SHAS method introduced in Ch. 3.

The overall pipeline, illustrated in Figure 4.3, consists of three main steps. For a given

speech document $\mathbf{S}^{(i)}$, we first generate alternative segmentation boundaries $\hat{b}$ using SHAS with specific length parameterizations (§4.4.1). Next, we recover the corresponding source transcriptions $\hat{\mathbf{x}}$ via CTC-based forced alignment (§4.4.2). Finally, we obtain the target translations $\hat{\mathbf{y}}$ by aligning the text with a Machine Translation model (§4.4.3).

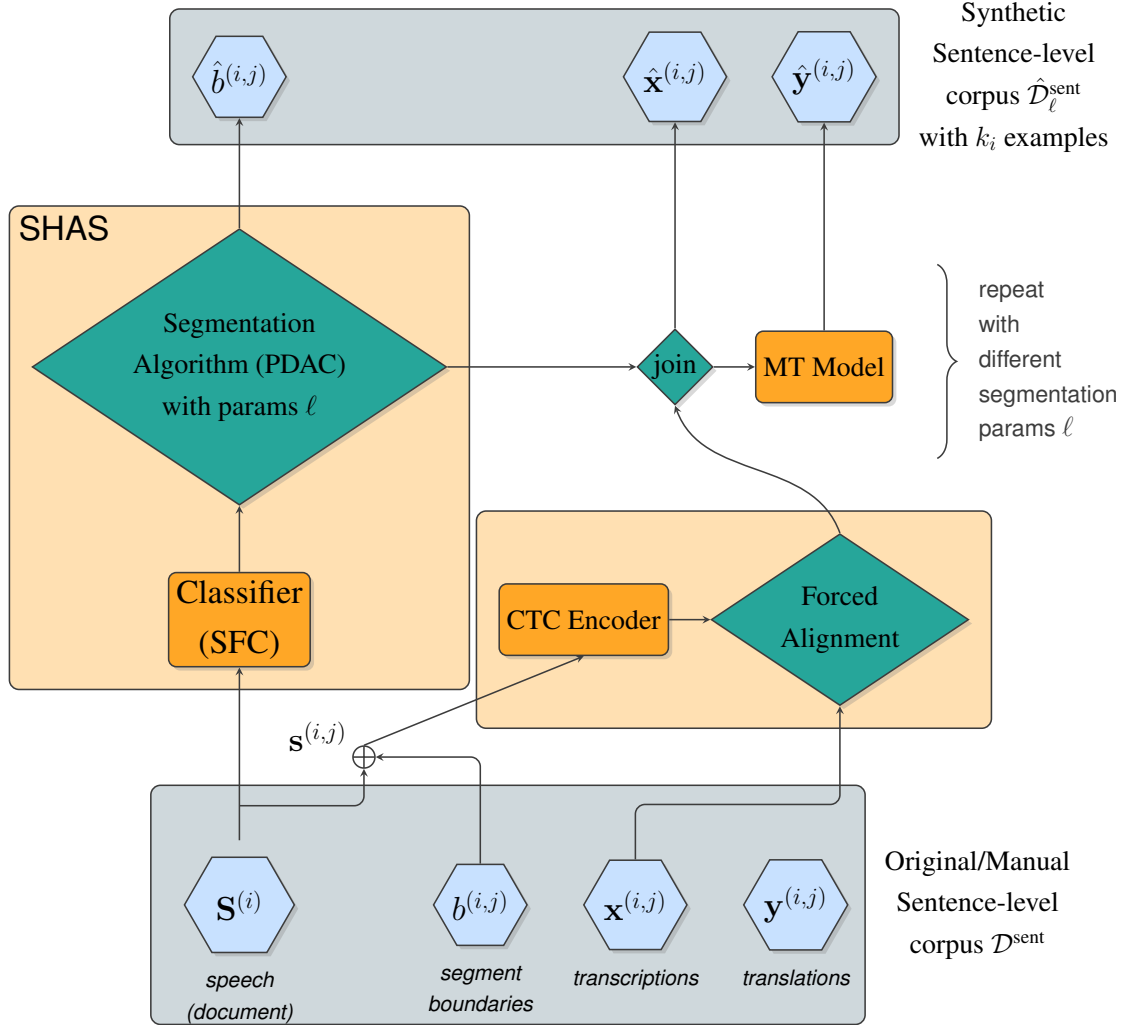


Figure 4.3: The SEGAUGMENT pipeline. Given a document-level example from $\mathcal{D}^{\text{doc}}$, we create k_i synthetic sentence-level examples by re-segmenting the audio with SHAS. We then recover the transcription via CTC forced alignment and generate the translation using an MT model. Changing the segmentation parameters ℓ results in different synthetic corpora $\hat{\mathcal{D}}_{\ell}^{\text{sent}}$.

4.4.1 Diversifying Inputs via Re-segmentation

The first step is to obtain new, acoustically valid segments from the continuous audio. We follow the inference process described for SHAS in §3.4. We obtain a set of alternative

segmentation boundaries $\{b\}_j^{k_i}$ for each document $S^{(i)}$ in the training corpus by applying the SHAS Segmentation Frame Classifier (SFC) (§3.4.1) and the Probabilistic Divide-and-Conquer (PDAC) segmentation algorithm (§3.4.2).

There is a key difference in how we apply the method here compared to the inference-time scenario in Ch. 3. Previously, our goal was to find a single, optimal segmentation that maximized translation quality. Here, our goal is diversity. We purposely induce variation by running the segmentation algorithm with arbitrary values for the length constraints (ℓ_{min}, ℓ_{max}) .

By defining different, non-overlapping tuples of ℓ , we force the algorithm to view the document through different temporal windows. For example, setting $\ell = (0.4, 3)$ forces the creation of short, sub-sentence segments, while $\ell = (10, 20)$ produces long, multi-sentence segments. This generates multiple synthetic versions of the dataset where the same speech content is sliced in distinct ways, providing the downstream ST model with diverse contextual views of the same data.

Specifically, for a document i , we first perform inference on the speech $S^{(i)}$ with the classifier of SHAS to obtain the sequence of frame-level probabilities $\mathbf{p}^{(i)}$ that indicate the probability that each frame in the audio contains important information that should be part of a segment. Then, using the segmentation algorithm and a parameterization $\ell = (\ell_{min}, \ell_{max})$, we obtain a set of new (plausible) segmentation boundaries.

$$\mathbf{p}^{(i)} = \text{CLASSIFIER}(S^{(i)}) \quad (4.4)$$

$$b^{(i,1)}, \dots, b^{(i,k_i)} = \text{SEGMENTATION-ALGORITHM}(\mathbf{p}^{(i)}; \ell) \quad (4.5)$$

where k_i is the number of new segment boundaries.

4.4.2 Transcript Recovery via Forced Alignment

Once we have defined new segmentation boundaries, the original sentence-level transcriptions $\{\mathbf{x}^{(i,j)}\}_j^{m_i}$ are no longer valid as they correspond to the manual segmentation boundaries. We must “re-transcribe” the new segments. Since we possess the ground truth transcription for the full document, this is an alignment problem rather than a recognition problem.

We employ CTC-based forced alignment (Kürzinger et al., 2020) to map the text to the audio. First, we pass the speech of the manually segmented examples through a CTC encoder (e.g., a pre-trained WAV2VEC 2.0 (Baevski et al., 2020) model) to obtain character-level probabilities (see §2.4.2 for more details on CTC). Simultaneously, we normalize the original text to match the model’s character set (e.g., removing punctuation, lower-casing) while maintaining a mapping to the original surface forms.

Using the CTC probabilities and the cleaned text, the forced alignment algorithm computes the optimal alignment path, assigning start and end timestamps to each character. We then merge these characters into words. Finally, for each new synthetic segment $b^{(i,j)}$, we retrieve the corresponding source text $\hat{\mathbf{x}}^{(i,j)}$ by retrieving all words whose timestamps fall within the start and end segment boundary. The original casing and punctuation are preserved via the stored mapping.

The pseudocode for this alignment and retrieval process is detailed in §A.1.

4.4.3 Translation Recovery via Text Alignment

The final component is the translation. Unlike the transcription, which implies a monotonic relationship with the audio, the translation involves reordering and complex dependencies. Thus, simple timestamp-based slicing is infeasible.

Standard cross-lingual sentence alignment tools (Braune and Fraser, 2010) are not applicable as our synthetic segments may not correspond to full sentences. Furthermore, we found that traditional word alignment tools (Dyer et al., 2013) yielded sub-optimal results for these partial segments.

To address this, we treat alignment as a translation task. We learn the alignment using a Machine Translation (MT) model (e.g., a small transformer encoder-decoder with 10M parameters (Vaswani et al., 2017)). This model is trained on the manually segmented sentence-level parallel data $\{\mathbf{x}^{(i,j)}, \mathbf{y}^{(i,j)}\}_{i=1, j=1}^{n, m_i}$.

Crucially, we minimize regularization to encourage the model to memorize the alignment patterns of the training set rather than generalize to unseen text. Since no sentence-level references exist for the synthetic segments, we monitor the document-level BLEU score on a subset of training documents and stop training when this metric plateaus.

Once trained, we use this model to translate the synthetic transcriptions $\{\hat{\mathbf{x}}^{(i,j)}\}_j^{k_i}$ generated in the previous step, yielding the synthetic translations $\{\hat{\mathbf{y}}^{(i,j)}\}_j^{k_i}$. Note, that this process implicitly introduces Knowledge Distillation (KD) (Hinton et al., 2015) into the pipeline (since the targets for the synthetic data are generated by a model), but as we later show in our analysis, the effect of KD is minimal (§4.8).

Creating Multiple Sentence-level Versions. By repeating this pipeline with different parameterizations of the segmentation algorithm, we generate multiple synthetic versions of the original corpus. For a tuple of length constraints $\ell = (\ell_{min}, \ell_{max})$, we define the synthetic speech corpus $\hat{\mathcal{D}}_\ell^{\text{sent}}$ with $k = \sum_i^n k_i$ examples as:

$$\hat{\mathcal{D}}_\ell^{\text{sent}} = \left\{ \mathbf{S}^{(i)}, \left\{ \hat{b}^{(i,j)}, \hat{\mathbf{x}}^{(i,j)}, \hat{\mathbf{y}}^{(i,j)} \right\}_{j=1}^{k_i} \right\}_{i=1}^n \quad (4.6)$$

where $\mathbf{S}^{(i)}$ is the original i -th document audio, and the triplets inside the set represent the new segments, recovered transcripts, and generated translations.

The computational cost of generating multiple versions is relatively low. The computationally expensive steps, which are the SHAS classification and CTC encoding, depend only on the audio documents and not the segmentation boundaries. Therefore, their outputs can be cached and reused across different length configurations; only the lightweight segmentation algorithm and the MT generation need to be re-run for each new parameterization ℓ .

4.5 Experimental Setup

4.5.1 Data

We conduct extensive experiments across three standard ST benchmarks: MuST-C (Di Gangi et al., 2019a), mTEDx (Salesky et al., 2021), and CoVoST 2 (Wang et al., 2020c). The statistics for all datasets are summarized in Table 4.1.

MuST-C. We utilize versions v1.0 and v2.0 of this corpus, which is derived from TED talks. Since these data originate from full documents, they are ideal for validating our

Dataset	Lang. Pair	# Docs	# Sents	Hours
MuST-C v1.0	eng-deu	2,093	234K	408
	eng-spa	2,514	266K	496
	eng-fra	2,460	275K	485
	eng-ita	2,324	254K	457
	eng-nld	2,219	248K	434
	eng-por	2,001	206K	377
	eng-ron	2,166	236K	424
	eng-rus	2,448	265K	482
MuST-C v2.0	eng-deu	2,537	251K	450
mTEDx	spa-eng	378	36K	64
	por-eng	279	31K	53
	spa-fra	43	4K	6
CoVoST 2	eng-deu	-	231K	362

Table 4.1: Training Data Statistics. Note that for CoVoST 2, the data is not document-level, so the document count is not applicable.

re-segmentation approach. For our main experiments, we use the eight language pairs from MuST-C v1.0, translating from English (eng) to German (deu), Spanish (spa), French (fra), Italian (ita), Dutch (nld), Portuguese (por), Romanian (ron), and Russian (rus). Additionally, we use the eng-deu pair from v2.0 for ablation studies and analysis.

mTEDx. To assess performance in low-resource and non-English source scenarios, we experiment with the mTEDx corpus. We focus on three language directions: Spanish-to-English (spa-eng), Portuguese-to-English (por-eng), and the extremely low-resource pair Spanish-to-French (spa-fra). Similar to MuST-C, mTEDx is natively document-level.

CoVoST 2. Finally, to test the applicability of our method on data that is not document-based, we use the eng-deu direction of CoVoST 2. Unlike the TED-based corpora, CoVoST is based on Common Voice (Ardila et al., 2020) and consists of isolated sentences. We treat these sentences as short “documents” to apply our pipeline.

4.5.2 Method Configuration

For the segmentation step, we utilize the pre-trained SHAS classifiers introduced in Ch. 3. To generate diverse views of the data, we define four non-overlapping tuples of length constraints (ℓ_{min}, ℓ_{max}) :

- **Short (s):** (0.4, 3) seconds
- **Medium (m):** (3, 10) seconds
- **Long (l):** (10, 20) seconds
- **Extra-Long (xl):** (20, 30) seconds

We use the Probabilistic Divide-and-Conquer (PDAC) algorithm (§3.4.2) for the s , m , and l configurations. For the xl configuration, we employ the Streaming (PSTRM) algorithm (Gaido et al., 2021b), as we observed that PDAC struggled to satisfy the long duration constraints in the optimization.

This process yields four synthetic corpora: $\hat{\mathcal{D}}_s^{\text{sent}}$, $\hat{\mathcal{D}}_m^{\text{sent}}$, $\hat{\mathcal{D}}_l^{\text{sent}}$, and $\hat{\mathcal{D}}_{xl}^{\text{sent}}$. During training, we simply concatenate these synthetic datasets to the original corpus $\mathcal{D}^{\text{sent}}$ and remove any exact duplicates.

For the alignment steps, we use pre-trained WAV2VEC 2.0 models (Baevski et al., 2020) as the CTC encoder, and for the text alignment, we train small Transformer models with 6 layers.

4.5.3 Speech Translation

Models. Our primary experimental testbed is the Speech-to-Text (S2T) Transformer architecture (Wang et al., 2020b) (see §2.6.4 for details). We use the small configuration, which consists of 12 encoder layers, 6 decoder layers, and a model dimension of 256 (31M parameters). For the extremely low-resource pair mTEDx spa-fra, we use the extra-small configuration (10M parameters) to prevent overfitting. Following standard practice, we pre-train the encoder of these models on the ASR task using the available transcriptions. Unless stated otherwise, this ASR pre-training uses only the original data.² We then fine-tune the entire model on the ST task.

²In certain experiments we also apply SEGAUGMENT to the ASR pre-training.

For our state-of-the-art comparison experiments, we also employ a stronger baseline that combines a WAV2VEC 2.0 encoder with an MBART-50 decoder (see §2.6.4). The encoder is composed of a 7-layer convolutional feature extractor and 24-layer Transformer encoder, while the decoder has 12 layers (770M parameters).

Full details regarding hyperparameters, training schedules, and model architectures are provided in §A.2.

Evaluation. For inference, we average the 10 best checkpoints on the validation set and decode with a beam size of 5. We evaluate with BLEU (Papineni et al., 2002) using sacreBLEU (Post, 2018), and perform statistical significance testing using paired bootstrap resampling (Koehn, 2004) to ensure valid comparisons. To evaluate translation quality on an automatic segmentation, the hypotheses are aligned to the references from the manual segmentation using mwerSegmenter (Matusov et al., 2005).

4.6 Maximizing Data Utility and Generalization

In this section, we conduct a comprehensive evaluation of our proposed data augmentation method to quantify its impact on translation quality and robustness. We structure our analysis to answer five key questions:

- Does re-segmentation improve performance on standard benchmarks (MuST-C) and standard baselines (S2T-TRANSFORMER)? (see §4.6.1)
- Can it boost state-of-the-art models that already utilize large-scale pre-training? (see §4.6.2)
- Is the method effective in low-resource settings (mTEDx) where data scarcity is most acute? (see §4.6.3)
- Can it generalize to sentence-level corpora (CoVoST 2) lacking document context? (see §4.6.4)
- Can the diversity provided by SEGAUGMENT compensate for the removal of the computationally expensive ASR pre-training stage? (see §4.6.5)

We present these findings sequentially below.

Model	deu	spa	fra	ita	nld	por	ron	rus	Average
<i>Previous Works</i>									
S2T-TRANSFORMER (Wang et al., 2020b)	22.7	27.2	32.9	22.7	27.3	28.1	21.9	15.3	24.8
<i>This Work</i>									
S2T-TRANSFORMER	23.2	27.5	33.1	22.9	27.3	28.7	22.2	15.3	25.0
+ SEGAUGMENT	25.0	29.3	35.7	25.6	30.3	31.8	25.0	16.8	27.4
+ ASR SEGAUGMENT	25.5	29.5	35.8	25.7	30.0	31.6	25.0	16.7	27.5

Table 4.2: BLEU($\uparrow$) scores on MuST-C v1.0 tst-COMMON eng-xxx. In **bold** is the best score overall. All results with SEGAUGMENT and ASR SEGAUGMENT are statistically different from the S2T-TRANSFORMER baseline with $p < 0.001$. All models use the same architecture.

4.6.1 Augmenting Standard Baselines

We first evaluate the effectiveness of our proposed method on the eight language pairs of MuST-C v1.0. We compare ST models trained with and without SEGAUGMENT against a standard Speech-to-Text Transformer baseline (§2.6.4). For reference, we also include results from Wang et al. (2020b), which utilize the same model architecture.

As shown in Table 4.2, models leveraging SEGAUGMENT achieve significant and consistent improvements across all language pairs. This confirms that generating synthetic views of the training data allows the model to better utilize the available acoustic and linguistic information. Specifically, the improvements range from 1.5 to 3.1 BLEU points, with an average gain of 2.4 points over the baseline.

We also investigate the application of SEGAUGMENT during the ASR pre-training stage (“+ ASR SEGAUGMENT”). While this strategy brings further marginal gains in four language pairs (e.g., eng-deu, eng-spa), the average improvement over using SEGAUGMENT only for the ST fine-tuning is negligible (0.1 BLEU). This suggests that the primary benefit of the augmentation comes from the diversity it provides to the translation task itself, although as we see later, in lower-resource scenarios, the benefits of the pre-training are more evident (§4.6.3).

For a fine-grained analysis on how different combinations of synthetic data from SEGAUGMENT (e.g., “short”, “long”) affect translation quality of the trained model, see Table A.2 in the Appendix. For the results of the ASR-pretraining, see Table A.1.

Model	eng-spa	eng-fra
<i>Previous Works</i>		
CHIMERA (Han et al., 2021)	30.6	35.6
STEMM (Fang et al., 2022)	31.0	37.4
CONST (Ye et al., 2022)	32.0	38.3
STPT (Tang et al., 2022)	33.1	39.7
SPEECHUT (Zhang et al., 2022b)	33.6	41.4
<i>This Work</i>		
w2v-MBART	33.3	40.8
+ SEGAUGMENT	33.7	41.5

Table 4.3: BLEU($\uparrow$) scores on MuST-C v1.0 tst-COMMON with SOTA models. In **bold** is the best score overall. Results with SEGAUGMENT are statistically different from the w2v-mBART baseline model with $p < 0.001$ for eng-deu and $p < 0.005$ for eng-spa.

4.6.2 Pushing State-of-the-Art Performance

In this section, we study the impact of the proposed method when combined with a much stronger ST model. We utilize w2v-MBART, a model comprised of 24 encoder layers and 12 decoder layers with a model dimension of 1024. The encoder is initialized from WAV2VEC 2.0 (Baevski et al., 2020) and the decoder from MBART-50 (Liu et al., 2020a) (see §A.2 for more details).

We fine-tune this model end-to-end on MuST-C v1.0, specifically for the English-to-Spanish (eng-spa) and English-to-French (eng-fra) directions. We compare the performance with and without the synthetic data generated by SEGAUGMENT. Additionally, we compare our results against other state-of-the-art (SOTA) methods, including CHIMERA (Han et al., 2021), STEMM (Fang et al., 2022), CONST (Ye et al., 2022), STPT (Tang et al., 2022), and SPEECHUT (Zhang et al., 2022b).

As presented in Table 4.3, even with a highly competitive baseline (w2v-MBART), utilizing SEGAUGMENT achieves further significant improvements.³ Notably, this combination allows us to achieve new state-of-the-art performance on both eng-spa and eng-fra.

³Compared with §4.6.1, smaller gains are expected here since the effect of data augmentation typically diminishes as the model capacity and pre-training data volume increase.

Model	spa-eng	por-eng	spa-fra
<i>Previous Works</i>			
S2T-TRANSFORMER (Salesky et al., 2021)	7.0 ^{xs}	8.1 ^{xs}	1.7 ^{xs}
<i>This Work</i>			
S2T-TRANSFORMER	15.6 ^s	15.3 ^s	2.7 ^{xs}
+ SEGAUGMENT	18.5 ^s	17.8 ^s	3.8 ^{xs}
+ ASR SEGAUGMENT	19.1^s	20.3^s	5.3^{xs}

Table 4.4: BLEU(↑) scores on the mTEDx test set. In **bold** is the best score overall. All results with SEGAUGMENT and ASR SEGAUGMENT are statistically different from the S2T-TRANSFORMER baseline with $p < 0.001$. The superscript indicates the architecture configuration used, small (s, 31M) or extra-small (xs, 10M).

4.6.3 Impact in Low-Resource Scenarios

We next explore the application of SEGAUGMENT in the low-resource and non-English source settings of mTEDx. In Table 4.4, we present the results for Spanish-to-English (spa-eng), Portuguese-to-English (por-eng), and the extremely low-resource pair Spanish-to-French (spa-fra), which contains only 6 hours of training data. For comparison, we include the BLEU scores from Salesky et al. (2021) that uses the same architecture.

SEGAUGMENT yields significant improvements across all pairs. Notably, unlike the high-resource scenario in MuST-C (§4.6.1), utilizing SEGAUGMENT during the ASR pre-training brings substantial additional gains here, improving the final ST BLEU scores by 2.6 to 5.0 points over the baseline. This suggests that in data-scarce regimes, the robust acoustic representations learned via augmentation during pre-training are crucial for downstream translation performance.

4.6.4 Applicability to Sentence-Level Data

Finally, we consider the application of the method to CoVoST 2, a corpus where the data does not originate from continuous documents but rather from isolated sentences. To apply SEGAUGMENT, we treat these sentence-level examples as if they were short “documents” and run the pipeline as described in §4.4. However, due to the limited duration of the original clips, we only generate synthetic versions using the *short* and *medium* length configurations (§4.5.2).

Table 4.5 presents the performance for the English-to-German (eng-deu) direction. We

Model	eng-deu
<i>Previous Works</i>	
S2T-TRANSFORMER (Wang et al., 2020c)	16.3
S2T-TRANSFORMER w/ STR (Lam et al., 2022)	18.8
<i>This Work</i>	
S2T-TRANSFORMER	17.4
+ SEGAUGMENT	19.0
+ ASR SEGAUGMENT	19.3

Table 4.5: BLEU(↑) scores on the CoVoST 2 test set. In **bold** is the best score overall. All results with SEGAUGMENT and ASR SEGAUGMENT are statistically different from our S2T-TRANSFORMER baseline with $p < 0.001$.

compare our method against two previous works: (a) a bilingual baseline from Wang et al. (2020c), and (b) the Sample-Translate-Recombine (STR) augmentation method (Lam et al., 2022). All methods including ours, use the S2T-TRANSFORMER model.

Although SEGAUGMENT was explicitly designed for document-level data, it yields significant improvements over our baseline.⁴ Furthermore, it outperforms the STR augmentation method by 0.5 BLEU points, demonstrating its versatility even when long-form context is unavailable.

4.6.5 Independence from ASR Pre-training

Standard ST training pipelines typically rely on an ASR pre-training phase to initialize the encoder (Wang et al., 2020b). While effective, this stage is computationally expensive and requires access to transcribed speech data. In this section, we investigate whether the data diversity provided by SEGAUGMENT can compensate for the lack of this initialization.

Table 4.6 presents the results on MuST-C eng-spa and eng-fra. We compare models trained with and without ASR pre-training, both for the baseline and our method. We also include results from the ST-FROM-SCRATCH method (Zhang et al., 2022a), which was specifically designed to train high-performance ST models without ASR initialization.

Remarkably, we find that models trained with SEGAUGMENT and without any ASR pre-training are highly competitive. In fact, they surpass the S2T-TRANSFORMER baseline

⁴We hypothesize a potential denoising effect: CoVoST clips often contain unvoiced audio at the start or end, which our segmentation method likely excludes from the synthetic segments.

Model	# Params	ASR Pre-train	eng-spa	eng-fra
<i>Previous Works</i>				
ST-FROM-SCRATCH (Zhang et al., 2022a)	48M	✗	28.1	33.4
<i>This Work</i>				
S2T-TRANSFORMER	31M	✗	25.3	30.1
+ SEGAUGMENT	31M	✗	28.5	34.6
S2T-TRANSFORMER	31M	✓	27.5	33.1
+ SEGAUGMENT	31M	✓	29.3	35.7

Table 4.6: BLEU(↑) scores on MuST-C *tst*-COMMON investigating the impact of ASR pre-training. #Params indicates the model size. In **bold** is the best score among the models trained *without* ASR pre-training (✗).

that *does* use ASR pre-training (e.g., 28.5 vs 27.5 BLEU on eng-spa) and outperform the specialized (and also larger in terms of parameters) ST-FROM-SCRATCH system.

These results suggest that in scenarios where computational resources are limited, ASR pre-training could potentially be skipped in favor of using SEGAUGMENT. However, as shown in the bottom row of the table, the best overall performance is achieved by combining both techniques, indicating that they offer complementary benefits to the model.

4.7 Solving the Training-Inference Segmentation Mismatch

Unlike the controlled environments of most research benchmarks, manual segmentation is rarely available in real-world deployment scenarios. Consequently, ST models must rely on automatic segmentation methods to process continuous audio. However, as discussed in §3.1.1, evaluating on automatic segmentation typically incurs a performance penalty of 5–10% BLEU compared to the manual (gold) segmentation (§3.6).

Since the synthetic data generated by SEGAUGMENT are derived via an automatic segmentation process, we hypothesize that they can bridge this training-inference mismatch. By exposing the model to diverse, machine-generated boundaries during training, it should become more robust to different conditions at inference time.

To test this, we evaluate our models on MuST-C *tst*-COMMON using two distinct segmentation schemes:

1. **Manual:** The gold-standard segmentation provided with the dataset.
2. **Automatic:** A machine-generated segmentation obtained using the SHAS method (Ch. 3). We select the *long* configuration as it generally yields the best trade-off between latency and quality (see Table A.4 for more information).

Special Tokens Strategy. To further explicitly model the segmentation characteristics, we introduce a variant of our training setup using special tokens. We prepend a tag to the source transcript of each training example indicating its origin: `<original>` for manual data, and `<s>`, `<m>`, `<l>`, or `<x1>` for the synthetic SEGAUGMENT versions. During inference, we prompt the model with the token that best corresponds to the segmentation method being used (e.g., prompting with `<l>` when evaluating on SHAS-*long*).

Experimental Findings. The results are presented in Table 4.7. First, we confirm the findings of previous research: the S2T-TRANSFORMER baseline model suffers a significant degradation when moving from Manual to Automatic segmentation, dropping 1.6 BLEU points (6%) on average.

However, applying SEGAUGMENT validates our hypothesis. The method improves performance on the Automatic segmentation by 3.5 BLEU ($23.4 \rightarrow 26.9$), a gain significantly larger than the 2.4 BLEU improvement seen on the Manual segmentation.

Most notably, when combining SEGAUGMENT with the Special Tokens strategy (“+ *special tok.*”), the ST models achieve an average score of 27.3 BLEU on the automatic segmentation. This effectively closes the gap with the manual segmentation performance (27.4 BLEU). In three language pairs (eng-deu, eng-spa, eng-rus), the model evaluated on automatic segments actually outperforms the baseline evaluated on gold segments.

Remarkably, this result challenges the long-standing assumption that manual segmentation represents the upper bound of performance for sentence-level ST models. It demonstrates that by aligning the training data distribution with the inference constraints via SEGAUGMENT, we can achieve high-quality translation even in realistic real-world scenarios, where a segmentation is not available, and has to be inferred.

Lang. Pair	Model	Test Set Segmentation	
		Manual	Automatic
eng-deu	S2T-TRANSFORMER	23.2	21.2
	+ SEGAUGMENT	25.0	24.5
	+ <i>special tok.</i>	25.1	25.2
eng-spa	S2T-TRANSFORMER	27.5	26.1
	+ SEGAUGMENT	29.3	29.0
	+ <i>special tok.</i>	29.2	29.4
eng-fra	S2T-TRANSFORMER	33.1	30.6
	+ SEGAUGMENT	35.7	34.9
	+ <i>special tok.</i>	35.6	35.3
eng-ita	S2T-TRANSFORMER	22.9	21.5
	+ SEGAUGMENT	25.6	24.4
	+ <i>special tok.</i>	25.0	25.1
eng-nld	S2T-TRANSFORMER	27.3	25.7
	+ SEGAUGMENT	30.3	29.7
	+ <i>special tok.</i>	29.4	29.8
eng-por	S2T-TRANSFORMER	28.7	26.9
	+ SEGAUGMENT	31.8	31.4
	+ <i>special tok.</i>	31.5	31.8
eng-ron	S2T-TRANSFORMER	22.2	20.5
	+ SEGAUGMENT	25.0	24.1
	+ <i>special tok.</i>	24.8	24.6
eng-rus	S2T-TRANSFORMER	15.3	14.6
	+ SEGAUGMENT	16.8	16.8
	+ <i>special tok.</i>	16.4	17.0
Average	S2T-TRANSFORMER	25.0	23.4
	+ SEGAUGMENT	27.4	26.9
	+ <i>special tok.</i>	27.1	27.3

Table 4.7: BLEU($\uparrow$) scores of different methods on MuST-C v1.0 tst-COMMON evaluated on Manual (gold) and Automatic (SHAS-*long*) segmentations. The best score for each language pair and segmentation type is highlighted. The best score overall for each language pair is in **bold**. Extended results including more automatic segmentations are available at Table A.4 of the Appendix.

4.8 Why it Works: Mechanisms of Improvement

Having established the empirical effectiveness of SEGAUGMENT, in this section we analyze the potential mechanisms driving these improvements. We identify four key factors: contextual diversity, positional diversity, length specialization, and the knowledge distillation effect.

4.8.1 Contextual Diversity

The primary hypothesis behind SEGAUGMENT is that re-segmenting the audio creates new training examples where speech and text units appear within novel and diverse contexts. We can categorize the relationship between a new synthetic segments and the original manual segments into four types:

1. **Isolated:** The new segment is a subset of an original one.
2. **Expanded:** The new segment is a superset of an original one.
3. **Mixed:** The new segment overlaps with boundaries of original segments.
4. **Equal:** The new segment is identical to an original one.

In Table 4.8, we quantify these types for the MuST-C v2.0 eng-deu dataset. We observe that only a small fraction (8.0%) of the synthetic examples are identical to the manual ones (which we discard during training). The vast majority (92%) present the data in a new light. For instance, the *short* configuration (*s*) generates mostly “Isolated” examples (56.9%), effectively creating sub-sentence training pairs. Conversely, the *long* configuration (*l*) produces “Expanded” examples (63.2%), providing the model with supra-sentential⁵ context. We hypothesize that this variance reduces overfitting to specific context windows, forcing the model to rely on local acoustic cues.

Configuration	Expanded	Isolated	Mixed	Equal
<i>s</i> (0.4-3s)	12.3%	56.9%	22.3%	8.5%
<i>m</i> (3-10s)	31.1%	33.3%	24.3%	11.3%
<i>l</i> (10-20s)	63.2%	7.9%	24.6%	4.2%
<i>xl</i> (20-30s)	55.7%	1.2%	43.0%	0.1%
Average	28.3%	38.7%	25.0%	8.0%

Table 4.8: Contextual categorization of synthetic examples compared to the original manual segmentation (MuST-C v2.0 eng-deu). Approximately 92% of synthetic examples offer a new contextual view.

⁵Refers to linguistic elements, features, or phenomena that exist above the level of the individual sentence.

4.8.2 Positional Diversity

Standard Transformer models use absolute positional embeddings (Vaswani et al., 2017), which can lead to overfitting, particularly for rare words that may always appear at specific positions in the training sentences (Sinha et al., 2022).

SEGAUGMENT naturally mitigates this by shifting the absolute position of content. Because we slice the document at arbitrary points, a word that was mostly seen at the start of a segment in the original examples might appear now at several new absolute positions in a “Mixed” or “Expanded” synthetic segment.

In Figure 4.4, we visualize the positional frequency of four semi-rare sub-words in the target German text. In the original data (blue), these tokens often spike at specific positions (e.g., “Transistor” appearing mostly at the end of sentences). With SEGAUGMENT (orange), the distribution is smoothed and spread across a much wider range of positions. We hypothesize that this acts as a form of regularization, preventing the model from memorizing spurious position-content correlations.

4.8.3 Length Specialization

The manual segmentation of datasets like MuST-C typically follows a Bell-shaped distribution centered around 5–10 seconds. Consequently, standard models often struggle with “outlier” lengths, such as very short utterances or very long sequences.

SEGAUGMENT explicitly populates these tails of the distribution. To verify this “length specialization” effect, we evaluated model performance on two extreme subsets of the MuST-C `tst-COMMON` set: (a) *very short* examples (shorter than 1 second), and (b) *very long* examples (longer than 20 seconds).

As shown in Figure 4.5, there is a clear correlation between the training data length and performance on these buckets. Models trained with shorter synthetic data (e.g., *short*, *medium*) excel at translating the *very short* bucket, while models trained with longer synthetic data (*xlong*) dominate on the *very long* bucket. By combining all these versions, the final model gains the capability to handle the full spectrum of input lengths effectively.

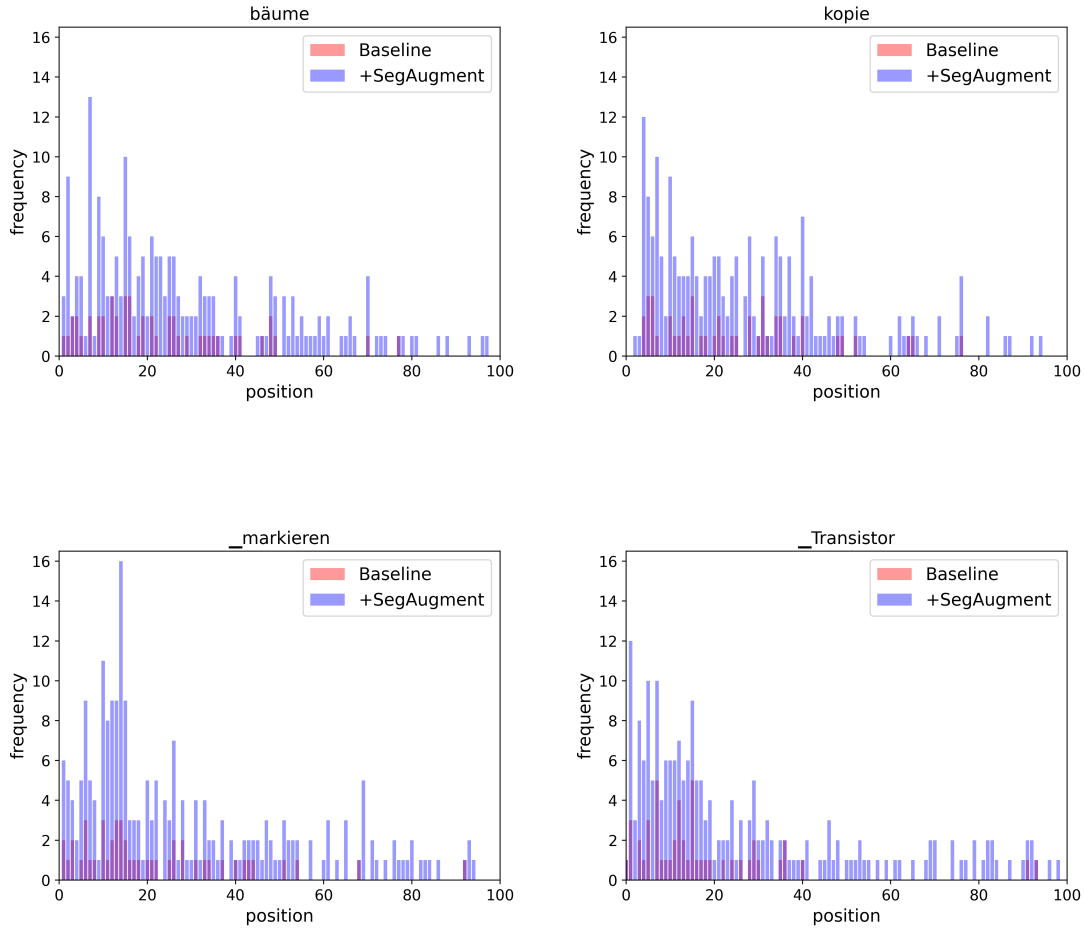


Figure 4.4: Positional frequency of semi-rare sub-words in the MuST-C v2.0 eng-deu training set. The original data (blue) shows sharp peaks, while SEGAUGMENT (orange) spreads the occurrences across diverse absolute positions.

4.8.4 Knowledge Distillation vs. Data Augmentation

Since the translations for our synthetic segments are generated by an MT model, our pipeline inherently involves a form of Knowledge Distillation (KD) (Liu et al., 2019; Gaido et al., 2020a). It is therefore important to disentangle how much of the gain comes from simply having synthetic targets (KD effect) versus the diversity of inputs (Augmentation effect).

To quantify this, we performed a controlled experiment on MuST-C v2.0 eng-deu. We created a standard “in-data KD” baseline by re-translating the *original* manual segments

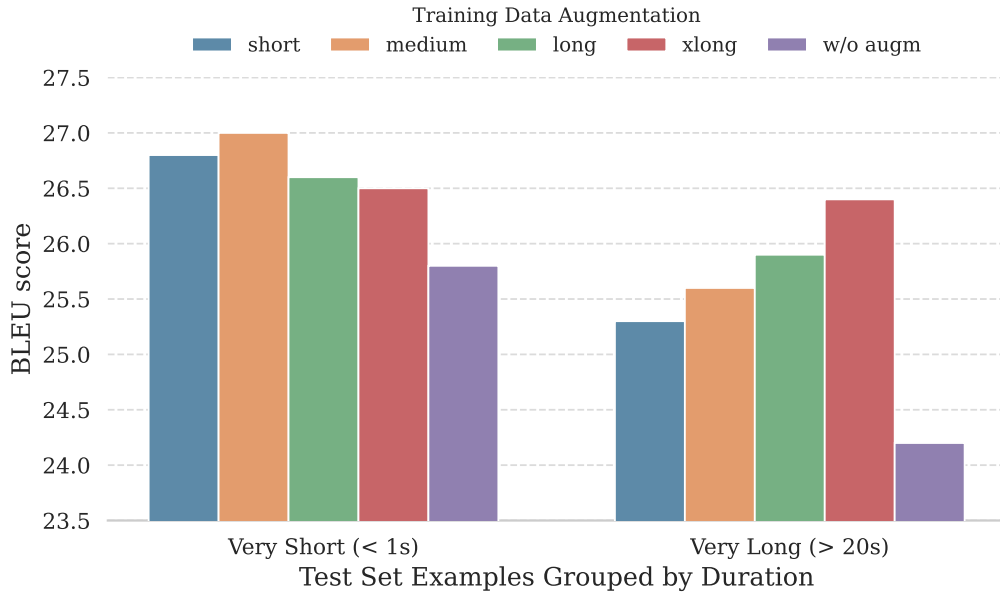


Figure 4.5: BLEU scores on extreme duration buckets of MuST-C tst-COMMON (averaged across 8 language pairs). The results show the “length specialization” effect: models trained with short synthetic data perform best on very short test examples (< 1s), while those trained with extra-long synthetic data perform best on very long test examples (> 20s). The baseline (*w/o augm*) struggles in both extremes.

Model	eng-deu
S2T-TRANSFORMER	24.3
S2T-TRANSFORMER + in-data KD	25.2
S2T-TRANSFORMER + SEGAUGMENT	26.2

Table 4.9: BLEU(↑) scores on MuST-C v2.0 tst-COMMON eng-deu. SEGAUGMENT statistically outperforms standard in-data Knowledge Distillation ($p < 0.001$), demonstrating that input diversification adds value beyond simple target cleaning.

using the same MT model used in SEGAUGMENT. This provides the target-side benefits of KD without the source-side diversity of re-segmentation.

Table 4.9 presents the comparison. While standard in-data KD improves over the baseline (24.3 → 25.2 BLEU), SEGAUGMENT provides a significantly larger gain (26.2 BLEU). This 1.0 BLEU difference confirms that while KD contributes to the performance, it is not the sole driver. SEGAUGMENT is more general: it provides diversity on both the source side (via re-segmentation) and the target side (via MT generation), whereas KD is limited to the target side.

4.9 Summary and Conclusions

In this chapter, we addressed the limitations imposed by the static nature of standard speech translation corpora. We introduced SEGAUGMENT, a data augmentation strategy that transforms the segmentation process from a fixed preprocessing step into a source of dynamic supervision. By leveraging the SHAS method (described in Ch. 3) with variable length constraints, we successfully generated diverse, acoustically valid views of the training data.

Through extensive experimentation, we demonstrated that this approach yields consistent and significant improvements across a wide range of conditions. On standard benchmarks like MuST-C, SEGAUGMENT improved translation quality by 1.5 to 3.1 BLEU points. In low-resource scenarios such as mTEDx, the gains were even more pronounced, reaching up to 5 BLEU points when the augmentation was also applied during ASR pre-training. Furthermore, when combined with strong architectures (w2v-MBART), our method pushed performance to state-of-the-art levels.

Beyond raw performance metrics, our analysis reveals two fundamental implications for ST training. First, we demonstrated that enriching the training data with diverse synthetic data from SEGAUGMENT effectively closes the performance gap between manual and automatic segmentation. This finding challenges the long-standing assumption that the manual segmentation constitutes an unbreakable upper bound for translation quality. Second, we disentangled the mechanisms behind these improvements, identifying that they are driven not merely by the target-side benefits of Knowledge Distillation, but crucially by the contextual and positional diversity introduced by re-segmenting the source audio. By presenting the model with dynamic views of the input, SEGAUGMENT prevents overfitting to specific sentence structures, resulting in a more robust and flexible translation system.

Part III

Cross-Modal Alignment

5 Zero-Shot Speech Translation via Cross-Modal Alignment

This chapter is based on our work “Pushing the Limits of Zero-shot End-to-End Speech Translation”, published in **ACL Findings 2024** (Tsiamas et al., 2024a).

5.1 Introduction

In Part II of the thesis, we focused on optimizing the processing of speech inputs to maximize the utility of existing Speech Translation (ST) datasets. We addressed the segmentation mismatch with SHAS (Ch. 3) and then utilized our segmentation model in a data augmentation strategy with SEGAUGMENT (Ch. 4). However, these methods still operate under the assumption that some paired speech-to-translation data is available for training.

A more fundamental bottleneck for the widespread adoption of End-to-End (E2E) ST is the scarcity of paired data (Sperber and Paulik, 2020). While Automatic Speech Recognition (ASR) and Machine Translation (MT) benefit from massive amounts of transcribed speech and parallel text respectively, high-quality ST corpora are rare and limited in size. This scarcity forces state-of-the-art ST methods to rely on pre-training or multi-task learning strategies that leverage external ASR and MT data (Anastasopoulos and Chiang, 2018; Bansal et al., 2019; Pino et al., 2019).

This naturally leads to the question: *Can we build an E2E ST system without any paired ST data at all?*

This setting, known as *Zero-Shot Speech Translation*, requires bridging the gap between the audio and text modalities using only disjoint ASR and MT resources. The challenge here is twofold. First, there is a *representation mismatch*: speech encoders output continuous, acoustic features, while text encoders operate in discrete, semantic embedding spaces. Second, there is a *length mismatch*: acoustic sequences are typically an order of magnitude

longer than their corresponding text translations (e.g., thousands of frames vs. dozens of subwords).

Prior work has attempted to mitigate these challenges by optimizing distance metrics (e.g., Euclidean, Cosine, or Contrastive) to bring speech and text representations closer (Liu et al., 2020b; Ye et al., 2022; Fang et al., 2022). However, these methods often require at least a small amount of paired ST data to anchor the alignment or fail to achieve competitive performance compared to supervised baselines.

In this chapter, we introduce ZEROSWOT (Zero-Shot ST with Subword Compression and Optimal Transport), a method that effectively bridges the modality gap without any paired ST data. Our approach aligns a trainable speech encoder (e.g., WAV2VEC 2.0 (Baevski et al., 2020)) with the fixed embedding space of a massively multilingual MT model (e.g., NLLB (NLLB, 2024)). We achieve this by leveraging two key mechanisms:

- **Optimal Transport (OT):** We use the Wasserstein distance (Frogner et al., 2015; Gabriel and Marco, 2019) to minimize the discrepancy between the distribution of speech features and text embeddings. Unlike point-wise metrics, OT naturally handles sequences of different lengths and provides a more robust alignment signal.
- **Learned Subword Compression:** We propose a novel compression adapter that uses Connectionist Temporal Classification (CTC) (Graves et al., 2006) to group acoustic frames into segments that correspond to the subword tokenization of the MT model. This directly resolves the length mismatch and allows the speech encoder to mimic the tokenized input expected by the MT system.

The contributions of this chapter are as follows:

- We propose ZEROSWOT, a framework for zero-shot ST that seamlessly connects a speech encoder into a pre-trained multilingual MT model without any paired ST data.
- We introduce a lightweight Subword Compression mechanism that learns to segment speech into variable-length units matching the specific subword vocabulary of the target MT model.
- We demonstrate that our method significantly outperforms previous zero-shot ap-

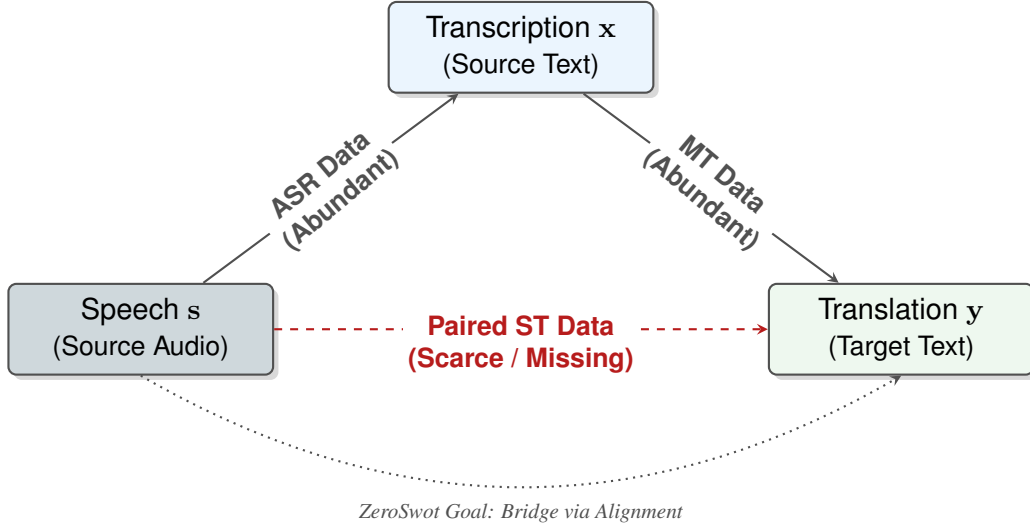


Figure 5.1: The data scarcity challenge. While ASR (Speech $\rightarrow$ Source Text) and MT (Source Text $\rightarrow$ Target Text) data are abundant, paired ST (Speech $\rightarrow$ Target Text) data is scarce. ZEROSWOT aims to bridge the dotted line by aligning the speech and text modalities using only the available solid links.

proaches on the MuST-C and CoVoST 2 benchmarks. Notably, when scaled up, ZEROSWOT achieves state-of-the-art results on MuST-C, surpassing even strong supervised baselines that use paired ST data, and is competitive on CoVoST 2 with SEAMLESSM4T which utilized 400K hours of audio data (Barrault et al., 2025).

- We show that although our model resembles a “soft cascade” of a CTC encoder and an MT encoder-decoder, it is vastly superior to standard cascades both in terms of performance and efficiency.
- We provide extensive analysis showing that our method effectively closes the modality gap in terms of both sequence length and representation similarity.
- We open-source our training code¹ and models² to facilitate further research.

The remainder of this chapter is organized as follows. §5.2 details the relevant works and their limitations. §5.3 formulates the task of zero-shot ST. §5.4 details how we solve the task using the proposed ZEROSWOT framework, including the compression adapter and the optimal transport loss. §5.5 describes the experimental setup. §5.6 presents the main results and comparisons with cascades and supervised models. Finally, §5.7 provides evidence that our method successfully bridges the modality gap, and §5.8 concludes this chapter

¹github.com/mt-upc/zeroswot

²huggingface.co/collections/johntsi/zeroswot

and discusses the limitations of subword-level alignment, motivating the character-level approach explored in Ch. 6.

5.2 Related Works: Advancements and Limitations

The development of ZEROSWOT sits at the intersection of three active research streams: strategies to mitigate data scarcity in E2E ST, methods for aligning multimodal representations, and the pursuit of zero-shot translation.

5.2.1 Strategies for Data Scarcity

The scarcity of paired ST data has long been the primary bottleneck for end-to-end models. To overcome this, the field has traditionally relied on leveraging the abundant resources available for the component tasks: Automatic Speech Recognition (ASR) and Machine Translation (MT).

Early approaches focused on transfer learning, where the ST model is initialized via pre-training on ASR or MT corpora (Bérard et al., 2016; Bansal et al., 2019; Wang et al., 2020b; Alinejad and Sarkar, 2020). This was often complemented by multi-task learning, where the model is trained jointly on ST, ASR, and MT objectives to induce a shared representation (Anastasopoulos and Chiang, 2018; Indurthi et al., 2021). Beyond architectural modifications, data augmentation has played a crucial role, particularly through the generation of synthetic ST data from MT and ASR corpora (Jia et al., 2019; Pino et al., 2019), or via Knowledge Distillation (KD), where the ST model is trained to mimic the output distribution of a strong MT teacher (Liu et al., 2019; Gaido et al., 2020a; Hinton et al., 2015).

Recently, the field has shifted toward utilizing Foundation Models (FMs) (Bommasani et al., 2022) and Large Language Models (LLMs) (Brown et al., 2020). Rather than training from scratch, modern methods initialize the ST architecture with large, pre-trained acoustic encoders (e.g., WAV2VEC 2.0 (Baevski et al., 2020)) and massively multilingual MT models (e.g., MBART-50 (Liu et al., 2020a), NLLB (NLLB, 2024)) or LLMs (e.g., LLAMA-2 (Touvron et al., 2023)). These models bring robust internal representations

learned from vast amounts of monolingual data, allowing systems to achieve state-of-the-art results even with limited paired supervision (Li et al., 2021; Ye et al., 2021; Han et al., 2021; Tsiamas et al., 2022b; Zhang et al., 2023b). Our work builds upon this foundation, asking how we can connect these powerful pre-trained modules without any supervision at all.

5.2.2 Addressing the Modality Gap

Utilizing pre-trained text models for speech tasks requires overcoming the modality gap. Approaches to this problem generally fall into two categories: implicit and explicit alignment.

Representation Alignment Strategies. Implicit methods rely on the model learning to bridge the gap during training, often facilitating this process through a shared semantic encoder in a multi-tasking framework (Liu et al., 2020b; Ye et al., 2021; Tang et al., 2021b). This can be further refined using mixup strategies (Fang et al., 2022; Zhou et al., 2023) that interpolate between speech and text features. Explicit alignment, conversely, directly optimizes a distance metric to force speech and text representations closer in the shared space. While some works have employed Euclidean distance (Liu et al., 2020b; Tang et al., 2021a) or contrastive learning (Ye et al., 2022; Ouyang et al., 2023), these metrics often struggle with the sequential nature of the data. To compute a distance, they typically require collapsing the sequence into a single vector (e.g., via mean-pooling), which destroys the temporal structure and limits the granularity of the alignment. In contrast, our approach utilizes Optimal Transport, a metric that operates directly on the sequence level, preserving temporal fidelity.

Handling Length Mismatch. Ideally, the speech representation should not only match the text in embedding space but also in sequence length. Standard methods achieve this via convolutional length adapters (Li et al., 2021; Gállego et al., 2021; Fang et al., 2022; Zhao et al., 2022). Although they reduce the length discrepancies, they are too rigid, employing a fixed reduction ratio to all speech sequences. Another line of works utilizes CTC-based compression, aiming to collapse the speech representations into units that resemble characters or phonemes (Liu et al., 2020b; Gaido et al., 2021a; Xu et al., 2021a). A major drawback of such methods is that the text representations of modern models are

based on subwords (Kudo and Richardson, 2018), and thus the granularity of the speech representation after CTC compression is not aligned with their textual counterparts. While some works utilize phonemized text to bridge this gap (Tang et al., 2021a, 2022; Le et al., 2023), this introduces conversion noise, reducing the quality of the translations. A key contribution of our work is the introduction of a novel CTC-based compression that aims to compress speech representations into units that resemble subwords, directly aligning the acoustic stream with the specific tokenization of the target MT model.

5.2.3 Advances in Zero-Shot Speech Translation

The ultimate test of cross-modal alignment is zero-shot ST, which involves translating speech without ever seeing paired ST data.

Escolano et al. (2021b) pioneered this direction by coupling a speech encoder with an MT model via language-specific bottlenecks (Escolano et al., 2021a). Subsequent works attempted to force a shared representation space: Dinh et al. (2022) used auxiliary Euclidean losses, T-modules (Duquenne et al., 2022) leveraged mined data to learn a fixed-size multimodal space, and DCMA (Wang et al., 2022) utilized an attention-driven memory module to align codebook distributions. More recently, Gao et al. (2024) applied cross-lingual consistency regularization (Gao et al., 2023) to align WAV2VEC 2.0 representations with MT encoders.

The closest precursor to our method is Yang et al. (2023), which aligns a speech encoder to a bilingual MT model using Optimal Transport and a CTC module. However, their approach has a critical limitation: the CTC module must share the exact vocabulary of the MT model. While feasible for bilingual models with small vocabularies, this is computationally prohibitive for massively multilingual models like NLLB (250k+ tokens). We overcome this by decoupling the compression from the vocabulary size. Our method learns the segmentation of the MT model without needing to predict its vast vocabulary, ensuring both efficiency and scalability to hundreds of languages.

5.3 Problem Formulation for Zero-shot Speech Translation

We define the task of Speech Translation (ST) as finding the most probable target translation $\mathbf{y}$ given a source speech sequence $\mathbf{s}$. In the standard supervised setting, this is achieved by maximizing the log-likelihood over a dataset of paired examples $\mathcal{D}_{ST} = \{(\mathbf{s}, \mathbf{y})_i\}_{i=1}^{|\mathcal{D}_{ST}|}$:

$$\hat{\theta} = \operatorname{argmax}_{\theta} \sum_{(\mathbf{s}, \mathbf{y}) \in \mathcal{D}_{ST}} \log P(\mathbf{y} | \mathbf{s}; \theta) \quad (5.1)$$

where θ represents the parameters of an end-to-end model.

In the zero-shot scenario, we assume that the paired dataset $\mathcal{D}_{ST}$ is empty or unavailable ($\mathcal{D}_{ST} = \emptyset$). Instead, we only have access to two disjoint datasets:

1. An ASR corpus $\mathcal{D}_{ASR} = \{(\mathbf{s}, \mathbf{x})_i\}$, consisting of speech paired with its source language transcription $\mathbf{x}$.
2. A Multilingual MT corpus $\mathcal{D}_{MT} = \{(\mathbf{x}, \mathbf{y})_j\}$, consisting of source text paired with translations in various target languages.

Our objective is to construct an ST model $P(\mathbf{y} | \mathbf{s})$ using only $\mathcal{D}_{ASR}$ and $\mathcal{D}_{MT}$. While a cascaded approach approximates this via $P(\mathbf{y} | \mathbf{x}) \cdot P(\mathbf{x} | \mathbf{s})$, our goal is to build a single differentiable model that translates directly from $\mathbf{s}$ to $\mathbf{y}$ at inference time.

5.3.1 The Alignment Objective

To achieve this without supervision, we adopt a representation alignment strategy. We assume access to a pre-trained, multilingual MT model parameterized by ϕ , which encodes source text $\mathbf{x}$ into a sequence of continuous embeddings $\mathbf{H}^x \in \mathbb{R}^{N \times d}$ and decodes them into $\mathbf{y}$. Since this MT model is already capable of high-quality translation (i.e., $P_{\phi}(\mathbf{y} | \mathbf{H}^x)$ is reliable), we aim to re-use it for speech.

We introduce a speech encoder parameterized by θ , which maps the input audio $\mathbf{s}$ to a sequence of acoustic representations $\mathbf{H}^s \in \mathbb{R}^{T \times d}$. The zero-shot learning problem then

reduces to finding the optimal parameters θ such that the speech representations become indistinguishable from the text representations in the shared semantic space:

$$\mathbf{H}^s \approx \mathbf{H}^x \quad (5.2)$$

If this condition holds, we can feed $\mathbf{H}^s$ into the frozen MT decoder to generate translations, effectively treating speech as if it were text.

5.3.2 Dimensionality Constraints

The primary difficulty in solving $\mathbf{H}^s \approx \mathbf{H}^x$ arises from the dimensional mismatch between the modalities.

- **Sequence Length:** The speech representation $\mathbf{H}^s$ has length T (proportional to the duration of audio in frames), while the text representation $\mathbf{H}^x$ has length N (the number of subwords). Typically, $T \gg N$.
- **Distribution:** The speech vectors lie in a manifold determined by acoustic properties, whereas the text vectors lie in a manifold determined by linguistic semantics.

Consequently, a simple frame-to-token mapping is impossible. We require a mapping function $\mathcal{M}$ that compresses $\mathbf{H}^s$ to match the granularity of $\mathbf{H}^x$ while minimizing the distributional divergence between them. In the following section, we present our methodology for realizing this mapping through Learned Subword Compression and Optimal Transport.

5.4 The ZEROSWOT Methodology

To address the zero-shot challenge defined in §5.3, we propose ZEROSWOT, a framework designed to align a pre-trained acoustic encoder with the embedding space of a massive multilingual MT model. Our approach achieves this alignment without paired ST data by leveraging a novel subword compression mechanism and an Optimal Transport training objective.

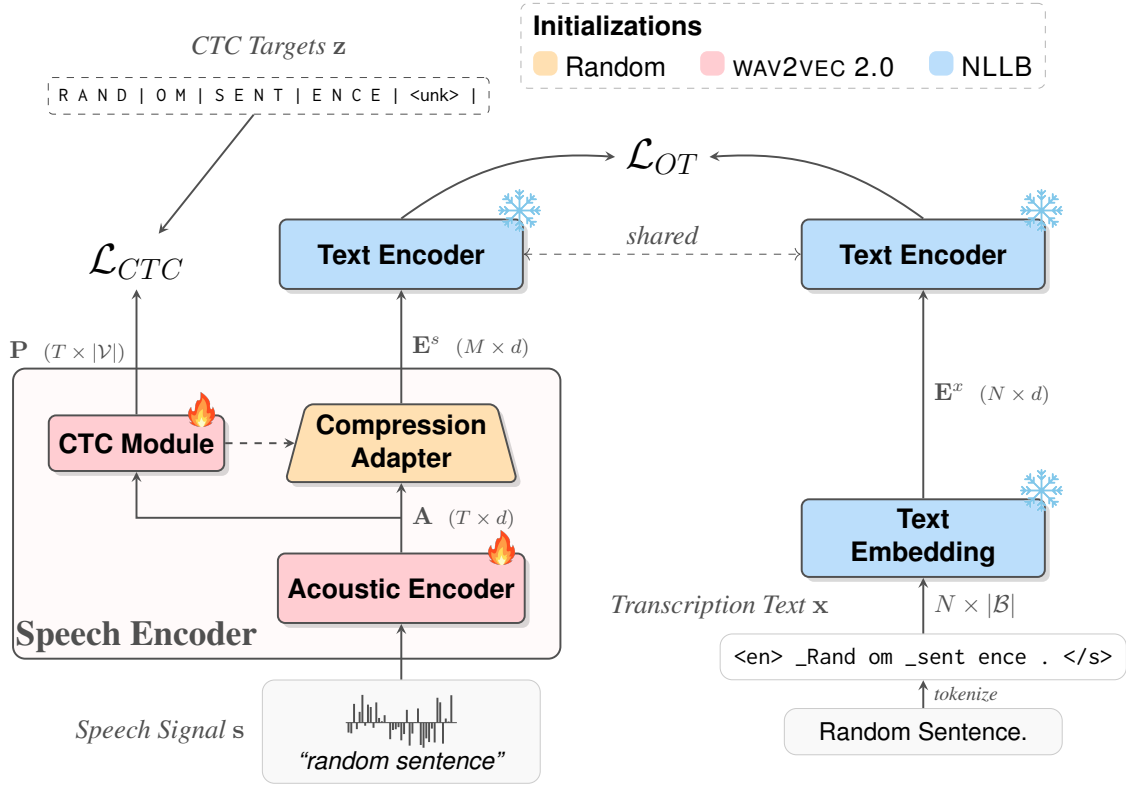


Figure 5.2: The ZEROSWOT architecture. The Text Branch (right) utilizes a frozen NLLB Text Encoder to define the target semantic space. The Speech Branch (left) processes audio via WAV2VEC 2.0 and a Compression Adapter, followed by the (shared) Text Encoder. The Optimal Transport loss ensures that the speech representation is aligned with the text one. The model is trained using only ASR speech-transcription pairs.

5.4.1 Model Architecture

The model follows a Siamese architecture comprised of a *Speech Branch* (trainable) and a *Text Branch* (frozen). The Text Branch serves as the semantic anchor, providing the target representation space. The Speech Branch is designed to project raw audio into this anchor space. Crucially, the final Semantic Encoder is shared between both branches, ensuring that speech representations are mapped into the same high-level abstraction as text.

The overall architecture is illustrated in Figure 5.2. We detail the specific components below.

Text Encoder. The text encoder serves as the foundation of our zero-shot approach. We utilize the encoder of a massively multilingual MT model, specifically NLLB (NLLB,

2024), which we keep frozen during training. As discussed in §2.5.2, this model is capable of handling over 200 languages.

Given a tokenized source text sequence $\mathbf{x}$ of length N ,³ we embed it using the TEXT-EMBEDDING module. We first map to continuous vectors using an embedding layer $\mathbf{W}^{emb} : |\mathcal{B}| \rightarrow d$, where $\mathcal{B}$ is the subword vocabulary, then scale the embeddings, and finally add sinusoidal positional encodings (Vaswani et al., 2017) (see §2.2.3):

$$\mathbf{E}^x = \text{TEXT-EMBEDDING}(\mathbf{x}) = \sqrt{d} \cdot \mathbf{x} \mathbf{W}^{emb} + \text{POS}(N) \quad (5.3)$$

This sequence is processed by the Transformer encoder, denoted here as TEXT-ENCODER, to obtain the semantic representations $\mathbf{H}^x \in \mathbb{R}^{N \times d}$:

$$\mathbf{H}^x = \text{TEXT-ENCODER}(\mathbf{E}^x) \quad (5.4)$$

Since the TEXT-ENCODER is already optimized for translation, our goal reduces to training the speech branch to produce outputs $\mathbf{H}^s$ that match the distribution of $\mathbf{H}^x$.

Acoustic Encoder. The acoustic encoder, ACOUSTIC-ENCODER, processes the raw speech signal $\mathbf{s} \in \mathbb{R}^L$ into a sequence of acoustic features $\mathbf{A} \in \mathbb{R}^{T \times d}$. We employ the WAV2VEC 2.0 architecture (Baevski et al., 2020) (described in §2.3.3), which consists of a multi-layer convolutional feature extractor followed by a stack of Transformer encoder layers. The model is initialized from a pre-trained WAV2VEC 2.0 checkpoint and finetuned.

$$\mathbf{A} = \text{ACOUSTIC-ENCODER}(\mathbf{s}) \quad (5.5)$$

Target Label Construction. A critical divergence from standard ASR training is the definition of the target labels $\mathbf{z}$. Standard ASR targets typically consist of words with punctuation removed. However, our text branch expects subwords and preserves punctuation. To minimize this discrepancy, we construct the targets $\mathbf{z}$ by tokenizing the transcription using the text branch tokenizer (SentencePiece). We explicitly insert a separator token `<sep>` (e.g., “|”) between subwords. Crucially, the resulting target sequence $\mathbf{z}$ is composed of the constituent *characters* of these subwords, separated by the boundary marker. Characters not present in the acoustic vocabulary are mapped to `<unk>`. Table 5.1 contrasts standard

³Including the prepend language token `<lang>` and appended end-of-sentence token `</s>` (NLLB, 2024).

Tokenization	Unknowns	CTC Labels Example
Word-level	✗	R A N D O M S E N T E N C E
Subword-level	✓	R A N D O M S E N T E N C E <unk>

Table 5.1: Comparison of CTC target construction. Unlike standard ASR which targets words, we target subword units separated by explicit boundary markers <sep> (here denoted as |) to align with the MT tokenizer.

ASR labels with our proposed format. This preparation is essential for the Subword Compression mechanism described below.

Connectionist Temporal Classification (CTC). We employ a CTC module (Graves et al., 2006) (see §2.4.2) to enforce linguistic content in the acoustic representations. More importantly, the CTC predictions provide the alignment boundaries necessary for compression. The module projects the features $\mathbf{A}$ to probabilities $\mathbf{P} \in \mathbb{R}^{T \times |\mathcal{V}|}$ over a small, letter-based vocabulary $\mathcal{V}$:

$$\mathbf{P} = \text{softmax}(\mathbf{A}\mathbf{W}^{ctc} + \mathbf{b}^{ctc}) \quad (5.6)$$

where $\mathbf{W}^{ctc} \in \mathbb{R}^{d \times |\mathcal{V}|}$ and $\mathbf{b}^{ctc} \in \mathbb{R}^{|\mathcal{V}|}$ are trainable parameters, initialized from the CTC layer of the pretrained WAV2VEC 2.0.

We optimize the standard CTC loss $\mathcal{L}_{\text{CTC}}$ calculated against our subword-aligned target sequence $\mathbf{z}$.

Compression Adapter. The Compression Adapter is the core component bridging the length mismatch discussed in §5.3. It transforms the frame-level acoustic sequence $\mathbf{A}$ into a sequence $\mathbf{E}^s$ that resembles the text embedding $\mathbf{E}^x$, utilizing the CTC probabilities $\mathbf{P}$:

$$\mathbf{E}^s = \text{COMPRESSION-ADAPTER}(\mathbf{A}, \mathbf{P}) \quad (5.7)$$

The process operates in two stages, as illustrated with an example in Figure 5.3, and is detailed below.

1. **CTC-based Character Compression:** We first reduce sequence redundancy using standard CTC compression (Gaido et al., 2021a). We obtain discrete labels $\pi_t =$

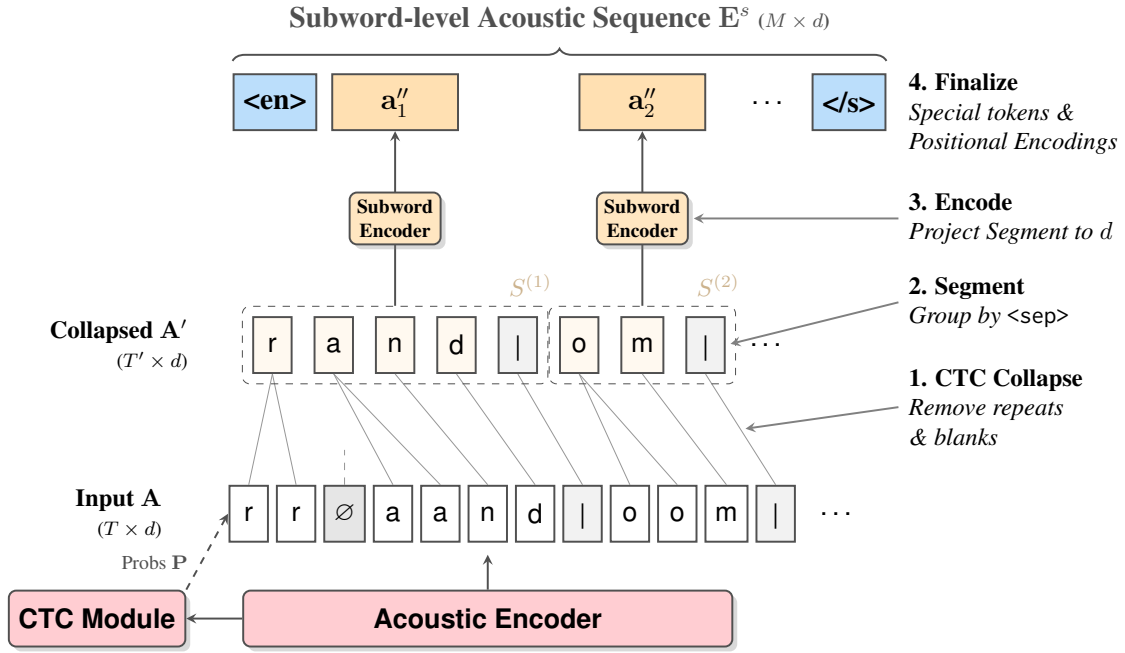


Figure 5.3: The Compression Adapter. The module transforms the high-frequency acoustic sequence $\mathbf{A}$ into a text-aligned representation $\mathbf{E}^s$. The process proceeds in four stages: (1) **CTC Collapse** uses the predicted probabilities to remove redundant frames and blanks; (2) **Segmentation** groups the remaining frames into subword units $S^{(i)}$ based on separator tokens; (3) **Encoding** projects each segment into a semantic vector via a BERT-style Transformer block; and (4) **Finalize** appends special tokens and adds positional encodings to match the structure of the text embeddings.

$\text{argmax}(\mathbf{p}_t)$, group consecutive frames with identical labels, and mean-pool them, discarding blanks ($\langle \text{blank} \rangle$). This yields the character-level acoustic sequence $\mathbf{A}' \in \mathbb{R}^{T' \times d}$.

- 2. Character-to-Subword Compression:** We then aggregate the acoustic character-level sequence into a subword-level one. The sequence $\mathbf{A}'$ is split into segments $S^{(1)}, \dots, S^{(T')}$ using the positions of the separator token $\langle \text{sep} \rangle$ (which we treat as the boundary).⁴ Each segment $S^{(i)} \in \mathbb{R}^{k_i \times d}$ represents the variable-length acoustic span of a single subword. To convert this into a fixed-size vector $\mathbf{a}_i'' \in \mathbb{R}^d$, we process the segments with a small BERT-style Transformer encoder (Devlin et al.,

⁴Note that the separators are not discarded, but included in each segment, to avoid any potential information loss.

2019):⁵

$$\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(T'')} = \text{segment}(\mathbf{A}''|\mathbf{P}) \quad (5.8)$$

$$\mathbf{a}_i'' = \text{SUBWORD-ENCODER}(\mathbf{S}^{(i)}) \quad (5.9)$$

Note that each segment is processed independently from the rest. We concatenate the $\{\mathbf{a}_i''\}_1^{T''}$ vectors into a sequence, while also prepending and appending the special embeddings for $\langle \text{lang} \rangle$ and $\langle /s \rangle$ from the text branch. We then mimic the function of the TEXT-EMBEDDING module of Equation (5.3), thus scaling and adding positional encodings:

$$\mathbf{E}^s = \sqrt{d} \cdot [\mathbf{e}^{\langle \text{lang} \rangle}, \mathbf{a}_1'', \dots, \mathbf{a}_{T''}'', \mathbf{e}^{\langle /s \rangle}] + \text{Pos}(M) \quad (5.10)$$

where $\mathbf{E}^s \in \mathbb{R}^{M \times d}$, with $M = T'' + 2 \approx N$, now matches the granularity of the text embedding space $\mathbf{E}^x$ (i.e., $M \approx N$).

Shared Semantic Encoder. Finally, this sequence of subword-level acoustic embeddings is passed through the *shared* TEXT-ENCODER:

$$\mathbf{H}^s = \text{TEXT-ENCODER}(\mathbf{E}^s) \quad (5.11)$$

By reusing the parameters of the pre-trained TEXT-ENCODER for the speech branch, we ensure that if the input embeddings $\mathbf{E}^s$ are successfully aligned with $\mathbf{E}^x$, the resulting high-level features $\mathbf{H}^s$ will be immediately compatible with the MT decoder.

The supervision from the CTC loss ensures that the compression adapter will create an acoustic representation that is similar in length with the textual one. In the following section, we present how we further align the two representations in terms of semantics.

5.4.2 Optimal Transport Alignment

The core objective of ZEROSWOT is to force the speech representations $\mathbf{H}^s \in \mathbb{R}^{M \times d}$ to reside in the same semantic space as the text representations $\mathbf{H}^x \in \mathbb{R}^{N \times d}$. Although

⁵We prepend a learnable $\langle \text{cls} \rangle$ vector $\mathbf{e}^{\langle \text{cls} \rangle} \in \mathbb{R}^d$ to the processed segment and treat its output as the final subword representation.

our Compression Adapter minimizes the length discrepancies between these sequences ($M \approx N$), we cannot always ensure one-to-one alignment and thus, cannot rely on element-wise distance metrics like Mean Squared Error or Cosine Similarity. Instead, we frame the alignment as a distribution matching problem.

We model the speech and text sequences as discrete probability distributions over the shared embedding space $\mathbb{R}^d$. Formally, we define:

$$\mu = \sum_{i=1}^M u_i \delta(\mathbf{h}_i^s), \quad \nu = \sum_{j=1}^N v_j \delta(\mathbf{h}_j^x) \quad (5.12)$$

where $\delta(\omega)$ represents a Dirac delta function centered at location ω in the d -dimensional embedding space.⁶ The weights $\mathbf{u} \in \Delta_M$ and $\mathbf{v} \in \Delta_N$ represent the probability mass of each token. Following standard practice for sequence alignment (Grave et al., 2019), we assume uniform masses, setting $u_i = 1/M$ and $v_j = 1/N$, implying that each subword unit holds equal semantic weight for the alignment, regardless of its acoustic duration.

The Wasserstein Distance. We quantify the discrepancy between μ and ν using the Wasserstein distance (also known as the Earth Mover’s Distance). Intuitively, this measures the minimum “work” required to transform the speech distribution into the text distribution. The cost of moving mass from a speech vector $\mathbf{h}_i^s$ to a text vector $\mathbf{h}_j^x$ is defined by the cost matrix $\mathbf{C} \in \mathbb{R}_+^{M \times N}$. We employ the squared Euclidean distance:

$$\mathbf{C}_{ij} = \|\mathbf{h}_i^s - \mathbf{h}_j^x\|_2^2 \quad (5.13)$$

The alignment problem is then formulated as finding a transport plan $\Gamma \in \mathbb{R}_+^{M \times N}$ that minimizes the total transport cost:

$$\mathcal{L}_{OT} = \min_{\Gamma \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i=1}^M \sum_{j=1}^N \Gamma_{ij} \mathbf{C}_{ij} = \min_{\Gamma \in \Pi(\mathbf{u}, \mathbf{v})} \langle \Gamma, \mathbf{C} \rangle \quad (5.14)$$

subject to the marginal constraints $\Gamma \mathbf{1}_N = \mathbf{u}$ and $\Gamma^\top \mathbf{1}_M = \mathbf{v}$. These constraints ensure that the total mass from the speech sequence is fully transported to the text sequence, preserving the information content.

⁶The Dirac delta δ_ω is a measure that assigns a probability mass of 1 to the specific point ω and 0 elsewhere. This formulation allows us to model the sequence of continuous embedding vectors as a discrete empirical distribution.

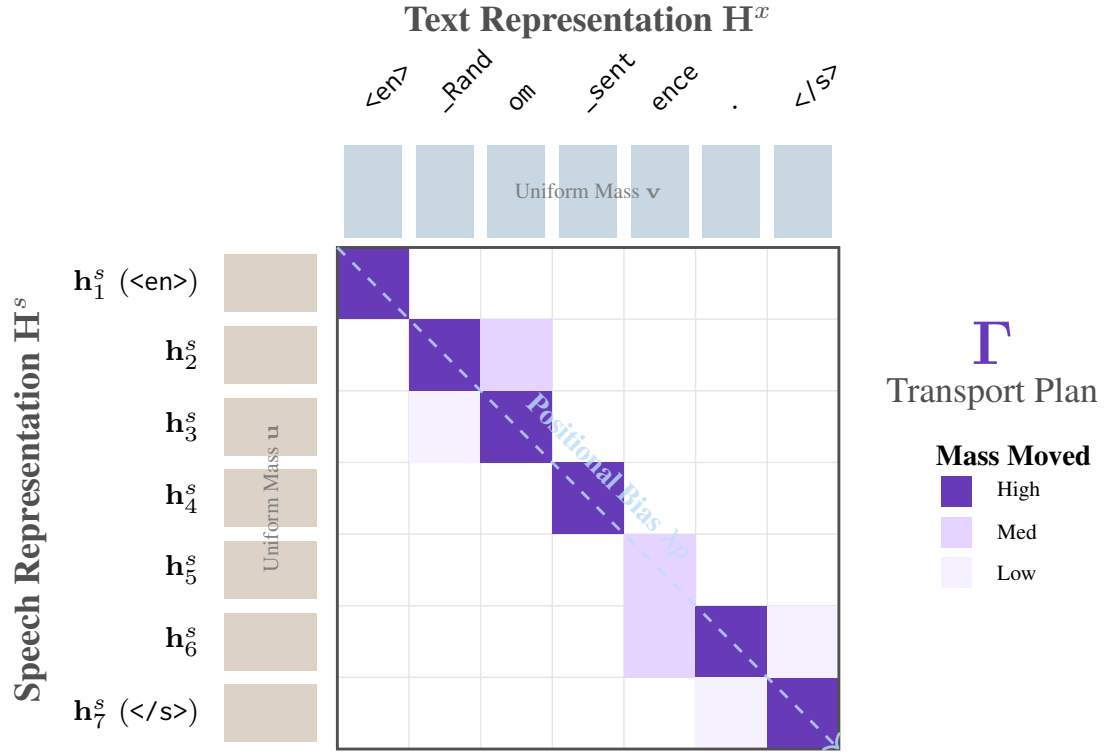


Figure 5.4: Visualization of the Optimal Transport plan Γ between the compressed speech representation H^s and the text representation H^x . The optimization finds a “soft” alignment where Γ_{ij} represents the mass transported from acoustic subword i to text token j . The diagonal tendency is encouraged by the positional regularization, while the marginal bars (grey) illustrate the global constraint that the total probability mass from the source (u) and target (v) distributions must be fully preserved.

Positional Regularization. A fundamental limitation of the standard Wasserstein distance is its permutation invariance; it treats the sequences as “bags of vectors” and ignores the temporal order. To enforce monotonicity (ensuring that the beginning of the speech signal aligns with the beginning of the text) we augment the representations with explicit positional information. Following [Le et al. \(2023\)](#), we concatenate a scaled positional scalar to each embedding vector. Let $p_i^s = \frac{i-1}{M-1}$ and $p_j^x = \frac{j-1}{N-1}$ be the relative positions of the speech and text tokens. Thus, the position-regularized cost matrix $\bar{C}$ is computed on the augmented vectors:

$$\bar{\mathbf{h}}_i^s = [\mathbf{h}_i^s; \lambda p_i^s], \quad \bar{\mathbf{h}}_j^x = [\mathbf{h}_j^x; \lambda p_j^x] \quad (5.15)$$

where $\lambda > 0$ is a hyperparameter controlling the strength of the regularization. This penalizes transport plans that map vectors from distant relative positions (e.g., mapping the first speech segment to the last text token), effectively biasing the solution towards the diagonal.

Entropy Regularization and Sinkhorn Algorithm. Solving the exact linear programming problem for OT is computationally expensive ($O(N^3 \log N)$). To integrate this objective into a differentiable training loop, we use the Entropy-Regularized Wasserstein distance:

$$\mathcal{L}_{OT} = \min_{\Gamma \in \Pi(\mathbf{u}, \mathbf{v})} \langle \Gamma, \bar{\mathbf{C}} \rangle - \epsilon H(\Gamma) \quad (5.16)$$

where $H(\Gamma)$ is the Shannon entropy of the transport plan and $\epsilon > 0$ is the regularization coefficient. This formulation makes the problem strictly convex and solvable via the Sinkhorn-Knopp algorithm (Cuturi, 2013), which consists of efficient iterative matrix scaling operations.

5.4.3 Combined Training Objective

The complete training process optimizes both the alignment of the representations and the linguistic content of the acoustic encoder.

The parameters of the acoustic encoder and the CTC module are supervised by a CTC loss $\mathcal{L}_{CTC}$ using the labels $\mathbf{z}$ (see the example in Table 5.1), ensuring that acoustic content is preserved and that the compression adapter receives high-quality CTC probabilities $\mathbf{P}$ to carry out the two-step compression process.

The parameters of the acoustic encoder and the compression adapter are supervised by several Wasserstein losses (Equation (5.16)) to drive the semantic alignment. We apply the Wasserstein loss not only to the final output of the shared encoder but also to a set of intermediate layers $\mathcal{K}$, forcing the speech branch to mimic the processing hierarchy of the text branch.

The final loss function is a weighted sum:

$$\mathcal{L} = \alpha \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathcal{L}_{OT}^{(k)} + (1 - \alpha) \mathcal{L}_{CTC} \quad (5.17)$$

where $\alpha > 0$ is a hyperparameter to balance the importance of the two objectives. By minimizing this objective using only ASR the model learns to perform Speech Translation in a zero-shot manner.

Zero-shot ST Inference. The inference pipeline is illustrated in Figure 5.5. To enable zero-shot ST, we effectively replace the Text Embedding layer of the translation model with our finetuned Speech Encoder (comprising the Acoustic Encoder and Compression Adapter). Given a raw speech signal s , the Speech Encoder generates a sequence of discrete, text-aligned representations E^s . Since these are structurally compatible with the text embedding space, they are passed directly to the frozen Text Encoder. The Text Encoder processes them as if they were text, and the resulting contextual representations are passed to the Text Decoder to autoregressively generate the translation $\hat{y}$.

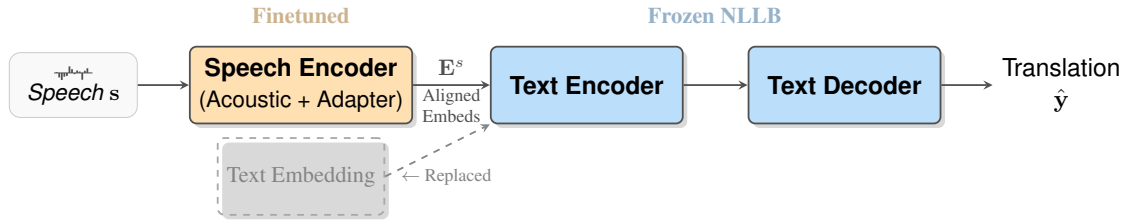


Figure 5.5: Zero-Shot ST Inference Workflow. The finetuned Speech Encoder acts as a plug-and-play replacement for the source Text Embedding. It produces text-aligned representations E^s that are aligned in length and semantics with the text embedding space, allowing the frozen Text Encoder-Decoder to generate translations directly from speech input.

5.5 Experimental Setup

In this section, we describe the datasets, model configurations, and training details used to evaluate ZEROSWOT. We assess our method on three multilingual benchmarks: MuST-C (Di Gangi et al., 2019a), CoVoST 2 (Wang et al., 2020c), and FLEURS (Conneau et al., 2023).

5.5.1 Datasets

For the training of the speech encoder, we strictly adhere to the zero-shot constraint: we do not use any paired speech-translation (ST) data (audio paired with foreign translations). Instead, we rely on Automatic Speech Recognition (ASR) data (audio paired with transcripts). We optionally utilize Machine Translation (MT) data to adapt the NLLB model to the specific (e.g., MuST-C) via standard fine-tuning, before freezing it to train the speech alignment.

Dataset	Examples	Hours	Avg. Length (s)
Common Voice	949K	1,503	5.7
MuST-C (ASR)	352K	681	7.0

Table 5.2: Statistics of the ASR datasets used for ZEROSWOT training.

ASR Training Data. We utilize the following English ASR corpora for training our speech encoder, with their data statistics being available at Table 5.2.

- **MuST-C ASR:** We create this by using the speech-transcription pairs from the English-to-X portion of the MuST-C v1.0 train set (Di Gangi et al., 2019a).
- **Common Voice:** We use version 8.0 (Ardila et al., 2020) of the English train split.

ST Evaluation Data. We use the following corpora for zero-shot ST evaluation:

- **MuST-C:** We report results on the `tst-COMMON` split for the English→8 language pairs of v1.0.
- **CoVoST:** We report results on the `test` split of CoVoST 2 that includes 15 language pairs with source language being English.

MT Training Data. We utilize the following multilingual MT data to optionally finetune the MT model used in the cross-modal alignment training.

- **MuST-C MT:** We create this dataset by using the transcription-translation pairs from the English-to-X portion of the MuST-C v1.0 train.
- **CoVoST MT:** Similarly, this is created by concatenating the transcription-translation pairs of the English-to-X trainset of CoVoST 2.

More information on the datasets, including processing, filtering and data sizes are available in the Appendix (§B.1).

5.5.2 Model Configuration

We evaluate ZEROSWOT across different model sizes to assess scalability. The architecture consists of the trainable Speech Branch and the frozen Text Branch.

Acoustic Encoder. We employ the WAV2VEC 2.0 architecture (Baevski et al., 2020). For our primary experiments, we use the LARGE configuration (24 layers, model dimension $d = 1024$, 16 attention heads). The encoder is initialized from a checkpoint finetuned on LibriSpeech and Libri-light (Kahn et al., 2020) with a CTC loss. The CTC module uses a character vocabulary of size $|\mathcal{V}| = 30$.

Compression Adapter. The subword encoder within the adapter consists of a small BERT-style Transformer encoder with 3 layers and dimensionality $d = 1024$. It matches the feed-forward dimensions and activation functions of the acoustic encoder.

Text Branch (NLLB). For the text encoder, we leverage the NLLB models (NLLB, 2024) which support 200+ languages. Specifically, we use the distilled versions, and experiment with two sizes:

- **NLLB-600M:** Based on nllb-200-distilled-600M. The text encoder has 12 layers with dimension 1024.
- **NLLB-1.3B:** Based on nllb-200-distilled-1.3B. The text encoder has 24 layers with dimension 1024.

Both versions share a SentencePiece vocabulary of $|\mathcal{B}| = 256k$ tokens. Note that while the text encoder is frozen during training, its parameters contribute to the inference cost. Table 5.3 summarizes the parameter counts. More details on the ZEROSWOT model configuration are provided in the Appendix (§B.2.1).

Experimental Design. A key variable in our experiments is the composition of the training data used for the Speech Encoder and the initialization of the frozen Text Branch. To isolate the impact of domain adaptation, we define three distinct scenarios, summarized in Table 5.4. **Scenario A** represents a fully open-domain setting where we train the

Model	Components (Init.)	Train Params	Inference Params
SMALL	WAV2VEC 2.0 BASE NLLB-600M	110M	0.7B
MEDIUM	WAV2VEC 2.0 LARGE NLLB-600M	350M	0.95B
LARGE	WAV2VEC 2.0 LARGE NLLB-1.3B	350M	1.65B

Table 5.3: Model configurations for ZEROSWOT. “Train Params” refers to the trainable parameters in the speech branch. “Inference Params” includes the frozen text encoder and decoder required for generation.

speech encoder on Common Voice and use the off-the-shelf NLLB model. This entails a strict zero-shot condition where neither component has seen in-domain data. **Scenario B** assesses the alignment of in-domain speech (MuST-C ASR) to the generic multilingual space of the original NLLB. Finally, **Scenario C** evaluates the upper bound of our method by initializing the text branch with an NLLB model finetuned on in-domain text-to-text data, thus adapting the target semantic space to the specific domain.

Scenario	ASR Data	MT Initialization	Tests for...
A	Out-of-domain (CV)	Original NLLB	Open-domain generalization
B	In-domain (MuST-C)	Original NLLB	In-domain speech alignment
C	In-domain (MuST-C)	In-domain finetuned NLLB	Benefit of target domain adaptation

Table 5.4: Definition of the experimental data scenarios used to evaluate ZEROSWOT.

5.5.3 Training and Evaluation

We train the acoustic encoder, the CTC module, and the compression adapter, while keeping the text branch frozen. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $3 \cdot 10^{-4}$ and a tri-stage scheduler (warmup, hold, and decay). The input to the speech branch is raw waveforms sampled at 16kHz and the effective batch size is set to 32 million tokens. To prevent overfitting, we apply extensive masking to the speech features: we mask 10 consecutive time steps with a probability of 0.3 and 64 frequency channels with a probability of 0.2. To accelerate training, we extract the text encoder representations offline, allowing us to bypass the forward pass of the frozen text branch during the speech encoder optimization.

Loss Configuration. We set the weighting factor $\alpha = 0.9$ (Eq. 5.4.3), prioritizing the alignment objective while retaining the CTC loss for structural guidance. The Wasserstein loss is computed using the Sinkhorn algorithm (Cuturi, 2013). We apply the loss not only at the final layer output but also at intermediate layers to enforce hierarchical alignment.

- For the SMALL and MEDIUM model sizes, we apply losses at last layers $\mathcal{K} = \{6, 7, \dots, 12\}$.
- For the LARGE model size, we use every second layer: $\mathcal{K} = \{12, 14, \dots, 24\}$.

We set the positional regularization coefficient to $\lambda = 1$ and the entropy regularization strength to $\epsilon = 10$.

Further details on the training and hyperparameters of the ZEROSWOT models are available in the Appendix (§B.2.2).

Evaluation. We evaluate performance using the standard BLEU metric (Papineni et al., 2002) on the detokenized text, and report case-sensitive scores using sacreBLEU (Post, 2018). For inference, we average the 10 best checkpoints based on the validation loss. We use beam search with a beam size of 5. Unless otherwise stated, we report the average BLEU score across all supported language directions for each benchmark (8 languages for MuST-C, 15 for CoVoST 2, 88 for FLEURS). See the Appendix for more information (§B.5).

5.6 Experimental Results

We present our experimental evaluation of ZEROSWOT on the MuST-C v1.0 and CoVoST 2 benchmarks. We compare our approach against previously proposed zero-shot methods, state-of-the-art supervised models, and strong cascade baselines.

5.6.1 Comparison with Zero-Shot Baselines

In Table 5.5, we compare our most lightweight configuration, ZEROSWOT-SMALL, against previous state-of-the-art zero-shot methods. We report results using Scenario B to ensure

Model	Data Usage ASR MT	Params (B) Train/Infer	deu	spa	fra	ita	nld	por	ron	rus	Avg	
Previous Zero-shot Methods												
T-MODULES	✓	✓	2.0 / 2.0	23.8	27.4	32.7	24.6	22.4	26.2	24.0	25.1	25.8
DCMA	✓	✗	0.15 / 0.15	22.4	24.6	29.7	18.4	22.0	24.2	16.8	11.8	21.2
DCMA	✓	✓	0.15 / 0.15	24.0	26.2	33.1	24.1	28.3	29.2	22.2	16.0	25.4
SIMZEROCR	✓	✓	0.15 / 0.15	25.1	27.0	34.6	–	–	–	–	15.6	–
TOWARDSZEROST	✓	✗	0.10 / 0.10	23.4	26.5	33.7	–	–	–	–	–	–
TOWARDSZEROST	✓	✓	0.10 / 0.10	26.5	29.5	35.3	24.1	28.3	29.2	22.2	16.0	26.4
This work (Data Scenario B)												
ZEROSWOT-SMALL	✓	✗	0.11 / 0.70	27.3	31.7	35.8	26.8	30.9	31.6	25.3	17.8	28.4

Table 5.5: Comparison with previous Zero-Shot ST methods on MuST-C eng→xxx tst-COMMON. The Data Usage column indicates whether the method utilizes in-domain ASR (speech-transcript) or MT (text-translation) data. Results of previous methods are taken from T-MODULES (Duquenne et al., 2022), DCMA (Wang et al., 2022), SIMZEROCR (Gao et al., 2024), and TOWARDSZEROST (Yang et al., 2023). Extended results available in the Appendix (Table B.3).

a fair comparison, as all previous baselines leverage ASR and/or MT data in domain data from MuST-C. Our method significantly outperforms all prior approaches. Notably, ZEROSWOT-SMALL surpasses T-MODULES (Duquenne et al., 2022) by +2.6 BLEU and the previous best method, TOWARDSZEROST (Yang et al., 2023), by +2.0 BLEU on average. It is important to highlight the data efficiency of our approach: while methods like T-MODULES, SIMZEROCR (Gao et al., 2024), and the stronger version of TOWARDSZEROST rely on both ASR and MT data to induce alignment, our model achieves superior performance using *solely* ASR data. This stems from our direct alignment strategy, which compresses speech into the existing subword space of the translation model without requiring complex intermediate codebooks (Wang et al., 2022) or error-prone CTC predictions over large vocabularies (Yang et al., 2023).

5.6.2 Comparison with Supervised Baselines

Scaling our architecture to the MEDIUM and LARGE configurations allows us to compete directly with fully supervised End-to-End ST systems. Table 5.6 presents the results in MuST-C and Table 5.7 presents the results in CoVoST 2.

MuST-C. In Scenario B, where we use the standard NLLB model, our LARGE model already achieves an average BLEU of 31.4. This performance is notably higher than strong supervised baselines such as CHIMERA (Han et al., 2021) (+4.0 BLEU), SIAMESE-PT (Le

Model	Scen.	Params (B)	deu	spa	fra	ita	nld	por	ron	rus	Avg
<i>(Previous Works) Supervised End-to-End ST (using paired ST data)</i>											
CHIMERA	–	0.15	27.1	30.6	35.6	25.0	29.2	30.2	24.0	17.4	27.4
STEMM	–	0.15	28.7	31.0	37.4	25.8	30.5	31.7	24.5	17.8	28.4
SPEECHUT	–	0.15	30.1	33.6	41.4	27.7	31.7	<u>34.2</u>	<u>27.0</u>	18.5	29.8
SIAMESE-PT	–	0.25	27.9	31.8	39.2	27.6	32.2	33.6	26.4	19.7	30.3
CRESS	–	0.15	29.4	33.2	40.1	27.6	32.2	33.6	26.4	19.7	30.3
SIMREGCR	–	0.15	29.2	33.0	40.0	<u>28.2</u>	<u>32.7</u>	<u>34.2</u>	26.7	<u>20.1</u>	<u>30.5</u>
LST-13B	–	13.0	<u>30.4</u>	<u>35.3</u>	41.6	–	–	–	–	–	–
<i>(This Work) Zero-shot ST</i>											
ZEROSWOT-MEDIUM	A	0.35 / 0.95	24.8	30.0	32.6	24.1	28.6	28.8	22.9	16.4	26.0
ZEROSWOT-MEDIUM	B	0.35 / 0.95	30.0	33.1	37.5	28.2	32.3	32.9	26.0	18.7	29.6
ZEROSWOT-MEDIUM	C	0.35 / 0.95 [†]	<u>30.5</u>	34.9	39.4	<u>30.6</u>	<u>35.0</u>	<u>37.1</u>	<u>27.8</u>	<u>20.3</u>	<u>31.9</u>
ZEROSWOT-LARGE	A	0.35 / 1.65	26.5	31.1	33.5	25.4	29.9	30.6	24.3	18.0	27.4
ZEROSWOT-LARGE	B	0.35 / 1.65	<u>30.1</u>	34.8	38.9	<u>29.8</u>	<u>34.4</u>	<u>35.3</u>	<u>27.6</u>	<u>20.4</u>	<u>31.4</u>
ZEROSWOT-LARGE	C	0.35 / 1.65 [†]	31.2	35.8	40.5	31.4	36.3	38.3	28.0	21.5	32.9

Table 5.6: Comparison with Supervised ST Baselines on MuST-C eng→ xxx tst-COMMON. The Params column indicates Trainable / Inference parameters (in Billions). The Scenario column refers to the data configuration (A: Out-of-Domain, B: In-Domain ASR, C: In-Domain ASR/MT). Underlined is the best supervised score, highlighted are all zero-shot scores that are better than the best supervised ones, and in **bold** is best overall. [†] NLLB parameters are finetuned separately. Results of previous methods taken from CHIMERA (Han et al., 2021), STEMM (Fang et al., 2022), SPEECHUT (Zhang et al., 2022b), SIAMESE-PT (Le et al., 2023), CRESS (Fang and Feng, 2023), SIMREGCR (Gao et al., 2024), and LST (Zhang et al., 2023b). Extended results available in the Appendix (Table B.3).

et al., 2023) (+1.1 BLEU), and SIMREGCR (Gao et al., 2024) (+0.9 BLEU). This confirms that a strong, pre-trained multilingual text encoder can serve as an effective anchor for speech translation even without explicit domain adaptation of the embedding space.

The full potential of ZEROSWOT is realized in Scenario C. By simply swapping the frozen text branch with one finetuned on MuST-C text data, our LARGE model reaches a state-of-the-art average of 32.9 BLEU. This highlights the modularity of our approach: improvements in the text components can be directly transferred to the speech modality. Most strikingly, our zero-shot model even surpasses the recent LST model (Zhang et al., 2023b) in high-resource directions like English-German (31.2 vs 30.4) and English-Spanish (35.8 vs 35.3). This is significant given that LST relies on the massive LLLAMA-2-13B model (8× more inference parameters than ours) and supervised ST data, whereas ZEROSWOT achieves superior performance using a fraction of the parameters and no paired audio-translation supervision.

Models	ZS / Size	arb	cat	cym	deu	est	fas	ind	jpn	lav	mon	slv	swe	tam	tur	zho	Avg
<i>Medium-sized models</i>																	
XLS-R-1B	✗/ 1.0	19.2	32.1	31.8	26.2	22.4	21.3	30.3	39.9	22.0	14.9	25.4	32.3	18.1	17.1	36.7	26.0
SEAMLESSM4T-v1-M	✗/ 1.2	20.8	37.3	29.9	31.4	23.3	17.2	34.8	37.5	19.5	12.9	29.0	37.3	18.9	19.8	30.0	26.6
ZEROSWOT-M (C)	✓/ 1.0	24.4	38.7	28.8	31.2	26.2	26.0	36.0	46.0	24.8	19.0	31.6	37.8	24.4	18.6	39.0	30.2
<i>Large-sized models</i>																	
XLS-R-2B	✗/ 2.0	20.7	34.2	33.8	28.3	24.1	22.9	32.5	41.5	23.5	16.2	27.6	34.5	19.8	18.6	38.5	27.8
SEAMLESSM4T-v1-L	✗/ 2.3	24.5	41.6	33.6	35.9	28.5	19.3	39.0	39.4	23.8	15.7	35.0	42.5	22.7	23.9	33.1	30.6
SEAMLESSM4T-v2-L	✗/ 2.3	25.4	43.6	35.5	37.0	29.3	19.2	40.2	39.7	24.8	16.4	36.2	43.7	23.4	24.7	35.9	31.7
ZEROSWOT-L (C)	✓/ 1.7	25.7	40.0	29.0	32.8	27.2	26.6	37.1	47.1	25.7	18.9	33.2	39.3	25.3	19.8	40.5	31.2

Table 5.7: BLEU scores on CoVoST 2 eng→xxx. Models are grouped in Medium and Large sizes according to their inference parameters. ZS indicates whether the ST model is zero-shot. Size indicates the inference parameters in Billions. **Scenario C** denotes using Common Voice ASR data and an NLLB model finetuned on CoVoST MT data. With **bold** is the best score for each size category. Supervised baselines from XLS-R (Babu et al., 2022), SEAMLESSM4T-v1 (Seamless Communication, 2023a) and SEAMLESSM4T-v2 (Seamless Communication, 2023b). Extended results available in the Appendix (Table B.5).

CoVoST 2. For these experiments, we train the speech encoder on Common Voice 8.0, which constitutes the in-domain ASR data for this task. Following the definitions in Table 5.4, we evaluate under Scenario C, where the text branch is initialized from an NLLB model finetuned on the CoVoST 2 text-to-translation data. This allows us to assess the method’s scalability and its ability to adapt to a domain where ASR data is available (Common Voice) but paired ST data is treated as unavailable.

Table 5.7 presents the comparison against state-of-the-art supervised baselines, specifically XLS-R (Babu et al., 2022) and SEAMLESSM4T (Barrault et al., 2025). In the MEDIUM size category, ZEROSWOT demonstrates remarkable performance. Despite using no paired ST data, it achieves an average BLEU of 30.2, surpassing the supervised SEAMLESSM4T-v1-MEDIUM (26.6 BLEU) by a substantial margin of +3.6 points. It outperforms the supervised baseline in 12 out of 15 language directions, with particularly large gains in jpn (+8.5) and zho (+9.0). Scaling to the LARGE configuration, our method remains highly competitive. ZEROSWOT-LARGE achieves 31.2 BLEU, outperforming the original SEAMLESSM4T-v1-LARGE (30.6 BLEU) and although is behind the improved v2 version by 0.5 points on average, it sets new state-of-the-art scores in 7 directions. These results reinforce the finding that distinct ASR and MT resources can be effectively combined to build high-performance ST systems without reliance on the scarce end-to-end ST supervision.

5.6.3 Comparison with Cascade Systems

A central question in the evaluation of End-to-End systems is how they compare to the traditional cascade paradigm, which chains an Automatic Speech Recognition (ASR) model with a Machine Translation (MT) model. Since ZEROSWOT effectively functions as a “soft” cascade, as it softly connects a speech encoder to a pre-trained MT decoder via continuous representations, it is crucial to benchmark it against explicit cascade systems constructed from the same components.

Baselines Construction. We construct strong cascade baselines using the exact same components as our ZEROSWOT models:

- **ASR Component:** We use the WAV2VEC 2.0 encoder finetuned on the corresponding ASR data (Common Voice or MuST-C). We explore two architectures:
 1. **CTC-based ASR:** A non-autoregressive model that predicts transcripts using greedy decoding. This mirrors the acoustic encoder of ZEROSWOT.
 2. **AED-based ASR:** An Attention-based Encoder-Decoder model where the WAV2VEC 2.0 encoder is coupled with a Transformer decoder (6 layers). This represents a stronger, autoregressive ASR baseline.
- **MT Component:** We use the NLLB model (MEDIUM or LARGE), either in its original state (Scenarios A/B) or finetuned (Scenario C).

This setup ensures that any performance difference is attributable to the method of integration (discrete vs. continuous) rather than the capacity of the sub-modules. Full model details for the cascades are available in the Appendix (§B.4).

Performance Analysis. Table 5.8 presents the BLEU scores across all three data usage scenarios. We observe that ZEROSWOT consistently outperforms the CTC-based cascades by a significant margin (ranging from +2.7 to +7.2 BLEU). This gap highlights the advantage of passing continuous representations (soft alignment) rather than discrete, error-prone decisions (hard alignment) to the translation model. Notably, in Scenario C, the performance of the CTC cascade drops significantly (e.g., from 27.5→26.5 for the Large model). We hypothesize that the domain-adapted NLLB, having been finetuned on clean ground-truth text, becomes less robust to the noisy transcripts generated by the CTC

Model Size	System Architecture	BLEU(↑) per Scenario			Inference Time(↓)
		A (Out-Domain)	B (In-Domain)	C (Adapted)	
MEDIUM	Cascade (CTC-ASR → NLLB)	21.8	25.4	24.7	0.41
	Cascade (AED-ASR → NLLB)	25.2	30.0	32.2	0.56
	ZEROSWOT	26.0	29.6	31.9	0.46
LARGE	Cascade (CTC-ASR → NLLB)	24.7	27.5	26.5	0.66
	Cascade (AED-ASR → NLLB)	26.5	31.1	33.2	0.82
	ZEROSWOT	27.4	31.4	32.9	0.78

Table 5.8: Comparison of ZEROSWOT against Cascade ST baselines grouped by model size and data usage scenarios (see table 5.4). Average BLEU scores on on MuST-C tst-COMMON eng→8. Inference time measured in seconds (average time per example). In **bold** is the best in each category. CTC-ASR: Non-autoregressive ASR. AED-ASR: Autoregressive ASR with Encoder-Decoder. Extended results available in the Appendix (Table B.3).

model. In contrast, ZEROSWOT is robust to this issue as it learns to align directly with the MT embedding space.

When compared to the stronger AED-based cascades, ZEROSWOT remains highly competitive. In Scenario A (Out-of-Domain), our method actually outperforms the AED cascade (25.2→26.0 BLEU for Medium, 26.5→27.4 for Large), suggesting that our cross-modal alignment generalizes better than the two-stage pipeline when in-domain ASR data is unavailable. In Scenarios B and C, the AED cascade generally achieves slightly higher scores, with the exception of the Large model in Scenario B where ZEROSWOT leads. This is likely due to the ASR decoder’s internal language model correcting phonetic errors before translation.

Efficiency. While AED-based cascades offer high quality, they are computationally expensive due to the two sequential autoregressive generation steps (ASR decoding + MT decoding). ZEROSWOT bypasses the text generation step entirely. We measured the average inference time per sentence and found ZEROSWOT to be approximately 18% faster (0.56→0.46) than the AED-based cascade for the MEDIUM configuration and 5% faster (0.82→0.78) for the LARGE configuration. Thus, our method offers a better trade-off, delivering quality comparable to strong autoregressive cascades while being more efficient.

Model Size	Zero-Shot Training	Zero-Shot	Finetuned	Δ
MEDIUM	B (ASR only)	28.5	30.8	+2.3
	C (ASR + MT)	30.5	30.7	+0.2
LARGE	B (ASR only)	30.1	32.0	+1.9
	C (ASR + MT)	31.2	31.4	+0.2

Table 5.9: BLEU scores on MuST-C eng-deu tst-COMMON before and after supervised ST fine-tuning.

5.6.4 Supervised ST Fine-tuning

A key question for any zero-shot method is whether it can serve as an effective initialization for supervised learning. To investigate this, we take our zero-shot models trained on MuST-C (from Table 5.6) and finetune them on the English-German direction of the MuST-C dataset using paired ST data.

We initialize our ST finetuning with either the MEDIUM or LARGE variants of ZEROSWOT. Additionally we consider initializations from model trained with two data usage scenarios (Table 5.4): Scenario B, where we trained with MuST-C ASR and Scenario C, where we additionally utilized an NLLB adapted with the MuST-C MT data. More information on the details of the supervised ST finetuning are available in the Appendix (§B.2.4).

The results in Table 5.9 reveal a distinct contrast between the two scenarios. Models initialized from (B) prove to be highly adaptable, yielding substantial improvements of +2.3 and +1.9 BLEU for the MEDIUM and LARGE configurations. Notably, the finetuned ZEROSWOT-LARGE (B) reaches 32.0 BLEU, setting a new state-of-the-art for this benchmark and surpassing the previous best published result by 1.6 points (Zhang et al., 2023b).

On the other hand, models initialized from (C) show only marginal gains (+0.2 BLEU). This suggests that when the model is trained to strictly align with a specific, finetuned MT space (as in Scenario C), it may settle into a local minimum that is difficult to escape during subsequent ST fine-tuning. In contrast, the alignment learned solely from ASR data (Scenario B) appears more flexible, providing a better starting point for supervised adaptation.

5.7 Analysis: The Modality Gap

In this section, we analyze the inner workings of ZEROSWOT to verify if our design choices effectively bridge the gaps between the speech and text modalities. We focus on two key aspects: the reduction of the sequence length mismatch via our compression adapter, and the semantic alignment of the learned representations.

5.7.1 Addressing the Length Gap

As discussed in §5.1, a major challenge in zero-shot ST is the discrepancy in sequence length between acoustic frames and text subwords. To evaluate the effectiveness of our Compression Adapter (§5.4.1), we compare it against alternative compression strategies.

We replace our proposed compression mechanism with three baselines, with their implementation details being available in the Appendix (§B.2.3).

1. **No Compression:** The raw acoustic features from the WAV2VEC 2.0 encoder are passed directly to the shared encoder.
2. **Length Adaptor:** We use a strided convolutional adaptor that down-samples the sequence by a fixed factor of 4, following Li et al. (2021); Gállego et al. (2021).
3. **Character-level Compression:** We employ standard CTC compression that merges repeated tokens but does not group them into subwords (Gaido et al., 2021a).

We also analyze the impact of the tokenization used to construct the CTC targets. We compare our proposed strategy (“Subword + Unk”), which includes the <unk> token for characters outside the vocabulary, against a naive “Subword” tokenization that might ignore them.

We evaluate these configurations on the MuST-C `tst-COMMON` set using the MEDIUM model in Scenario B. Apart from the translation quality (BLEU), we measure the alignment in terms of sequence length. We report the average absolute length difference $|\Delta \text{len}| = |M - N|$ and the length ratio $r(\text{len}) = M/N$, where M and N are the sequence lengths for the speech and text representations $\mathbf{H}^s, \mathbf{H}^x$. An ideal alignment should yield an absolute difference close to 0 and a ratio close to 1.

Compression	Tokenization	BLEU ($\uparrow$)	$ \Delta\text{len} $ ($\downarrow$)	$r(\text{len})$ ($\rightarrow 1$)
None	Word	28.9	267.6	10.9
Length Adaptor	Word	28.9	49.2	2.82
Character-level	Word	29.0	71.2	3.61
Compr. Adapter	Subword	28.4	3.3	0.88
	Subword + Unk (Ours)	29.6	1.4	0.98

Table 5.10: Analysis of sequence length alignment on MuST-C *tst*-COMMON. We report BLEU scores, average absolute length difference ($|\Delta\text{len}|$), and length ratio ($r(\text{len})$) between the compressed speech representations and the reference text embeddings. Our proposed method (Subword + Unk) achieves the best trade-off, effectively closing the length gap.

The results are presented in Table 5.10. We observe that standard methods (Rows 1-3) fail to align the sequence lengths, resulting in speech representations that are $2.8\times$ to $10.9\times$ longer than the text. While the *Length Adaptor* and *Character-level* compression achieve respectable BLEU scores (28.9 and 29.0), they do not solve the structural mismatch.

In contrast, our proposed Compression Adapter significantly reduces the length gap. When using the optimal “Subword + Unk” tokenization, the length ratio is 0.98, indicating that the speech sequence length is almost identical to the text sequence length ($|\Delta\text{len}| \approx 1.4$). This precise alignment contributes to the superior performance of 29.6 BLEU.

Interestingly, the “Subword” only tokenization (Row 4) results in a length ratio of 0.88, implying over-compression. This loss of information is likely due to dropping non-vocabulary characters and negatively impacts translation quality (28.4 BLEU), performing worse than even the uncompressed baseline. This finding highlights that while matching the length is important, preserving the information content via proper tokenization is crucial.

5.7.2 Closing the Representation Gap

Beyond length alignment, the core premise of ZEROSWOT is that we can map acoustic features into the semantic space of the MT model. To verify if we have successfully bridged this representation gap, we conduct a cross-modal retrieval experiment.

Model	Retrieval Accuracy (%)
CONST (Ye et al., 2022)	95.0
ZEROSWOT (Cosine)	98.1
ZEROSWOT (Wasserstein)	98.5

Table 5.11: Speech-to-Text retrieval accuracy on MuST-C eng-deu tst-COMMON. We compare the retrieval performance of ZEROSWOT against CONST, using both cosine similarity (on mean-pooled representations) and the negative Wasserstein distance (on non-pooled representations).

We use the 2,551 examples from the MuST-C English-German tst-COMMON set. For every speech input s , we attempt to retrieve its corresponding text transcription x from the pool of all test sentences. We compute the similarity between the computed speech representation H^s and the reference text representation H^x (extracted from the frozen NLLB encoder).

We evaluate retrieval accuracy using two metrics:

- **Cosine Similarity:** We apply mean-pooling over the time dimension for both sequences and compute the cosine similarity between the resulting vectors.
- **Wasserstein Distance:** We compute the negative Wasserstein distance directly between the sequences H^s and H^x , as defined in Equation (5.16).

We compare our method against CONST (Ye et al., 2022), a state-of-the-art model that explicitly uses contrastive learning to align modalities, which implies very accurate retrieval with cosine similarity.

The results in Table 5.11 demonstrate the effectiveness of our alignment strategy. ZEROSWOT achieves a retrieval accuracy of 98.5% when using the Wasserstein metric, outperforming CONST by 3.5 percentage points. Even when using simple cosine similarity on pooled vectors, our model maintains high accuracy (98.1%).

This strong retrieval score confirms that the speech representations generated by our encoder are not only structurally similar to text (in terms of length) but also semantically very close in the shared embedding space. This strong alignment explains why the frozen MT decoder can successfully translate speech inputs without ever having been trained on them.

5.8 Conclusions and Next Steps

Conclusions. In this chapter, we presented ZEROSWOT, a framework for zero-shot End-to-End Speech Translation that effectively bridges the modality gap between speech and text without relying on any paired ST data. By leveraging Optimal Transport and a novel Subword Compression mechanism, we demonstrated that a speech encoder can be trained to “mimic” the embedding space of a massively multilingual Machine Translation model using only ASR data.

Our experimental results on MuST-C and CoVoST 2 confirm the effectiveness of this approach. ZEROSWOT not only significantly outperforms previous zero-shot methods but also surpasses strong supervised baselines and cascaded systems in several high-resource settings. This performance is underpinned by our analysis, which shows that our method successfully aligns speech and text representations in terms of both sequence length and semantic content. The ability to seamlessly plug a learned speech encoder into a frozen, pre-trained text decoder opens new avenues for scaling ST to the long tail of languages where paired data is scarce but ASR and text-only resources are more readily available.

Limitations and Future Directions. Despite these advancements, our approach mimics the text embedding space, which inherently limits its ability to capture paralinguistic features. Since the speech representations are forced to align with text embeddings, acoustic nuances such as prosody and speaker identity, which are not present in the text modality are largely discarded. This suggests that while we solve the data scarcity problem, we may be sacrificing the richness of the spoken signal, a trade-off we investigate further in Part IV of the thesis.

Furthermore, the Subword Compression mechanism, while effective, relies on the specific tokenization of the target MT model. This rigid dependency on a fixed subword vocabulary can be suboptimal for cross-lingual transfer, as subwords are often language-specific. As we will discuss in Ch. 6, moving towards a more granular, character-level alignment can offer greater flexibility and improved generalization across diverse languages.

6 Character-level Modeling for Enhanced Modality Transfer

This chapter is based on our work “Improving Language and Modality Transfer in Translation by Character-level Modeling”, published in **ACL 2025** (Tsiamas et al., 2025a) and developed during a research internship at Meta FAIR between August 2024 – February 2025.

6.1 Introduction

In Ch. 5, we addressed the scarcity of paired Speech Translation (ST) data by introducing ZEROSWOT. That method demonstrated that it is possible to bridge the modality gap between speech and text without using any parallel speech-to-translation corpora. By leveraging Optimal Transport, we aligned the continuous output of a speech encoder with the discrete subword embedding space of a pretrained Machine Translation (MT) model. This allowed us to reuse the strong decoding capabilities of multilingual text models for speech inputs.

The Subword Bottleneck. ZEROSWOT and similar approaches rely heavily on the specific subword vocabulary of the underlying text model. Subword tokenization, while standard in modern Natural Language Processing (NLP), introduces a structural rigidity that becomes problematic in zero-shot scenarios. The tokenizer is typically trained on a fixed set of languages. When applied to unseen or low-resource languages, the subword segmentation often becomes semantically opaque or inefficient, fragmenting words into unrecognizable sequences. This dependency limits the ability of the model to generalize to the “long tail” of languages that were not present during the initial training of the text model.

A second limitation identified in the previous chapter is the granularity mismatch between modalities. Speech encoders trained with Connectionist Temporal Classification

(CTC) (Graves et al., 2006), such as WAV2VEC 2.0 (Baevski et al., 2020) or the Massively Multilingual Speech (MMS) model (Pratap et al., 2023), naturally output characters. In contrast, text models operate on subwords (Liu et al., 2020a; NLLB, 2024). In Ch. 5, we proposed to solve this length mismatch with a specialized adaptor module which compresses the acoustic representations into subword-level sequences. However, our study focused on English-based cross-modal adaptation, leading to an English-to-Many zero-shot ST model. While effective for high-resource source languages, applying this subword compression methodology to the 1,000+ languages covered by MMS is impractical, as the majority of them lack the resources to learn an accurate subword compression adapter.

Character-level Alignment. In this chapter, we propose shifting the interface of cross-modal alignment to the atomic level of written language: characters. We introduce CHARSONAR, a method that adapts the massive multilingual embedding space of SONAR (Duquenne et al., 2023) (see §2.5.3) to operate on character sequences. By moving to a character-level representation, we remove the dependency on language-specific tokenizers. This simplification provides a universal interface for both text and speech, as characters are shared across languages more broadly than subwords. This shared interface significantly improves cross-lingual transfer to low-resource languages and enables efficient cross-modal transfer from pretrained ASR models.

Our approach consists of two stages. First, we adapt the text encoder of SONAR to build CHARSONAR. We use a teacher-student framework where a character-based student learns to mimic the fixed-size sentence embeddings of the subword-based teacher. We find that predicting an interpolated embedding between the source and target sentences improves the quality of the student’s representation. To ensure the text encoder is robust to the errors typical of CTC-based speech recognition systems, we inject noise into the character sequences during training.

Second, we address the speech modality. Since both our adapted text encoder and pretrained CTC-based ASR models operate at the character level, the alignment process is simplified. We demonstrate that a lightweight adapter can effectively map the compressed acoustic features from the MMS encoder directly to the CHARSONAR embedding space. This compatibility is so strong that we can achieve competitive zero-shot ST performance using a projection derived solely from the pretrained weights of the speech and text modules, without any training.

We evaluate this approach on many-to-English text and speech translation on the massively multilingual FLORES+ (NLLB, 2024) and FLEURS (Conneau et al., 2023) benchmarks. The results show that character-level modeling outperforms subword baselines in text translation for low-resource languages and exhibits superior zero-shot generalization to languages unseen during training. In speech translation, CHARSONAR sets a new state-of-the-art for zero-shot performance, outperforming supervised models like WHISPER (Radford et al., 2023) and SEAMLESSM4T (Barrault et al., 2025) without any supervision from paired ST data.

The main contributions of this chapter are as follows:

- We introduce CHARSONAR, a character-based text encoder that aligns with the fixed-size embedding space of SONAR. We conduct a large-scale study on 75 languages, and demonstrate that character-level processing improves cross-lingual transfer, outperforming subword baselines like SONAR and NLLB on low-resource languages and showing superior zero-shot generalization to unseen languages.
- We propose a robust training objective for the text encoder that utilizes interpolated embeddings (between source and target) and ASR-like noise augmentation. This ensures the text encoder is resilient to phonetic errors and semantically aligned with the translation task.
- We develop a zero-shot speech translation framework by connecting the CHARSONAR text encoder with the MMS speech encoder. We show that the shared character interface allows for effective cross-modal alignment using simple, lightweight adapters, eliminating the need for complex subword compression modules.
- Our model achieves state-of-the-art zero-shot ST results on the FLEURS benchmark. It outperforms strong supervised baselines on 33 languages of various resourcefulness, demonstrating that it is possible to build competitive ST systems for over 1,000 languages using only ASR data and text-only translation resources.

6.2 Related Work

6.2.1 Character-level Machine Translation

The dominant paradigm in Neural Machine Translation (NMT) relies on subword segmentation algorithms, such as Byte-Pair Encoding (BPE) (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018). These methods balance the trade-off between vocabulary size and sequence length. However, operating directly at the character level has long been an alternative research direction. Early approaches utilized character-level inputs with attention-based sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015) to resolve out-of-vocabulary issues and improve the modeling of morphologically rich languages (Costa-jussà and Fonollosa, 2016; Chung et al., 2016; Lee et al., 2017; Cherry et al., 2018).

More recently, the introduction of “token-free” models like BYT5 (Xue et al., 2022) and CANINE (Clark et al., 2022) revived interest in this granularity. These architectures process raw text bytes or characters, eliminating the need for a dedicated tokenizer and making the models more robust to noise and spelling variations, as well as beneficial in low-resource machine translation (Edman et al., 2024). Despite these advantages, character-level models have not seen widespread adoption. The primary obstacle is computational efficiency (Libovický et al., 2022). Character sequences are significantly longer than their subword counterparts, often by a factor of three or four. In standard Transformer architectures, where the self-attention mechanism scales quadratically with sequence length, this increase results in prohibitive training and inference costs (Vaswani et al., 2017). Consequently, most state-of-the-art multilingual MT models, such as NLLB (NLLB, 2024), continue to rely on subword tokenization.

6.2.2 Cross-modal Transfer

Speech translation models frequently rely on cross-modal knowledge transfer from the data-rich task of Machine Translation. Standard approaches employ multi-tasking frameworks where text modules are shared between tasks, allowing the semantic text encoder to process either acoustic representations or text embeddings (Liu et al., 2020b; Zhou et al., 2023; Tang et al., 2021b; Fang et al., 2022). Other methods explicitly minimize the distance

between speech and text representations in a shared latent space (Tang et al., 2021a; Ye et al., 2022; Ouyang et al., 2023). We provided a comprehensive review of these representation alignment strategies and their limitations regarding sequence length in §5.2.2.

In Ch. 5, we introduced ZEROSWOT, which eliminated the dependency on parallel ST data by minimizing the Wasserstein distance (Frogner et al., 2015) between speech and text sequences. In this chapter, we stay close to the same zero-shot paradigm but leverage the fixed-size bottleneck of SONAR. This architectural choice simplifies the optimization problem, allowing us to minimize a Mean Squared Error (MSE) distance rather than the computationally intensive Optimal Transport metric used in the previous chapter.

A critical challenge in maximizing transfer is unifying the tokenization of the acoustic output and the text input. Acoustic encoders trained with CTC (Graves et al., 2006) typically output characters or phonemes, whereas text encoders operate on subwords. Previous works have attempted to bridge this gap by using phonemized text as input (Tang et al., 2021a; Le et al., 2023). However, phoneme-based inputs degrade performance due to semantic ambiguity, as different words can share identical phonetic realizations. Alternatively, some approaches force the acoustic encoder to output subwords directly (Liu et al., 2020b; Yang et al., 2023). This strategy is difficult to scale to massively multilingual settings where the combined vocabulary can reach hundreds of thousands of tokens (NLLB, 2024).

The ZEROSWOT method addressed this mismatch with a character-to-subword compression adapter (§5.4.1). While effective, that module requires a substantial amount of ASR data to learn the compression logic, which restricts its applicability to the long tail of low-resource languages. Our approach circumvents these issues by modifying the text encoder to accept character-level inputs. This modification aligns the text modality with the natural output of the acoustic encoder, enabling the use of data-efficient and lightweight adapters without the need for complex compression mechanisms.

6.2.3 SONAR and Fixed-size Embeddings

The architecture of the underlying text model is a determining factor for the efficiency of character-level translation. We build upon SONAR (Duquenne et al., 2023), a massively multilingual model designed for semantic similarity and mining tasks. SONAR follows

the design philosophy of LASER (Artetxe and Schwenk, 2019), employing a bottleneck architecture. It is initialized from the NLLB encoder-decoder and thus supports 200+ languages. More information is provided in §2.5.3.

In this framework, the encoder maps a variable-length input sequence to a single, fixed-size vector representation (sentence embedding). The decoder then generates the translation conditioned solely on this fixed vector. This decoupling of the encoder and decoder is critical for our method. By modifying only the encoder to process characters, we pay the computational cost of longer sequences only during the encoding step. The decoder, which typically dominates the computational budget in autoregressive generation, receives the same fixed-size input regardless of whether the source was tokenized into subwords or characters. This structural characteristic allows CHARSONAR to leverage the benefits of character-level modeling without the inference latency penalties associated with models like BYT5.

6.3 Methodology

Our proposed method, CHARSONAR, is built upon the premise that a shared, character-level embedding space can serve as a universal interface for both inter-lingual (text-to-text) and cross-modal (speech-to-text) transfer. We first recap the SONAR architecture (§6.3.1) and then detail our proposed methodology consisting of two distinct adaptation stages. In the first stage we adapt a pretrained subword-based text encoder to operate on characters (§6.3.3). In the second stage, we connect MMS, which is a massively multilingual CTC-based speech encoder, to this newly established character-level input space using a light-weight adapter (§6.3.4). Our methodology is depicted in Figure 6.1.

6.3.1 The Foundation: SONAR

The backbone of our approach is SONAR (Duquenne et al., 2023), a massively multilingual model designed for semantic sentence embedding and mining tasks. Unlike standard encoder-decoder architectures used in NMT (e.g., NLLB (NLLB, 2024)), which pass a sequence of hidden states from encoder to decoder, SONAR employs a bottleneck architecture similar to LASER (Artetxe and Schwenk, 2019).

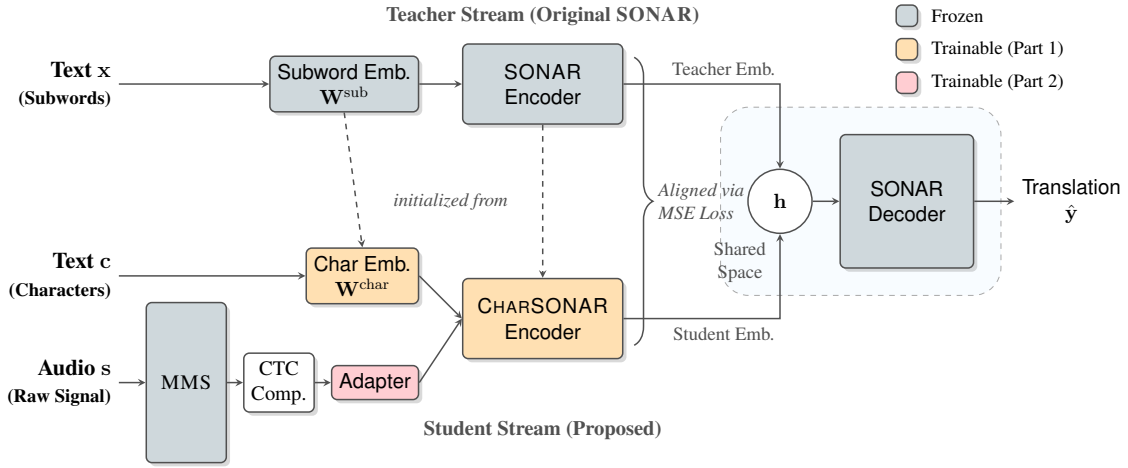


Figure 6.1: Schematic of the CHARSONAR methodology using a student-teacher distillation framework. The architecture is divided into a frozen Teacher Stream (top) and a trainable Student Stream (bottom). **Part 1 (Text Adaptation):** We construct a character-level encoder initialized from the SONAR backbone and train it (orange components) to align character sequences $\mathbf{c}$ with the teacher’s subword-based embeddings via MSE loss. **Part 2 (Speech Adaptation):** Keeping the character encoder frozen, we train a lightweight adapter (pink component) to map compressed MMS acoustic features $\mathbf{s}$ into the character input space. Both modalities are mapped to the shared semantic space $\mathbf{h}$, enabling zero-shot translation via the pretrained decoder.

The encoder is Transformer-based (Vaswani et al., 2017) with model dimension d_{sonar} . It processes a source sequence of subwords $\mathbf{x} = (x_1, \dots, x_N)$ from a standard subword-based vocabulary $\mathcal{V}_{\text{sub}}$ to a sequence of continuous embedding vectors $\mathbf{E}^x$ with $W^{\text{sub}} : |\mathcal{V}_{\text{sub}}| \rightarrow d_{\text{sonar}}$, and then a stack of layers produces a semantic representation $\mathbf{H}^x \in \mathbb{R}^{N \times d_{\text{sonar}}}$. The crucial feature of this architecture is the pooling mechanism: the sequential semantic representation is aggregated via mean-pooling into a single, fixed-size vector $\mathbf{h}^x \in \mathbb{R}^{d_{\text{sonar}}}$. The sentence embedding $\mathbf{h}^x$ contains the semantic content of the input, independent of its sequence length or source language.

$$\mathbf{E}^x = \text{EMBEDDING}_{\text{sub}}(\mathbf{x}) \quad (6.1)$$

$$\mathbf{H}^x = \text{ENCODER}_{\text{sub}}(\mathbf{E}^x) \quad (6.2)$$

$$\mathbf{h}^x = \text{mean-pooling}(\mathbf{H}^x) \quad (6.3)$$

The decoder, which also has a dimensionality d_{sonar} , conditions its generation solely on the sentence embedding $\mathbf{h}^x$ to produce the target sequence $\hat{\mathbf{y}}$. This decoupling of the encoder and decoder is advantageous for our purposes. It implies that we can modify the encoder’s input granularity (e.g., from subwords to characters) without necessitating any changes to

the decoder or incurring the computational penalty of cross-attending to longer sequences during generation.

6.3.2 Problem Formulation and Preliminaries

Before detailing our adaptation strategy, we formalize the problem of zero-shot modality transfer and define the notation used throughout this chapter.

Notation. We consider three distinct input modalities/granularities:

1. **Subword Text:** A sequence $\mathbf{x} = (x_1, \dots, x_N)$ where each token $x_i \in \mathcal{V}_{\text{sub}}$ is a subword from the original SONAR vocabulary.
2. **Character Text:** A sequence $\mathbf{c} = (c_1, \dots, c_M)$ where each token $c_j \in \mathcal{V}_{\text{char}}$ is a character. Note that typically $M > N$ for the same sentence.
3. **Speech Signal:** A raw audio waveform $\mathbf{s}$, which is processed into a sequence of acoustic frames $\mathbf{A} \in \mathbb{R}^{T \times d_{\text{mms}}}$ by a powerful multilingual CTC model (MMS).

Let $\mathcal{H} \subset \mathbb{R}^{d_{\text{sonar}}}$ denote the shared, fixed-size multilingual semantic space established by SONAR. We denote the final pooled sentence embedding as $\mathbf{h} \in \mathcal{H}$.

The Interface-Backbone Decomposition. Standard encoder architectures are often treated as monolithic functions $f(\text{input}) \rightarrow \text{output}$. However, to enable modality transfer, we explicitly decompose the encoder Φ into two functional components: an *Input Interface* $\mathcal{E}$ and a *Semantic Backbone* Φ_{body} .

$$\mathbf{z} = \Phi(\text{input}) = \text{pool} \left(\Phi_{\text{body}}(\mathcal{E}(\text{input})) \right) \quad (6.4)$$

The *Semantic Backbone* Φ_{body} consists of the deep stack of Transformer layers that reason about semantics and syntax. The *Input Interface* $\mathcal{E}$ is responsible for projecting discrete tokens or continuous signals into the hidden dimension d_{sonar} expected by the backbone.

Problem Statement. The original SONAR model provides a pretrained interface $\mathcal{E}_{\text{sub}} : \mathcal{V}_{\text{sub}} \rightarrow \mathbb{R}^{N \times d_{\text{sonar}}}$ and a powerful backbone Φ_{body} . However, this interface is coupled to

a rigid subword vocabulary $\mathcal{V}_{\text{sub}}$ that generalizes poorly to unseen languages and cannot ingest continuous speech signals.

Our objective is to learn two new interfaces:

1. A character interface $\mathcal{E}_{\text{char}} : \mathcal{V}_{\text{char}}^* \rightarrow \mathbb{R}^{M \times d_{\text{sonar}}}$ that allows the model to process text at the character level.
2. A speech interface (adapter) $\mathcal{E}_{\text{speech}} : \mathbb{R}^{T \times d_{\text{mms}}} \rightarrow \mathbb{R}^{T' \times d}$, with $T' \approx M$, that allows the model to process acoustic frames.

Crucially, we aim to learn these interfaces such that they are *interchangeable*. For a semantic concept represented by subword sequence $\mathbf{x}$, character sequence $\mathbf{c}$, and speech signal $\mathbf{s}$, we require:

$$\Phi_{\text{body}}(\mathcal{E}_{\text{sub}}(\mathbf{x})) \approx \Phi_{\text{body}}(\mathcal{E}_{\text{char}}(\mathbf{c})) \approx \Phi_{\text{body}}(\mathcal{E}_{\text{speech}}(\mathbf{s})) \quad (6.5)$$

We ensure that all modalities are mapped to the exact same semantic coordinates $\mathbf{h}$ used by the pretrained decoder, enabling zero-shot translation.

6.3.3 Text Adaptation: Constructing CHARSONAR

To create a character-level encoder, we employ a knowledge distillation (teacher-student) framework. Our goal is to train a student encoder, CHARSONAR, which processes character sequences, to mimic the semantic embedding space of the frozen SONAR teacher encoder.

Architecture and Initialization. The CHARSONAR student inherits the architecture of the SONAR encoder having a dimensionality d_{sonar} . We initialize the student’s weights directly from the teacher to preserve the pretrained multilingual knowledge. This encoder processes a sequence of characters $\mathbf{c} = (c_1, \dots, c_M)$, and thus we must adapt the embedding layer to the new granularity. We construct a character vocabulary $\mathcal{V}_{\text{char}} \subset \mathcal{V}_{\text{sub}}$ by retaining only those tokens from the original subword-based vocabulary $\mathcal{V}_{\text{sub}}$ that correspond to single characters. The embeddings vectors for these characters are copied from the teacher embedding $\mathbf{W}^{\text{sub}}$ to the new student embedding $\mathbf{W}^{\text{char}} : |\mathcal{V}_{\text{char}}| \rightarrow d_{\text{sonar}}$. We

obtain the character-level sentence embedding $\mathbf{h}^c$ as:

$$\mathbf{E}^c = \text{EMBEDDING}_{\text{char}}(\mathbf{c}) \quad (6.6)$$

$$\mathbf{H}^c = \text{ENCODER}_{\text{char}}(\mathbf{E}^c) \quad (6.7)$$

$$\mathbf{h}^c = \text{mean-pooling}(\mathbf{H}^c) \quad (6.8)$$

Training Objectives. We train the student encoder to project the character-level embedding $\mathbf{h}^c$ of a character-tokenized source sentence $\mathbf{c}$ to a target teacher embedding. We investigate three distinct Mean Squared Error (MSE) objectives to determine the optimal signal for cross-lingual alignment:

- **Reconstruction:** The student learns to reproduce the teacher’s embedding of the same sentence. Given the sentence embeddings $\mathbf{h}^c$ and $\mathbf{h}^x$ of the same sentence:

$$\mathcal{L}_{\text{recon}} = \text{MSE}(\mathbf{h}^c, \text{sg}[\mathbf{h}^x]) \quad (6.9)$$

where $\text{sg}[\cdot]$ indicates the stop-gradient operator (i.e., the teacher encoder is frozen).

This approach relies on monolingual data and ensures the student retains the teacher’s representation ability.

- **Translation:** The student learns to predict the teacher’s embedding of a translation $\mathbf{y}$. Given the student character-level embedding $\mathbf{h}^c$ of the source sentence, and the teacher embedding of the target sentence $\mathbf{h}^y$:

$$\mathcal{L}_{\text{trans}} = \text{MSE}(\mathbf{h}^c, \text{sg}[\mathbf{h}^y]) \quad (6.10)$$

This objective explicitly forces cross-lingual alignment, encouraging the student to map the source text directly to the semantic space of the target language. Note that although SONAR is a cross-lingual space, due to data scarcity and noise, low-resource languages tend to under-perform in downstream tasks (Duquenne et al., 2023). Thus, the translation objective can potentially close this gap by re-arranging the embedding space to benefit the under-represented languages. On the other hand, compared to the reconstruction objective, it has to rely on relatively scarcer and more noisy MT data. Another benefit of this objective is that it can allow us to add new languages to the embedding space, assuming access to MT data with a target language already included in SONAR.

- **Interpolation:** We propose a novel objective that encourages language agnosticity by mapping the input to a language-neutral “centroid”. We hypothesize that the optimal semantic representation lies between the source and target embeddings in the multilingual space. Thus, we minimize the distance to the average of the teacher’s source and target embeddings:

$$\mathcal{L}_{\text{interpol}} = \text{MSE} \left(\mathbf{h}^c, \frac{\text{sg}[\mathbf{h}^x] + \text{sg}[\mathbf{h}^y]}{2} \right) \quad (6.11)$$

Empirically, we find that this objective produces higher quality embeddings for downstream translation tasks (see §6.5.1).

Robustness via Augmentation. A primary motivation for using characters is to bridge the gap with ASR CTC-based outputs, which are often noisy and lack standard orthography. This is due to two reasons. First, they are trained on normalized text (uncased and stripped of punctuation). Secondly, they are not auto-regressively generated and thus they are prone to errors. To prepare the CHARSONAR encoder for this imperfect input, we apply two different augmentations to the character-level sequence $\mathbf{c}$ seen by the student encoder during the text adaptation phase.

First, we apply *text normalization* with probability p_{norm} . This involves lowercasing the text and removing punctuation, simulating the raw output of a CTC model. Second, we apply *noise injection* with probability p_{noise} . We randomly insert, delete, or replace characters within the sequence. This forces the encoder to learn robust representations that are invariant to local spelling errors, effectively simulating the phonetic ambiguities encountered in speech transcription.

6.3.4 Speech Adaptation: Bridging MMS and CHARSONAR

Having established a character-level text embedding space (CHARSONAR), we now turn to the second stage of our methodology: aligning speech representations to this space. Our objective is to map the acoustic features from a pretrained CTC encoder to the input space of CHARSONAR, thereby enabling zero-shot speech-to-text translation.

Acoustic Encoder. We utilize the Massively Multilingual Speech (MMS) model (Pratap et al., 2023) as our acoustic backbone. MMS is a WAV2VEC 2.0 (Baevski et al., 2020) model finetuned with the Connectionist Temporal Classification (CTC) loss (Graves et al., 2006) on labeled speech data from over 1,000 languages. The encoder processes a raw audio waveform s to produce a sequence of contextualized acoustic representations $\mathbf{A} \in \mathbb{R}^{T \times d_{\text{mms}}}$, where T is the length of the sequence and d_{mms} is the model dimension of MMS.

$$\mathbf{A} = \text{ACOUSTIC-ENCODER}(s) \quad (6.12)$$

A CTC head predicts a probability distribution over the character vocabulary for each time step. MMS utilizes language-specific CTC heads, so each language also has its own small character-level vocabulary. For a language ℓ :

$$\mathbf{P} = \text{softmax}(\mathbf{A}\mathbf{W}_{(\ell)}^{\text{ctc}}) \quad (6.13)$$

Where $\mathbf{P} \in \mathbb{R}^{T \times |\mathcal{B}^{(\ell)}|}$ is the CTC probability distribution over a language-specific MMS character vocabulary $\mathcal{B}^{(\ell)}$, and $\mathbf{W}_{(\ell)}^{\text{ctc}}$ is the language-specific CTC linear head parameters.

Aligning Granularity via CTC Compression. As discussed in §6.1, a key advantage of our approach is the matching granularity between the ASR output and our CHARSONAR input. Both operate on characters. However, the sequence length T of the acoustic representation $\mathbf{A}$ is significantly longer than the corresponding character sequence due to the high sampling rate of audio (typically 50 frames per second). To resolve this, we employ CTC compression (Gaido et al., 2021a), similar to the first stage of compression used in ZEROSWOT (§5.4.1). We use the CTC greedy predictions to collapse the acoustic sequence $\mathbf{A}$ into a compressed sequence $\mathbf{A}' \in \mathbb{R}^{T' \times d_{\text{mms}}}$ (with $T' \approx M$), by averaging consecutive acoustic frames that correspond to the same non-blank character prediction and discard frames classified as the CTC blank token.

$$\mathbf{A}' = \text{compress}(\mathbf{A}, \mathbf{P}) \quad (6.14)$$

Adapter Architecture. We introduce an adapter module to project the compressed acoustic features $\mathbf{A}'$ into the CHARSONAR input space, effectively replacing its embedding layer, to produce pseudo-character embeddings $\mathbf{E}^s \in \mathbb{R}^{T' \times d_{\text{sonar}}}$. We define the speech

embedding $\mathbf{h}^s \in \mathbb{R}^{d_{\text{sonar}}}$ from CHARSONAR as:

$$\mathbf{E}^s = \text{ADAPTER}(\mathbf{A}')$$

$$\mathbf{H}^s = \text{ENCODER}_{\text{char}}(\mathbf{E}^s)$$

$$\mathbf{h}^s = \text{mean-pooling}(\mathbf{H}^s)$$

We explore two adapter configurations, which are depicted in Figure 6.2.

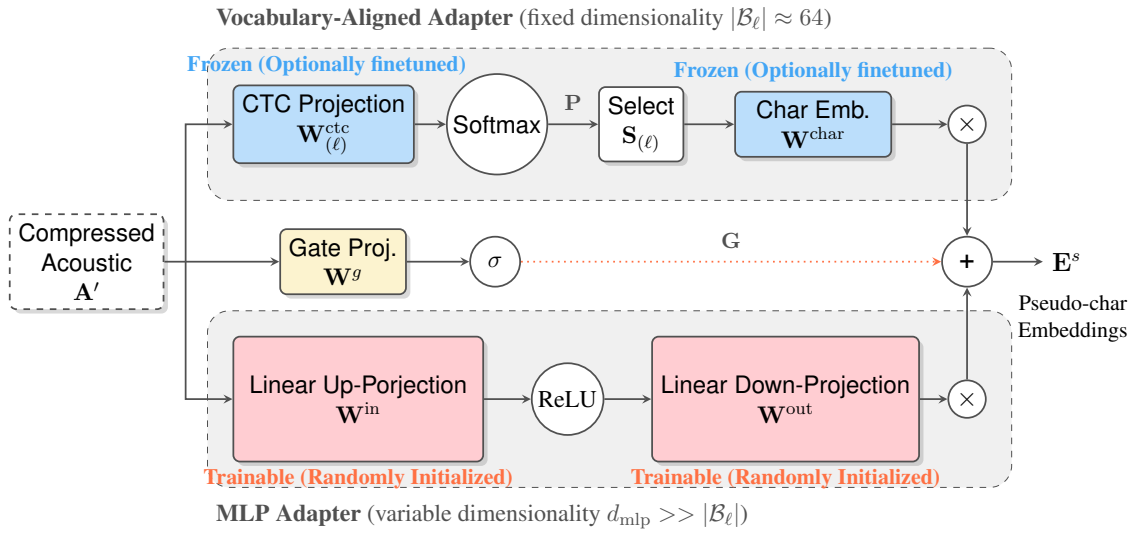


Figure 6.2: The Vocabulary-Aligned and Gated Dual Adapter architectures. The adapter maps compressed acoustic features $\mathbf{A}'$ to the embedding space $\mathbf{E}^s$ via two parallel branches. The Vocabulary-Aligned Adapter path (top, blue) leverages the CTC head and character embeddings to retain explicit phonetic alignment. The residual path (bottom, pink) employs a trainable MLP to capture paralinguistic information missed by the rigid character projection and overall increase the modeling capacity of the Adapter. A dynamic gating mechanism (middle, yellow) computes a scalar G to regulate the trade-off between the two information branches.

- **Vocabulary-Aligned Adapter:** All language-specific MMS vocabularies are subsets of our CHARSONAR vocabulary, i.e., $\mathcal{B}^{(\ell)} \subset \mathcal{V}_{\text{char}}, \forall \ell$. While $\mathbf{W}_{(\ell)}^{\text{ctc}}$ projects to the local ASR vocabulary $\mathcal{B}^{(\ell)}$, and $\mathbf{W}^{\text{char}}$ accepts the global vocabulary $\mathcal{V}_{\text{char}}$, we can link them via the token identities.

We define a one-hot encoding selection matrix $\mathbf{S}_{(\ell)} \in \{0, 1\}^{|\mathcal{B}^{(\ell)}| \times |\mathcal{V}_{\text{char}}|}$ which maps the indices of the language-specific vocabulary to the global character vocabulary. This allows us to project the compressed representation $\mathbf{A}'$ directly to d_{sonar} :

$$\text{ADAPTER}_{\text{V-Align}}(\mathbf{A}') = \text{softmax}(\mathbf{A}' \mathbf{W}_{(\ell)}^{\text{ctc}}) \mathbf{S}_{(\ell)} \mathbf{W}^{\text{char}} \quad (6.18)$$

Technically, this computes a weighted average of CHARSONAR character embeddings, weighted by the ASR model’s confidence. This adapter serves as a strong, parameter-free baseline.

- **Gated Dual Adapter:** While the *Vocabulary-Aligned Adapter* captures the explicit character mapping required for semantic alignment, the acoustic features contain paralinguistic information (e.g., prosody) that may not be fully captured by a rigid character projection. Furthermore, due to its definition, we are restricted by $|\mathcal{B}^{(\ell)}|$ which acts as the dimensionality of the adapter, and thus we cannot scale to more parameters. Thus, we define a secondary randomly-initialized branch with an MLP and a learned gating mechanism to balance the two branches.

The Dual Adapter output is defined as:

$$\text{ADAPTER}_{\text{Dual}}(\mathbf{A}') = \mathbf{G} \odot \text{ADAPTER}_{\text{V-Align}}(\mathbf{A}') + (1 - \mathbf{G}) \odot \text{MLP}(\mathbf{A}') \quad (6.19)$$

where MLP is a two-layer learnable residual branch with a nonlinearity.

$$\text{MLP}(\mathbf{A}') = \text{ReLU}(\mathbf{A}'\mathbf{W}^{\text{in}})\mathbf{W}^{\text{out}} \quad (6.20)$$

where $\mathbf{W}^{\text{in}} : d_{\text{mms}} \rightarrow d_{\text{mlp}}$ and $\mathbf{W}^{\text{out}} : d_{\text{mlp}} \rightarrow d_{\text{sonar}}$. The gating matrix $\mathbf{G} \in [0, 1]^{T' \times d_{\text{sonar}}}$ is learnable and dynamically computed via a sigmoid activation.

$$\mathbf{G} = \sigma(\mathbf{A}'\mathbf{W}^g) \quad (6.21)$$

where $\mathbf{W}^g : d_{\text{mms}} \rightarrow d_{\text{sonar}}$.

This allows the model to dynamically shift its reliance between the explicit character mapping and the learned acoustic correction based on the ambiguity of the input. Furthermore, notice how the dimensionality d_{mlp} of the secondary branch is now a hyperparameter that we can choose according to our capacity needs, while for the Vocabulary-Aligned Adapter, the dimensionality $|\mathcal{B}^{(\ell)}|$ is fixed, and usually relatively small (approx. 64).

Training Objective. To train the Dual Adapter (or finetune the Vocabulary-Aligned Adapter), we use ASR data (speech-transcript pairs). We freeze both the MMS speech encoder and the CHARSONAR text encoder. The adapter is trained to minimize the MSE

distance between the speech sentence embedding $\mathbf{h}^s$ and the SONAR sentence embedding $\mathbf{h}^x$ of the corresponding transcript:

$$\mathcal{L}_{\text{adapt}} = \text{MSE}(\mathbf{h}^s, \text{sg}[\mathbf{h}^x]) \quad (6.22)$$

Our approach aims to pull the speech representation into the multilingual semantic space established in the first stage, by steering only the connection between the MMS acoustic encoder and the CHARSONAR semantic encoder.

6.4 Experimental Setup

6.4.1 Datasets

Text Adaptation Data. We construct a diverse group of 75 languages in terms of family and script, and with varying degrees of resourcefulness, including 12 *new* languages, not present in the *base* set of 202 languages supported by SONAR. To train CHARSONAR we use a combination of human-labeled data and mined parallel data from NLLB (NLLB, 2024). We filter our billion-scale parallel corpus with BLASER-2.0-QE (Dale and Costajussà, 2024) to remove noisy pairs with scores below 4 (see Figure C.1 in the Appendix). For the group of 12 new languages, we used various publicly available sources of parallel data with languages part of the base set of SONAR. For validation and testing we used the dev (997 examples) and devtest (1012 examples) splits of FLORES+,¹ which is an extension of FLORES-200 (NLLB, 2024) including new languages and improvements to the base ones.

Speech Adaptation Data. To train our adapters that connect MMS and CHARSONAR we use ASR data for 33 languages. Our criteria for choosing these languages were the following:

- supported by SONAR (i.e., semantic modeling)
- supported by MMS (i.e., acoustic modeling)
- included in Common Voice (Ardila et al., 2020) (i.e., access to ASR training data)
- included in FLEURS (Conneau et al., 2023) (i.e., access to ST evaluation data)

¹huggingface.co/datasets/openlanguagedata/flores_plus

For training we utilize the train split of Common Voice version 17.0,² with various data sizes per language, ranging from 100 to 1M. We optionally utilize the train split of FLEURS which contains 2K examples for each language. Evaluation is done on the dev and test splits of FLEURS, which contain approximately 400 and 900 examples each.

Details regarding the languages and the amount of training data for both text and speech are available in Appendix (§C.1).

6.4.2 Model Architecture

The SONAR text encoder (Duquenne et al., 2023) has 24 layers, with dimensionality d_{sonar} of 1024, and a subword-based vocabulary $\mathcal{V}_{\text{sub}}$ with 256K tokens. The total parameter size of the encoder is 750M. Our CHARSONAR encoder follows the same architecture, apart from the character-based vocabulary $\mathcal{V}_{\text{char}}$ with 8K tokens, totaling 500M parameters. The CTC-based acoustic encoder, MMS (Pratap et al., 2023), has 48 layers with dimensionality d_{mms} of 1280, totaling 1B parameters.³ It uses language-specific layers (Houlsby et al., 2019) and CTC classification layers. The language-specific character-level vocabularies of MMS, $\mathcal{B}^{(\ell)}$, are different for each language, usually having around 64 tokens. Since the size of each vocabulary is also the hidden dimension of our Vocabulary-Aligned Adapter, this adapter version has approximately 200K parameters. The gated Dual Adapter uses an MLP with a hidden dimension d_{mlp} of 1024, having a total of 2.5M parameters. To generate a translation, either from text (subword- or character-tokenized) or speech, we utilize the SONAR text decoder which has 24 layers. For into-English generation we use the standard SONAR decoder, while for all other generation tasks we use the finetuned decoder, which according to Duquenne et al. (2023), and our observations here, performs better.⁴

6.4.3 Training Details

Text Adaptation. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 4e-4, inverse square root scheduler with linear warmup, a batch size of 12K examples, dropout of 0.1. We train for 128K updates. To balance the data distribution we up-sample

²huggingface.co/datasets/mozilla-foundation/common_voice_17_0

³huggingface.co/facebook/mms-1b-all

⁴github.com/facebookresearch/SONAR

languages with a temperature of 0.5, based on their total number of examples. We apply ASR-like text normalization by uncasing and removing punctuation to the source character-tokenized sequence c with a probability p^{norm} of 0.25, and inject character-based noise with a probability p^{noise} of 0.125. Specifically, each character in the source sentence can be deleted, replaced, or a new character is further added, each with a probability of 0.0025. These values were tuned in a small validation set of Common Voice to approximate the character-error-rate (CER) of MMS (Pratap et al., 2023). For replacement and addition, we sample a new character from the character distribution of that language.

Speech Adaptation. Both the MMS CTC acoustic encoder and the CHARSONAR encoder remain frozen during the speech adaptation training, and only the adapter that connects them is finetuned. The learning rate is set to $2e-4$, the batch size 500 examples, and the Vocabulary-Aligned Adapter dropout to 0.1, while for the MLP branch of the gated Dual Adapter we use a higher rate of 0.3.

6.4.4 Evaluation

We apply checkpoint averaging according to the dev set performance, and generate with a beam search of 5. We evaluate our models on two tasks: zero-shot translation and embedding similarity search.

Translation. Translation quality is measured with xCOMET-XL (Guerreiro et al., 2023).⁵ When the target language is not supported, we use case-sensitive detokenized BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017), computed with sacreBLEU (Post, 2018).

Similarity Search. Cross-lingual text-based similarity is measured against the English side of FLORES+, where we encode the source with CHARSONAR and the English target with the original SONAR encoder. Cross-modal speech-based similarity is measured against the text target side of FLEURS, where we encode the target text with the original SONAR encoder and the non-English source speech with our speech-enabled CHARSONAR encoder. We measure the xsim++ error rate, which is computed on the

⁵huggingface.co/Unbabel/XCOMET-XL

augmented English text target side of FLORES+ (for text) or FLEURS (for speech) with an additional 40K hard negatives. It serves as a better measure than a simple error rate (xsim) which has been found to saturate easily (Chen et al., 2023a).

6.5 Advancing Language Transfer: Text Adaptation Results

Before addressing the modality gap between speech and text, we must first validate the capabilities of the character-based text encoder. The core hypothesis of this chapter is that a character-level representation can generalize better across languages than subwords, particularly when the target embedding space is fixed. In this section, we present the text translation results, comparing the proposed CHARSONAR student against the SONAR teacher. We investigate the impact of the training objective, the effectiveness of the model in low-resource settings, and the source of its performance gains. Note that while we also evaluated cross-lingual embedding similarity (xsim++), we reserve those detailed results for the Appendix (see §C.2) to focus the main narrative on translation quality.

6.5.1 Initial Exploration and Objective Analysis

We begin by determining the optimal training strategy for aligning character sequences to the SONAR sentence embedding space. First we present an experiment that motivates the use of the interpolated MSE objective and then conduct an ablation study to confirm our hypothesis for the objective and understand the impact of the other mechanisms of our methodology. For the ablations here we train different CHARSONAR models on a subset of 15 languages.⁶

The Case for Interpolated MSE. A standard approach for student-teacher distillation is to minimize the Mean Squared Error (MSE) between the student’s embedding of a sentence and the teacher’s embedding of the same sentence (Reconstruction MSE). Alternatively,

⁶The group consists of 3 Uralic languages and 12 languages utilizing the Cyrillic script (see Tables C.1 and C.2 in the Appendix).

one could minimize the distance to the teacher’s embedding of a translation (Translation MSE).

However, we hypothesize that the SONAR space is not perfectly continuous and that the centroids of individual languages may not represent the optimal semantic point for cross-lingual transfer. To test this, we analyzed the geometry of the embedding space by decoding from the midpoint between two language representations. We use FLORES that is multi-way parallel, providing the same sentence is available in all languages.

For a sentence $\mathbf{x}$ in two non-English languages x^{ℓ_1}, x^{ℓ_2} , we encode them with SONAR to obtain $\mathbf{h}^{x_{\ell_1}}, \mathbf{h}^{x_{\ell_2}}$ and use the SONAR decoder to generate their standard translations into-English $\hat{\mathbf{y}}^{\ell_1 \rightarrow \text{eng}}, \hat{\mathbf{y}}^{\ell_2 \rightarrow \text{eng}}$. We also generate a third translation $\hat{\mathbf{y}}^{\text{avg} \rightarrow \text{eng}}$ from their interpolated sentence embedding $\mathbf{h}^{\text{avg}} = (\mathbf{h}^{x_{\ell_1}} + \mathbf{h}^{x_{\ell_2}})/2$. Then we can assess the quality of the individual language embedding spaces, and the interpolated space, by evaluating the translation quality of the three candidate translations.

Table 6.1 presents the results of this experiment, where we evaluate 3 grouping categories based on resourcefulness levels. Decoding from the interpolated embedding yields higher quality translations than decoding from the individual embeddings of either low-resource languages.⁷ This suggests that the interpolated regions between languages are semantically richer or more language-neutral than the language-specific clusters. This finding motivates us to use the interpolated MSE objective for training CHARSONAR, forcing the source sentences into the higher quality, interpolated regions of the SONAR space.

Pairs		xCOMET			$\Delta (\text{avg} - \ell_i)$	
ℓ_1	ℓ_2	$\mathbf{h}^{x_{\ell_1}}$	$\mathbf{h}^{x_{\ell_2}}$	$\mathbf{h}^{\text{avg}}$	ℓ_1	ℓ_2
Low	Low	0.788	0.795	0.864	+0.076	+0.069
Low	High	0.793	0.937	0.920	+0.137	-0.017
High	High	0.939	0.937	0.944	+0.005	+0.007

Table 6.1: xCOMET scores of translating from average (interpolated) embeddings compared to translating from individual embeddings, grouped by language pair resourcefulness level. Results on $\text{xxx} \rightarrow \text{eng}$ FLORES dev averaged over 50 randomly-sampled pairs per category. The Δ columns indicate the “Advantage” of decoding from the average embedding over the individual language embeddings.

Ablations on Design Choices. The upper section of Table 6.2 validates our hypothesis regarding the Interpolated MSE objective, where it outperforms both Reconstruction

⁷The only exception being the high-resource language, paired with a low-resource one.

and Translation MSE, as well as their combination. The lower section of Table 6.2 investigates the training schedule. We find that a two-stage process is optimal: pretraining with Reconstruction MSE to initialize the character modeling, followed by fine-tuning with Interpolated MSE to optimize the semantic alignment. Regarding data augmentation, adding noise or normalization does not improve performance in this text-only setting where inputs are error-free. However, we retain these augmentations in the final configuration as they prove essential for robustness against ASR errors in the speech experiments discussed later.

MSE Objective	Pretrain	Norm	Noise	xCOMET
Reconstruction	✗	✗	✗	0.929
Translation	✗	✗	✗	0.924
Reconstruction + Translation	✗	✗	✗	0.929
Interpolation	✗	✗	✗	0.931
Interpolation	✓	✗	✗	0.934
Interpolation	✓	✓	✗	0.934
Interpolation	✓	✓	✓	0.934

Table 6.2: Ablations on design choices for CHARSONAR trained on a group of 15 languages. Text translation average xCOMET scores on FLORES dev 15→eng.

6.5.2 Scaling-up Language Coverage

Based on the findings of our initial exploration, we scale up the training of CHARSONAR to 75 languages, using the Interpolated MSE objective and the two-stage training schedule with Reconstruction pretraining. For the subset of “New” languages not supported by the original SONAR teacher, we relied on the Translation MSE objective. We compare our approach against two baselines: the original SONAR encoder and a SONAR-75 model that we finetuned on the same data as CHARSONAR to ensure a fair comparison with a subword-based encoder. We also include NLLB-1.3B as a topline; unlike the SONAR family, NLLB uses a full encoder-decoder attention mechanism and is not constrained by a fixed-size sentence bottleneck.

Table 6.3 presents the translation results grouped by language resource availability. The character-based encoder demonstrates a clear advantage:

- **Overall Performance:** CHARSONAR-75 consistently outperforms the comparable SONAR-75 baseline across all resource groups. Notably, it also surpasses the NLLB-1.3B topline in overall xCOMET scores.

- **Low-Resource Gains:** The performance gap is most distinct in the 21 Low-Resource languages. In this group, CHARSONAR achieves the largest improvement over SONAR (+0.007 points). This empirical evidence supports the claim that character-level modeling mitigates the fragmentation issues inherent in subword tokenization for data-scarce languages.
- **Zero-shot Generalization:** To evaluate the robustness of our method on unseen languages, we trained models solely on the 63 base languages and evaluated them on the 12 new languages in a zero-shot setting (bottom section of Table 6.3). CHARSONAR-63 generalizes better to unseen languages than the subword-based SONAR-63, achieving a gain of +0.013 xCOMET points. This improvement suggests that while subword tokenizers struggle with the unknown morphology or script of unseen languages, character sequences offer a universal interface that allows the model to leverage script and family similarities more effectively.

Model	Low (21)	Med (21)	High (21)	All (63)	New (12)
<i>Previous Works (trained on the 200 base languages)</i>					
NLLB-1.3B (NLLB, 2024)	0.877	0.914	0.949	0.913	0.454 [†]
SONAR (Duquenne et al., 2023)	0.851	0.894	0.944	0.897	0.450 [†]
<i>This work (trained on 75 languages; 63 base + 12 new)</i>					
SONAR-75	0.882	0.909	0.948	0.913	0.859
CHARSONAR-75	0.889	0.914	0.949	0.917	0.863
Δ	0.007	0.005	0.001	0.004	0.004
<i>This work (trained only on 63 base languages)</i>					
SONAR-63	0.882	0.909	0.947	0.913	0.517 [†]
CHARSONAR-63	0.889	0.914	0.949	0.917	<u>0.530[†]</u>
Δ	0.007	0.005	0.002	0.004	0.013

Table 6.3: Translation xCOMET (↑) scores on FLORES devtest (xxx→eng), grouped by resourcefulness level (Low/Med/High) and New languages. † indicates zero-shot evaluation. In **bold** is the best overall and underlined is the best zero-shot. Δ refers to the difference between CHARSONAR-N and SONAR-N models. Appendix: See Table C.4 for the corresponding cross-lingual similarity search results, and Table C.3 for the per-language results.

6.5.3 Analysis

Increased Compute vs Cross-lingual Transfer. Character-level modeling inherently results in input sequences that are approximately $3\times$ longer than their subword counterparts. This increases the FLOP count for the CHARSONAR encoder relative to SONAR. It is

therefore crucial to verify whether the performance gains observed in Table 6.3 are a result of better cross-lingual transfer or simply a consequence of the increased computational capacity.

We isolated these factors by training both architectures on a single language: English. If the gains were driven by compute or granularity alone, CHARSONAR should outperform SONAR in this monolingual setting. However, as shown in Table 6.4, the character-based model does not exhibit an advantage in the single-language scenario. In fact, it performs slightly below the subword baseline. This result strongly indicates that the superiority of CHARSONAR in multilingual settings stems from its ability to facilitate cross-lingual knowledge transfer rather than raw computational scaling.

Efficiency Implications. To understand the computational overhead stemming from the longer character-level sequences, we measured the average inference time on FLORES dev. Table 6.5 shows that while the input token count increases by a factor of 3.2, the inference time increases by only $1.1\times$. This modest overhead is due to the bottleneck architecture, which decouples the decoder’s computational cost from the source sequence length, providing a favorable trade-off.

Model	BLEU	chrF++
<i>Previous Works</i>		
NLLB-1.3B (NLLB, 2024)	17.4	45.3
SONAR (Duquenne et al., 2023)	15.6	44.0
<i>This Work (trained only on English)</i>		
SONAR-Eng	15.9	44.8
CHARSONAR-Eng	15.8	44.7

Table 6.4: Text translation results for from-English generation. Average BLEU and chrF++ scores on FLORES devtest (eng→200).

Model	Average Tokens per Sequence	Inference Time (s)	
		Sequential	Batched
SONAR	49	0.84	127
CHARSONAR	158 ($\times 3.2$)	0.94 ($\times 1.1$)	142 ($\times 1.1$)

Table 6.5: Average number of tokens and average inference time measured on FLORES dev. Sequential inference (no batching), and length-based batching (5K tokens per batch).

6.6 Closing the Modality Gap: Speech Adaptation Results

Having validated the CHARSONAR text encoder, we now turn to the central challenge of this chapter: bridging the modality gap between speech and text. In Ch. 5, we aligned speech to subwords using Optimal Transport. Here, we test the hypothesis that aligning speech to characters—via the proposed CHARSONAR space—enables more effective zero-shot transfer, particularly for languages where subword tokenization is suboptimal.

In this section, we present results on Speech Translation (ST) and Cross-Modal Similarity Search. Unless otherwise stated, we use the CHARSONAR-75 encoder described in §6.5 as the semantic encoder for our speech encoder. The acoustic encoder is the frozen MMS (1B), and we train only the lightweight adapter using ASR data (mainly Common Voice), meaning the system sees *no* paired speech-to-translation data.

6.6.1 Adapter Design and Robustness

We first conduct an ablation study to determine the optimal configuration for our cross-modal adapter as well as the impact of the augmentation used to train the CHARSONAR encoder (§6.3.3). We evaluate performance on zero-shot ST (xxx→eng) using FLEURS dev for four languages representing different linguistic families and different ASR resource levels: Estonian (est), Russian (rus), Turkish (tur), and Spanish (spa).

Adapter Configuration			Encoder Augmentations			xCOMET			
Type	Dim	Trained	Norm	Noise	est	rus	tur	spa	Average
V-ALIGN	≈64	✗	✗	✗	0.845	0.849	0.828	0.837	0.840
V-ALIGN	≈64	✓	✗	✗	0.901	0.910	0.877	0.890	0.894
MLP	256	✓	✗	✗	0.837	0.872	0.831	0.878	0.854
MLP	1024	✓	✗	✗	0.882	0.889	0.869	0.889	0.882
DUAL	256	✓	✗	✗	0.914	0.912	0.889	0.888	0.901
DUAL	1024	✓	✗	✗	0.911	0.909	0.894	0.905	0.905
DUAL	1024	✓	✓	✗	0.914	0.910	0.891	0.897	0.903
DUAL	1024	✓	✓	✓	0.915	0.923	0.906	0.905	0.912

Table 6.6: Ablations in speech adaptation training. Zero-shot Speech Translation (xxx→eng) results on FLEURS dev.

Vocabulary-Aligned vs. MLP Adapters. As described in the methodology (see §6.3.4), we proposed a Vocabulary-Aligned (V-Align) Adapter initialized directly from the MMS

CTC head and the CHARSONAR embedding table. Table 6.6 compares this against a standard randomly initialized MLP-style adapter, which is essentially the secondary branch in the Dual Adapter. We observe that the Vocabulary-Aligned Adapter, despite its minimal size (≈ 64 hidden dimension), outperforms the significantly larger MLP adapter ($d_{\text{mlp}} = 1024$) in low and medium resource settings. For example, on Estonian (3K examples), the Vocabulary-Aligned Adapter achieves 0.901 COMET versus 0.882 for the random one. This confirms that leveraging the existing alignment between the CTC character vocabulary and the CHARSONAR vocabulary provides a strong inductive bias for alignment. The utility of the initialization is also evident by the very competitive performance of the Vocabulary-Aligned Adapter, without any training (Trained $\times$).

The Dual Adapter. For high-resource languages like Spanish, the capacity of the Vocabulary-Aligned Adapter becomes a bottleneck. The proposed Dual adapter addresses this by gating the output of the Vocabulary-Aligned Adapter with a larger, randomly initialized MLP branch. This configuration yields the best performance across the board, effectively combining the stability and pretrained knowledge of the first branch with the capacity of the secondary branch.

Robustness to ASR Noise. A critical finding in this exploration is the importance of the text encoder’s robustness. In §6.5.1, we noted that adding noise and normalization during text training did not improve text-to-text results. However, Table 6.6 shows that using a “robust” CHARSONAR encoder (trained with Normalization and Noise augmentations) is beneficial for speech adaptation. The performance of the Dual Adapter improves from 0.911 to 0.915 COMET when connecting to the robust text encoder. This is because the output of the acoustic adapter naturally contains jitter, deletions, and normalization artifacts (e.g., lack of punctuation) that resemble the synthetic noise injected during text training.

6.6.2 Zero-shot Speech Translation

Baselines. We now evaluate the proposed system on an extensive scale of 33 languages. We compare our zero-shot approach S-CHARSONAR-75, which connects MMS and CHARSONAR with a lightweight adapter using only ASR data, against state-of-the-art supervised E2E ST systems and strong cascades. For the supervised models we report

numbers with WHISPER-LARGE-V3⁸ (Radford et al., 2023) and SEAMLESSM4T-LARGE-V2⁹ (Barrault et al., 2025). For the cascades we report numbers with different combinations of ASR and MT models. For the ASR models we use a CTC model, MMS, or an Attention-based Encoder-Decoder (AED) model, WHISPER-LARGE-V3. For the MT model, we use either NLLB-1.3B or SONAR. We also report numbers by combining the previous ASR models with our SONAR versions trained for this work, which are directly comparable to our S-CHARSONAR-75 model.

Table 6.7 presents the results on FLEURS test. The comparison reveals several key insights regarding the effectiveness of our proposed framework:

- **Initialization Matters:** Our proposed S-CHARSONAR using the Vocabulary-Aligned Adapter (row 11) can work out-of-the-box and without any training outperforms WHISPER by a large margin overall (0.714→0.812). The gap is particularly large for low-resource languages (0.598→0.772). This finding indicates that the input space of our character-based encoder is fully compatible with the output space of MMS with the semantic initialization of our Vocabulary-Aligned Adapter.
- **Training Efficiency:** Subsequently, by training this adapter with ASR data (row 12), we surpass the previous state-of-the-art SEAMLESSM4T-LARGE-V2. The benefits of cross-modal transfer from CHARSONAR are evident for some extremely low-resource languages such as Asturian, where with only 400 examples it surpasses SEAMLESSM4T by 0.1 xCOMET points (see Table C.7 in Appendix).
- **Capacity Scaling:** By using the Dual Adapter (row 13) we observe further gains for medium/high-resource languages, where there are enough data to learn the large, but randomly initialized, MLP branch. Finally, we show that by training the Dual Adapter additionally on the 2K per-language examples from FLEURS train we can surpass all previous configurations of supervised E2E ST and cascades.
- **Benefits to Cascade Systems:** Apart from the strong cross-modal transfer showcased by our speech adaptation of CHARSONAR, significant improvements are also obtained from utilizing CHARSONAR in cascade systems. Specifically a cascade with MMS (row 9) outperforms all other cascades (rows 3-8) and is on par with SEAMLESSM4T-LARGE-V2 in low/medium-resource settings. We hypothesize

⁸huggingface.co/openai/whisper-large-v3

⁹huggingface.co/facebook/seamless-m4t-v2-large

that a strong driver for this performance is the ASR-like augmentations with which trained CHARSONAR.

id	Model	Text Surface	Total Params	Adapter Train Params	Adapter Train Data	Low (11)	Med (11)	High (11)	All (33)
(Previous Works) Supervised E2E ST									
1	WHISPER-LARGE-V3	/	1.5B	/	/	0.598	0.754	0.790	0.714
2	SEAMLESSM4T-LARGE-V2	/	2.3B	/	/	<u>0.829</u>	<u>0.889</u>	<u>0.901</u>	<u>0.873</u>
(Previous Works) Cascade ST									
3	MMS → NLLB-1.3B	Subword	2.3B	/	/	0.786	0.834	0.822	0.814
4	WHISPER → NLLB-1.3B	Subword	2.8B	/	/	0.717	0.870	0.863	0.817
5	MMS → SONAR	Subword	2.3B	/	/	0.757	0.839	0.824	0.807
6	WHISPER → SONAR	Subword	2.8B	/	/	0.684	0.869	0.861	0.804
(This Work) Cascade ST									
7	MMS → SONAR-75	Subword	2.3B	/	/	0.811	0.870	0.854	0.845
8	WHISPER → SONAR-75	Subword	2.8B	/	/	0.721	0.871	0.865	0.819
9	MMS → CHARSONAR-75	Char	2.3B	/	/	0.833	0.889	0.875	0.866
10	WHISPER → CHARSONAR-75	Char	2.8B	/	/	0.755	0.893	0.882	0.843
(This Work) Zero-shot E2E ST									
11	S-CHARSONAR-75 w/ V-ALIGN	Char	2.3B	0	/	0.772	0.833	0.831	0.812
12	S-CHARSONAR-75 w/ V-ALIGN	Char	2.3B	0.2M	CV	0.837	0.893	0.894	0.875
13	S-CHARSONAR-75 w/ DUAL	Char	2.3B	2.5M	CV	0.615	0.899	0.902	0.805
14	S-CHARSONAR-75 w/ DUAL	Char	2.3B	2.5M	CV+FLEURS	0.853	0.906	0.910	0.889

Table 6.7: Speech Translation average xCOMET scores on FLEURS test xxx→eng. Results reported separately according to language resourcefulness level, based on availability of ASR Common Voice data. Underlined are the previous best scores. **Highlighted** are our scores with char-based models that are at least on par with the previous best. In **bold** are the best overall. See Table C.7 in the Appendix for the detailed per-language results.

6.6.3 Cross-modal Similarity Search

Beyond translation generation, we evaluate the quality of the embeddings themselves. We perform a cross-modal (and cross-lingual) retrieval task where the query is a non-english speech segment (encoded by our system) and the target is the corresponding English text (encoded by SONAR).

Table 6.8 shows the xsim++ error rates. Our S-CHARSONAR encoder with the Vocabulary-Aligned Adapter achieves an error rate of 12.9, which improves to 10.2 with the Dual Adapter. Notably, this outperforms the original SONAR speech encoders (Duquenne et al., 2023) by a large margin of 4.1 points. Although they utilize a strong w2v-BERT (Chung et al., 2021) and more supervised data, they do not transfer knowledge from an acoustic model (i.e., MMS) nor the text modality (i.e., CHARSONAR). Overall, our results here

indicate that aligning to a character-based text space preserves semantic discriminability better than aligning to subwords or using purely acoustic pretraining objectives.

Model	xsim++ (↓)
<i>(Previous Works) Unified Encoders</i>	
SONAR w2v-BERT Speech Encoders (Duquenne et al., 2023)	14.3
<i>Cascades of Speech and Text Encoders</i>	
MMS → SONAR	15.7
MMS → SONAR-75	12.7
MMS → CHARSONAR-75	10.7
<i>This Work</i>	
S-CHARSONAR-75 w/ V-ALIGN	11.6
S-CHARSONAR-75 w/ DUAL	9.4

Table 6.8: Cross-modal and cross-lingual retrieval error rates (xsim++) on FLEURS test. Non-english source speech is encoded with the models presented here, and target English text is encoded with the original SONAR encoder. Results are average over the 26 languages supported by both S-CHARSONAR-75 and the speech encoders from Duquenne et al. (2023).

6.6.4 Impact of Text Surface

Model	oci # 0.3k	est # 3k	tur # 30k	spa # 330k
S-SONAR-75 w/ MLP	0.199	0.223	0.795	0.841
S-CHARSONAR-75 w/ MLP	<u>0.202</u>	<u>0.877</u>	<u>0.868</u>	<u>0.912</u>
S-CHARSONAR-75 w/ V-ALIGN	0.795	0.910	0.885	0.917
S-CHARSONAR-75 w/ DUAL	0.707	0.912	0.903	0.920

Table 6.9: Zero-shot Speech Translation xCOMET results on FLEURS test xxx→eng for subword-based vs character-based speech encoders with adapters. Underlined is the best among MLP Adapters and with **bold** is the best overall. The number under each language code indicates the number of ASR examples in thousands.

Finally, we address the core question of this chapter: *Is the character level actually better for speech adaptation than the subword level?* To test this, we replaced the CHARSONAR text encoder with the SONAR (subword) encoder and trained an adapter to map the MMS output to subword embeddings expected by SONAR. We mean-pool the indices of the compressed acoustic representation of MMS that belong to the same subword, as predicted by the CTC. The Vocabulary-Aligned Adapter version is not possible in this setting due to incompatibility with a proper initialization, and thus we conduct this experiment with the randomly-initialized MLP adapter.

Table 6.9 presents the zero-shot speech translation results for four languages of various resourcefulness levels: Occitan, Estonian, Turkish, and Spanish. Our results indicate that although we can learn an adapter to connect MMS and the subword-based SONAR, the quality is limited. It only works sufficiently well in high resource settings (Turkish, Spanish) and overall is several points behind the character-based speech encoder. This highlights both the data-efficiency and cross-modal adaptability of our proposed method.

6.7 Conclusions

In this chapter, we addressed the limitations of subword-based modeling identified in Ch. 5 by proposing a shift towards character-level representations for cross-modal alignment. We introduced CHARSONAR, a method that adapts a fixed-size multilingual sentence embedding space to accept character sequences as input, facilitating robust knowledge transfer across languages. We furthermore proposed a lightweight adapter that enables speech inputs through CHARSONAR, achieving impressive cross-modal transfer from the text modality, which led to state-of-the-art Speech Translation results, despite being a zero-shot method.

Summary of Contributions. Our investigation yielded two primary conclusions regarding the granularity of intermediate representations in zero-shot translation systems. First, in the text modality, we demonstrated that character-level modeling significantly improves cross-lingual transfer compared to subword-based baselines. By removing the rigid dependency on a fixed tokenizer, CHARSONAR avoids the fragmentation issues that plague low-resource languages and exhibits superior generalization to unseen languages and scripts. Second, in the speech modality, we showed that the character-level space is naturally more compatible with acoustic representations than the subword space. This compatibility allowed us to bridge the modality gap using a lightweight, data-efficient adapter trained solely on ASR data. Remarkably, our method achieves state-of-the-art results in zero-shot speech translation, outperforming heavy supervised baselines like SEAMLESSM4T in low-resource settings, and even functioning effectively without any training parameters by leveraging the inductive bias of the pretrained components.

The Limits of Semantic Alignment. With this chapter, we conclude Part III of the thesis, which focused on solving the data scarcity problem in Speech Translation via Cross-Modal Alignment. Both ZEROSWOT (Ch. 5) and CHARSONAR (Ch. 6) operate on the premise that the semantic content of speech can be fully mapped to a text-based embedding space. We have successfully proven that this mapping enables effective zero-shot translation; by aligning speech to the rich semantic vector space of massively multilingual text models, we can bypass the need for expensive speech-to-translation corpora.

However, this reliance on text-based embedding spaces introduces a fundamental trade-off. By forcing speech representations to mimic text embeddings, whether at the subword or character level, we inherently treat speech as “text with noise”. While this is sufficient for transferring *verbal* content (what is said), it explicitly discards the *non-verbal* and paralinguistic information (how it is said) that is unique to the audio signal. The rich prosodic features of speech—intonation, stress, rhythm, and emotion—are flattened during the compression into the SONAR space.

Therefore, while we have established a robust methodology for transferring meaning in data-scarce scenarios, a truly comprehensive speech translation system must also account for the expressive dimensions of the source utterance. In Part IV, we turn our attention to this missing dimension. Specifically, Ch. 7 will investigate the capacity of modern End-to-End systems to preserve prosody and the extent to which current evaluation metrics capture these acoustic nuances.

Part IV

Prosody-Aware Evaluation

7 Towards Prosody-aware Evaluation: Speech Is More Than Words

This chapter is based on our work “Speech Is More Than Words: Do Speech-to-Text Translation Systems Leverage Prosody?”, published in **WMT 2024** (Tsiamas et al., 2024b) and developed during a research internship at Apple AI/ML between April 2024 – July 2024.

7.1 Introduction

The previous parts of this thesis addressed the structural and data-related limitations of End-to-End (E2E) Speech Translation (ST). In Part II, we focused on the segmentation of the input audio, while in Part III, we developed methods to bridge the modality gap and combat data scarcity in order to enable zero-shot ST. Common to these approaches was the treatment of the speech signal primarily as a carrier of lexical content. In Ch. 6, specifically, we aligned speech representations to a character-based text embedding space. While effective for transferring verbal meaning, this text-centric view inherently discards the non-verbal and paralinguistic information unique to the acoustic modality. Speech is more than a sequence of words. It contains features such as stress, intonation, and rhythm, collectively known as prosody, that significantly influence semantics (Ladd, 1980; Bolinger, 1989).

Prosody serves multiple communicative functions. It directs focus, clarifies syntax (Bolinger, 1961; Halliday, 1967), and conveys the speaker’s emotional state (Banse and Scherer, 1996). Consider the single word “Really?”. Depending on the intonation, it can express genuine interest, surprise, or disbelief. In translation, these acoustic cues can be decisive. Table 7.1 illustrates two examples where prosodic variations of the same English sentence require distinct translations in German. For instance, stressing “German” in “These are German teachers” implies the nationality of the teachers, and needs to be translated as “Deutschlehrer”. On the other hand, stressing “teachers” implies they teach the German language, in which case it is translated to “deutsche Lehrer”.

Example: <i>These are German teachers.</i>	
Prosody A	These are GERMAN teachers.
Explanation	Teachers from Germany
Translation	Dies sind Deutschlehrer .
Prosody B	These are German TEACHERS .
Explanation	Teachers that teach German
Translation	Dies sind deutsche Lehrer .
Example: <i>John laughed at the Party.</i>	
Prosody A	John LAUGHED [pause] at the Party.
Explanation	Laughed while at the party (literal)
Translation	John lachte während der Party.
Prosody B	John LAUGHED AT [pause] the Party.
Explanation	Ridiculed the party (idiomatic)
Translation	John lachte über die Party.

Table 7.1: Examples of prosody-aware Speech Translation from English to German. Differences in stress or pauses change the underlying meaning and necessitate different translations.

The ability to capture these nuances is a theoretical advantage of E2E models over cascaded systems. Cascaded architectures, which pipe Automatic Speech Recognition (ASR) outputs into Machine Translation (MT) models, typically lose prosodic information at the transcription bottleneck. E2E models, by contrast, have access to the raw speech signal throughout the translation process (Sperber and Paulik, 2020). Despite this potential, it remains unclear whether current E2E systems effectively leverage prosody in practice. Prior research offers limited insight, often relying on anecdotal evidence (Huang et al., 2023b) or focusing on narrow subsets of prosodic phenomena (Zhou et al., 2024; Chen et al., 2024).

We identify three primary obstacles to reliable prosody evaluation in ST.

1. Existing benchmarks often lack prosody-rich spontaneous speech or do not provide translations that explicitly reflect source-side prosody (Wang et al., 2020a).
2. Standard evaluation paradigms are insufficient. First, standard references tend to be agnostic to acoustic nuance. Second, general-purpose metrics like BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020) are insensitive to the subtle lexical

changes induced by prosody. They are designed to measure general semantic overlap rather than the specific resolution of ambiguity.

3. Creating large-scale, high-quality resources for prosodic evaluation is difficult. Manual collection of minimal pairs, defined as sentences that differ only in prosody and meaning, is resource-intensive.

In this chapter, we address these challenges by introducing CONTRAPROST (Contrastive Prosody ST), a benchmark and evaluation methodology designed to isolate prosody awareness. Our approach draws on behavioral testing ([Ribeiro et al., 2020](#)) and contrastive evaluation ([Sennrich, 2017](#)) to tackle the limitation of standard ST evaluation metrics like BLEU and COMET. We address the data scarcity and resource creation bottlenecks by developing a semi-automated pipeline that utilizes Large Language Models (LLMs) to generate ambiguous sentences and Controllable Text-to-Speech (cTTS) to synthesize the corresponding prosodic variants. This allows us to cover a wide range of phenomena, including sentence stress, prosodic breaks, intonation patterns, emotional prosody, and politeness.

We evaluate the prosody awareness of 31 model variants across three language pairs: English to German, Spanish, and Japanese. The main contributions and findings of this chapter are as follows:

- We introduce CONTRAPROST, a comprehensive benchmark that enables the systematic evaluation of prosody in speech translation.
- We employ a double-contrastive evaluation metric and find that while ST models possess some internal representation of prosody, this signal is often too weak to alter the final generated translation.
- We demonstrate that E2E systems consistently outperform cascades on our benchmark, confirming their architectural advantage ([Sperber and Paulik, 2020](#)).
- We observe that cascaded systems can capture limited prosodic information, but this depends heavily on the surface form of the intermediate transcript, such as punctuation or the use of synonyms.

- We open-source the CONTRAPROST benchmark to facilitate for further research in prosody-aware evaluation in Speech Translation.¹

7.2 Related Work

Research on prosody has traditionally concentrated on speech synthesis rather than translation. In this section, we review prior work in Text-To-Speech (TTS) and Speech-to-Speech (S2S) systems, before discussing the limited studies available in Speech Translation.

7.2.1 Prosody in Speech Synthesis

Prosody is a central component of TTS research (Kohler, 1991). The primary focus in this field is either transferring prosody from a reference audio to the synthesized speech (Skerry-Ryan et al., 2018) or explicitly encoding it to control the output (Pamisetty and Sri Rama Murty, 2022). To evaluate these capabilities, researchers have developed datasets specifically designed to measure prosody transfer, covering categories similar to those we study in this chapter (Torresquintero et al., 2021).

Similarly, Speech-to-Speech systems prioritize prosody to generate expressive speech outputs in the target language (Aguero et al., 2006; Do et al., 2017; Barrault et al., 2025). In these architectures, the goal is often to preserve the paralinguistic features of the source speaker in the synthesized target audio. However, this line of research focuses on the acoustic realization of the translation rather than how prosody influences the lexical selection of the translated text.

7.2.2 Prosody in Speech-to-Text Translation

The role of prosody in ST has received significantly less attention. Most existing benchmarks focus on content preservation rather than paralinguistic nuances. Recently, Chen et al. (2024) introduced a dataset for emotional prosody derived from TV series. They

¹github.com/apple/ml-speech-is-more-than-words

demonstrated that finetuning ST models with emotion labels can improve general translation quality. However, their study is limited to emotional prosody and does not address other linguistic phenomena such as stress or intonation patterns.

Closer to our work, [Zhou et al. \(2024\)](#) investigated the prosody awareness of the WHISPER model ([Radford et al., 2023](#)) in both E2E and cascaded settings. Their study focused on a specific grammatical phenomenon: Korean *wh*-phrases. They used contrastive likelihood to evaluate the models and found evidence that the E2E approach outperforms the cascade. Our work expands upon this by proposing a broader double-contrastive benchmark that covers five distinct prosodic categories and evaluates a wider range of ST models across three language pairs.

Finally, [de Seyssel et al. \(2023\)](#) proposed a benchmark to evaluate prosody awareness in self-supervised acoustic representations. Their findings suggest that these representations encode prosodic information and that model size correlates positively with prosody awareness. In contrast, our results in §7.7 indicate that while representations may encode prosody, this does not necessarily translate to prosody-aware generation in ST models, regardless of model size.

7.3 The CONTRAPROST Benchmark

CONTRAPROST is composed of double-contrastive examples designed to isolate the effect of prosody on translation. Unlike standard contrastive evaluation, which typically pairs a single source with correct and incorrect targets, our benchmark employs a dual-branch structure. Each example consists of a single English source text x that is semantically ambiguous in its written form. This text is paired with two distinct speech signals, s^a and s^b , which correspond to differing semantic interpretations due to prosodic cues. These interpretations necessitate two distinct translations in the target language, y^a and y^b . See Table 7.1 for two illustrative examples.

We identify five categories of prosodic phenomena where acoustic features significantly influence sentence semantics. These categories serve as the foundation for our data generation pipeline.

7.3.1 Sentence Stress

Sentence stress is manifested through increased loudness, vowel length, or higher pitch (Fry, 1955). These acoustic markers place emphasis on specific words within a sentence, which can shift the semantic focus and alter the required translation (Wagner, 2020). We categorize prosodic stress into four distinct subcategories based on the linguistic function of the emphasis:

- **Contrastive Stress:** Highlights differences or corrects previous statements by emphasizing contrasts between elements (Bolinger, 1961).
- **New vs. Given Information:** Differentiates between information that is already known and information that is introduced for the first time (Halliday, 1967).
- **Relational vs. Descriptive Adjectives:** Stress on either the adjective or the noun determines whether an attributive adjective is used in a relational or descriptive capacity (Liberman and Sproat, 1992).
- **Focus-Sensitive Operators:** Stress indicates the scope of adverbs of quantification, such as “only” or “just,” which changes the meaning of the utterance (Halliday, 1967; Jackendoff, 1972).

7.3.2 Prosodic Breaks

This category examines the existence or placement of pauses and breaks in the flow of speech. These breaks, primarily associated with tempo, create phrasal boundaries that help disambiguate syntax and sentence structure (Bolinger, 1989). We utilize six subcategories, as outlined in Hirschberg (2017), to capture different syntactic ambiguities:

- **Direct vs. Indirect Statements:** A prosodic break differentiates between a direct quote and an indirect report (Klewitz and Couper-Kuhlen, 1999; Jansen et al., 2001).
- **Restrictive vs. Non-Restrictive Clauses:** Pauses indicate whether a clause provides essential or supplementary information (Nespor and Vogel, 1986).
- **Attachment Ambiguities:** The location of a break resolves whether a trailing phrase attaches to the verb phrase or the noun phrase (Pynte, 1996).
- **Phrasal Verbs:** Prosody distinguishes between the literal meaning of a verb plus a preposition and their idiomatic phrasal verb combination (Price et al., 1991).

- **Modifier Scope:** Breaks signal whether a modifier applies to a single noun or a list of nouns ([Hirschberg, 2017](#)).
- **Complementizer vs. Parenthetical:** The phrasing indicates whether an intermediate clause acts as a complementizer or a parenthetical statement ([Dehé, 2014](#)).

7.3.3 Intonation Patterns

Intonation patterns concern the modality of the sentence. We specifically look at the distinction between statements, characterized by a falling tone, and declarative questions, characterized by a rising tone ([Gunlogson, 2002](#)). In many languages, this distinction requires different lexical or grammatical structures in translation.

7.3.4 Emotional Prosody

The emotional tone of a speaker can alter the semantics of an utterance. Emotional prosody is typically conveyed through variations in pitch, tempo, and loudness ([Banse and Scherer, 1996](#)). For instance, happiness is associated with higher values in pitch and tempo, while sadness exhibits lower values across these features ([Larrouy-Maestri et al., 2024](#)). We consider the seven basic emotions : happy, sad, angry, disgust, surprisal, fear, and neutral ([Ekman and Friesen, 1971](#)). By constructing all possible pairs among these emotions, we form 21 subcategories of emotional contrast.

7.3.5 Politeness

The level of politeness in speech influences the pragmatic context of a conversation and often dictates the register used in translation. Non-verbal cues convey this information; a polite tone is generally associated with a higher pitch and smooth rhythm, whereas an impolite tone is manifested through low pitch, irregular rhythm, and extreme loudness levels ([Culpeper et al., 2003](#); [Culpeper, 2011](#)).

7.4 Data Generation Methodology

Collecting a large-scale dataset of minimal prosodic pairs from natural speech is prohibitively expensive and difficult. To address this, we employ a semi-automated generation pipeline that leverages the capabilities of Large Language Models (LLMs) for text generation and translation, combined with Controllable Text-to-Speech (cTTS) for audio synthesis. While natural speech is the gold standard, it lacks the “minimal pair” property required to isolate prosody from lexical content. Our synthetic pipeline allows for the *ceteris paribus* (all else being equal) evaluation that natural datasets cannot provide.

This pipeline, depicted in Figure 7.1, consists of four stages: prosodic example generation, oracle translation, speech synthesis, and quality assessment.

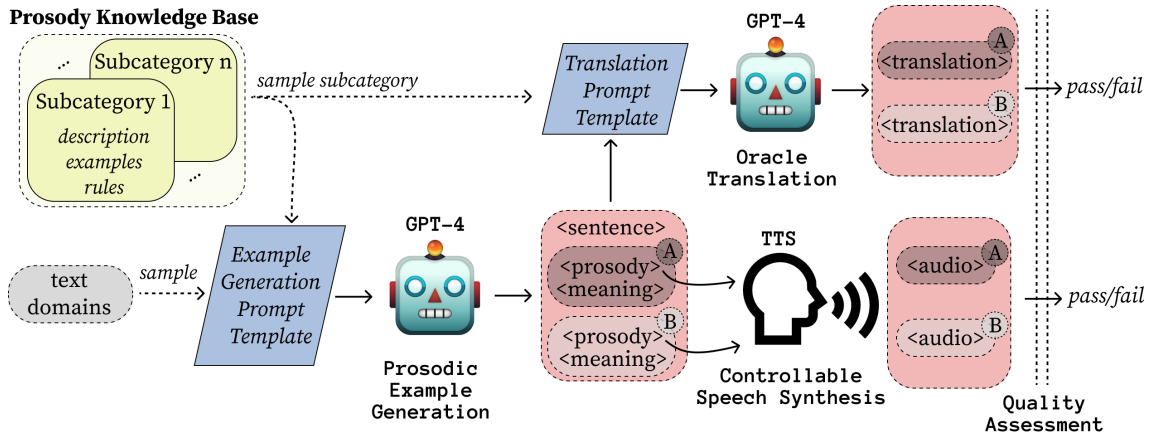


Figure 7.1: The Data Generation process for CONTRAPROST. We utilize GPT-4 to generate ambiguous sentences and their translations based on specific prosodic categories, followed by controllable TTS and rigorous quality filtering.

7.4.1 Prosodic Example Generation

For each prosodic category defined in §7.3, we prompt GPT-4 (OpenAI, 2024) to generate ambiguous English sentences (x). The prompt structure, shown in Figure 7.2, employs in-context learning (Brown et al., 2020) by providing the model with hand-crafted examples that illustrate the specific prosodic phenomenon (see §D.1 in the Appendix for some examples). The model is instructed to produce the sentence x , along with two textual descriptions of the prosody (p^a, p^b) and their corresponding semantic interpretations (i^a, i^b).

Role: You are a helpful assistant with expert knowledge in linguistics, speech, and prosody.

Task: Come up with examples of English sentences where different prosody would change the meaning of the sentence significantly.

(1) Category Details: [Details for current Subcategory]

(2) In-Context Examples: Here are some examples to guide you: [List of Examples]

(3) Constraints: Strictly follow these rules: [List of Rules, e.g., “do not include prosodic annotations in the sentence”]

(4) Self-Correction: Provide a rating of how significant is the difference between the two meanings.

(5) Generation: Generate n such examples, with rating as high as possible, in the domain of [Domain].

Figure 7.2: Prompt template used for generating prosodic examples (Prompt 1). The model generates sentences conditioned on specific linguistic constraints and text domains.

To ensure high quality and diversity, we incorporate several conditioning strategies. First, we implement a self-reflection mechanism (Huang et al., 2023a), where the model rates its own generations based on the distinctiveness of the two interpretations (Item 4 in Figure 7.2). Second, to prevent repetitive outputs and ensure broad coverage, we condition the generation on specific text domains (Chung et al., 2023). We first prompt GPT-4 to generate a list of relevant domains for each subcategory (e.g., legal testimonies, casual conversation), and subsequently generate a batch of 10 examples for each domain. This hierarchical generation approach yields a diverse set of candidate sentences rooted in varied semantic contexts.

7.4.2 Oracle Translation

Recent research indicates that LLMs can achieve high-quality machine translation, particularly for high-resource languages (Robinson et al., 2023; Alves et al., 2023; Zhang et al., 2023a; Vilar et al., 2023), and can effectively handle translation factors such as emotion (Brazier and Rouas, 2024). We leverage this capability by using GPT-4 as a prosody-aware “oracle” translator.

The translation process is two-fold. First, we prompt the LLM to translate the source sentence x into the target language translation y , without using any prosodic descriptions or interpretations. Then, the LLM is provided with full access to the prosodic descriptions and semantic interpretations (Prompt 2, Figure 7.3) and is tasked to generate the prosody-aware translations y^a, y^b . We enforce constraints to ensure that the differences between

the two target translations, y^a and y^b , arise solely from the prosodic differences in the source.

Role: You are a helpful assistant with expert knowledge in speech, prosody, linguistics and translation.

Input: English sentence x and two prosodic variations p^a, p^b corresponding to different interpretations i^a, i^b .

Task: Translate (1) x (without prosody), (2) x using p^a and i^a , and (3) x using p^b and i^b into [Target Language] as y, y^a , and y^b .

Steps:

1. Translate without using prosody x into y .
2. Translate prosody-aware cases, focusing on how the translation should change to reflect the prosody.

Constraints: [List of Constraints, e.g., “differences must be due to prosody only”]

Figure 7.3: Prompt template used for Oracle Translation (Prompt 2).

However, strictly enforcing differences can sometimes lead to unnatural translations. To mitigate this, we employ a post-editing step (Prompt 3, Figure 7.4). The LLM is presented with the generated candidates and asked to select the most appropriate translation for each prosodic case, effectively filtering out non-idiomatic or overly explanatory outputs. Finally, we explicitly filter out cases where the translations remain identical ($y^a = y^b$) or where the translation becomes overly explanatory, effectively hallucinating information not present in the source to resolve the ambiguity (Lim et al., 2024).

Role: You are a helpful assistant and an expert translator.

Input: English sentence with rich prosodic annotations and candidate translations.

Task: Select the most appropriate and prosody-aware translation.

Output: Provide a brief explanation of your reasoning followed by the index of the selected translation.

Figure 7.4: Prompt template used for Translation Post-editing (Prompt 3). This step ensures the selected translation is natural and fits the prosodic context.

7.4.3 Controllable Speech Synthesis

To convert the textual prosodic descriptions into acoustic signals (s^a, s^b), we utilize the OpenAI TTS API. While this system produces natural-sounding speech, it lacks explicit fine-grained controls for prosody. Through empirical trial-and-error, we identified a set of prompting strategies (using punctuation, capitalization, and emojis), that effectively steer the prosodic realization of the output. Table 7.2 summarizes these strategies.

For every example, we synthesize candidates using six different voice profiles. This diversity is crucial for evaluating the robustness of ST models to speaker variation.

Effect	TTS Prompting Strategy
Strong Emphasis	*WORD*
Normal Emphasis	*word*
Slight Emphasis	_word_
Pause	[pause]
Statement Intonation	Prepend “Statement:”
Question Intonation	Prepend “Question:” & Append “?”
Emotional/Polite Tone	Prepend & Append Emojis (e.g., 😊)

Table 7.2: Prompting strategies used to control the prosody of the OpenAI TTS model.

7.4.4 Quality Assessment and Filtering

Automated synthesis does not guarantee that the intended prosody is correctly rendered. To ensure the validity of our benchmark, we implement a rigorous quality assessment pipeline involving both lexical and signal-based verification. First, we verify the lexical content of the synthesized speech s using an off-the-shelf ASR model; we discard any candidate where the Word Error Rate (WER) is non-zero.

Second, we verify the prosodic content using signal processing and category-specific objectives, described in detail in the Appendix (see §D.2). For *Sentence Stress* and *Prosodic Breaks*, we employ forced alignment (Kürzinger et al., 2020) using WAV2VEC 2.0 (Baevski et al., 2020) to extract word-level duration, pitch, and energy features. We accept candidates only if the acoustic prominence (a weighted sum of loudness, pitch, and duration) aligns with the target word for stress, or if the pause duration at the target boundary is significant for breaks. For *Intonation Patterns* we predict the conditional probability of the corresponding end-of-sentence punctuation with WAV2VEC 2.0 to verify that the audio indeed entails a statement or a declarative question. For *Emotional Prosody* and *Politeness*, we utilize a speech emotion classifier trained on the RAVDESS dataset (Livingstone and Russo, 2018) to confirm that the synthesized audio matches the intended emotional label and tone.

7.5 Contrastive Evaluation Framework

Standard evaluation metrics for ST, such as BLEU (Papineni et al., 2002) and COMET (Rei et al., 2020), rely on comparing the generated hypothesis against a single reference translation. This paradigm is insufficient for evaluating prosody for two reasons. First, standard references often ignore the nuances of spoken communication and default to a “neutral” translation. Second, even if a prosody-specific reference is provided, these metrics are generally insensitive to the subtle lexical changes, such as a single word substitution or reordering, that distinguish prosodic minimal pairs. A model could produce a generic translation that scores reasonably well on BLEU against both references while failing to penalize the loss of prosodic meaning.

To address this, we adopt a contrastive evaluation approach (Sennrich, 2017). Instead of measuring raw generation quality, we measure the ability of the model to discriminate between the correct translation (which matches the source prosody) and the incorrect one (which matches the alternative prosody). We propose two complementary metrics: *Contrastive Likelihood*, which assesses the internal confidence of the model, and *Contrastive Quality*, which assesses the semantic correctness of the generated output.

7.5.1 Contrastive Likelihood

This metric evaluates whether the model assigns a higher probability to the correct translation given the specific audio input. It probes the internal representation of the model to determine if it is “aware” of the prosodic cues, regardless of whether it ultimately generates the correct translation during greedy decoding.

Let s be the input speech signal and y be a candidate translation. For an End-to-End (E2E) model with parameters θ_{E2E} , we define the likelihood score $\mathcal{L}$ as the length-normalized log-probability assigned by the model:

$$\mathcal{L}(y|s) = \frac{1}{|y|} \sum_{i=1}^{|y|} \log P(y_i | y_{<i}, s; \theta_{\text{E2E}}) \quad (7.1)$$

For cascaded systems, which consist of an ASR model and an MT model, we cannot compute the direct probability $P(y|s)$. Instead, we approximate the true likelihood by

marginalizing over the top- k ASR hypotheses $\mathcal{X} = \{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(k)}\}$. Assuming the lengths of the hypotheses are similar, and the parameters of the ASR and MT models are θ_{ASR} and θ_{MT} accordingly, the likelihood is approximated as:

$$\mathcal{L}(\mathbf{y}|\mathbf{s}) \approx \frac{1}{|\mathbf{y}|} \log \left(\sum_{j=1}^k P(\mathbf{y}|\hat{\mathbf{x}}^{(j)}; \theta_{\text{MT}}) P(\hat{\mathbf{x}}^{(j)}|\mathbf{s}; \theta_{\text{ASR}}) \right) \quad (7.2)$$

In our evaluations, we use $k = 5$.

Furthermore, simply comparing raw likelihoods can be misleading if the model has a strong prior bias towards specific translations, such as high-frequency sentences. To mitigate this, we employ a normalization strategy similar to Pointwise Mutual Information (PMI). We subtract the unconditioned likelihood of the target sentence $\mathcal{L}(\mathbf{y})$, which acts as a language modeling score:

$$\bar{\mathcal{L}}(\mathbf{y}|\mathbf{s}) = \mathcal{L}(\mathbf{y}|\mathbf{s}) - \mathcal{L}(\mathbf{y}) \quad (7.3)$$

We estimate $\mathcal{L}(\mathbf{y})$ by feeding an empty audio input to the E2E model or an empty source text to the MT model.

For a given input $\mathbf{s}^a$, which corresponds to interpretation (a) , a model is considered successful if it assigns a higher normalized score to the matching translation $\mathbf{y}^a$ than to the mismatching one $\mathbf{y}^b$:

$$\mathbb{1}_{\text{success}}(\mathbf{s}^a) = \begin{cases} 1 & \text{if } f_{\mathcal{L}}(\mathbf{y}^a|\mathbf{s}^a) > f_{\mathcal{L}}(\mathbf{y}^b|\mathbf{s}^a) \\ 0 & \text{otherwise} \end{cases} \quad (7.4)$$

We report the accuracy as the percentage of successful cases across the test set.

7.5.2 Contrastive Quality

While likelihood reveals internal model knowledge, it does not guarantee that this knowledge manifests in the final output. Due to issues like exposure bias or beam search artifacts, a model might assign high probability to the correct tokens but still generate a generic or incorrect translation. To evaluate the practical utility of the system, we propose *Contrastive Quality*.

We generate the hypothesis $\hat{\mathbf{y}}^a$ from the input $\mathbf{s}^a$ using standard decoding (e.g., beam

search). We then use a semantic evaluation metric, specifically xCOMET-XL (Rei et al., 2020), to score the hypothesis against the two reference translations. Let $\mathcal{Q}(\hat{y}, y)$ be the score reflecting the semantic similarity between the hypothesis and the reference. The evaluation criterion is:

$$\mathbb{1}_{\text{success}}(\mathbf{s}^a) = \begin{cases} 1 & \text{if } \mathcal{Q}(\hat{y}^a, \mathbf{y}^a) > \mathcal{Q}(\hat{y}^a, \mathbf{y}^b) \\ 0 & \text{otherwise} \end{cases} \quad (7.5)$$

This metric determines if the generated translation is semantically closer to the prosody-compliant reference than to the alternative.

7.5.3 Evaluation Conditions

Unlike previous works on contrastive evaluation (Sennrich, 2017; Vamvas and Sennrich, 2021; Zhou et al., 2024) that pair a single source with two or more targets, CONTRAPROST relies on double-contrastive pairs ($\mathbf{s}^a \leftrightarrow \mathbf{y}^a$ and $\mathbf{s}^b \leftrightarrow \mathbf{y}^b$). Thus, for K double-contrastive examples, we define two precise conditions for success:

- **Directional Condition ($\mathcal{C}_D$):** This measures the macro-average accuracy across all individual inputs. Following Ribeiro et al. (2020), it asks whether the model prefers the correct target more often than chance.

$$\mathcal{C}_D = \frac{1}{2K} \sum_{i=1}^K (\mathbb{1}_{\text{success}}(\mathbf{s}_i^a) + \mathbb{1}_{\text{success}}(\mathbf{s}_i^b)) \quad (7.6)$$

- **Global Condition ($\mathcal{C}_G$):** This is a stricter metric that requires the model to correctly handle *both* sides of a minimal pair simultaneously. It asks whether the model can effectively disambiguate the source based solely on prosodic changes.

$$\mathcal{C}_G = \frac{1}{K} \sum_{i=1}^K (\mathbb{1}_{\text{success}}(\mathbf{s}_i^a) \times \mathbb{1}_{\text{success}}(\mathbf{s}_i^b)) \quad (7.7)$$

The Global Condition is particularly challenging because it penalizes models that default to a single dominant translation, such as always translating “Really?” as a question regardless of intonation.

7.6 Experimental Setup

In this section, we describe the specific parameters used for the data generation pipeline, the ST models selected for evaluation, and the implementation details of the metrics.

7.6.1 Data Generation Statistics

We utilized the pipeline described in §7.4 to create the CONTRAPROST benchmark. For the text generation phase, we used GPT-4² with a temperature setting of 1.0 to maximize creativity. We conditioned the generation on 20 distinct text domains per subcategory. For each combination of subcategory and domain, we generated 10 examples (15 for intonation patterns and 20 for politeness), resulting in an initial pool of approximately 5.5K sentences across 27 subcategories.

Following the synthesis of 66K audio candidates with the OpenAI cTTS³ (6 voice profiles per sentence) and the subsequent quality assessment, we retained approximately 2.8K examples with verified prosodic content. These were translated into German (deu), Spanish (spa), and Japanese (jpn). After applying the final post-editing and filtering steps to ensure translation quality and contrastiveness, we obtained a final dataset of between 1,300 and 1,400 examples per language pair. Table 7.3 summarizes the final composition of the benchmark. The extensive data statistics, broken down by subcategories are available in Table D.5 in the Appendix.

Category	eng-deu	eng-spa	eng-jpn
Emotional Prosody	373	379	376
Sentence Stress	277	279	342
Prosodic Breaks	276	252	289
Politeness	212	193	206
Intonation Patterns	173	173	173
Total	1,311	1,294	1,387

Table 7.3: Number of double-contrastive examples for each language pair in CONTRAPROST.

²GPT-4o-2024-05-13

³TTS-1-HD, platform.openai.com/docs/api-reference/audio/

7.6.2 Speech Translation Models

To understand how different architectures handle prosody, we evaluated a diverse set of 31 model variants categorized into three paradigms: End-to-End (E2E), AED-based cascades, and CTC-based cascades.

End-to-End Models. E2E models process the input speech s directly into the target translation y without an intermediate discrete text representation. Theoretically, these models have full access to the prosodic features of the source audio. We evaluated the following systems:

- SEAMLESSM4T (Seamless Communication, 2023a; Seamless Communication, 2023b): A multitask, multimodal model. We evaluated both v1 and v2 in their Medium and Large variants.
- XLS-R (Babu et al., 2022): A multilingual model initialized with WAV2VEC 2.0 (Baevski et al., 2020) and MBART-50 (Liu et al., 2020a). We tested the 300M, 1B, and 2B parameter versions.
- ZEROSWOT: The zero-shot model introduced in Ch. 5, which aligns a WAV2VEC 2.0 encoder with an NLLB decoder (NLLB, 2024). We evaluated the Medium and Large variants.
- SALMONN (Tang et al., 2024): An audio-LLM that utilizes WHISPER (Radford et al., 2023) and BEATs (Chen et al., 2023b) as audio frontends, with the VICUNA LLM (Peng et al., 2023). We evaluated the 7B and 13B parameter versions.

AED-based Cascades. These systems consist of an Attentional Encoder-Decoder (AED) ASR model followed by an MT model. While the intermediate representation is text, AED models often generate punctuation and capitalization, which can carry prosodic information. Furthermore, they may exhibit “interpretative” behavior, generating synonyms that better fit the context. We evaluated:

- SEAMLESSM4T (Cascaded Mode): Since the model is capable of multi-tasking all text/speech tasks, here we also utilize it in a cascade setup, to first generate an ASR hypothesis, and then generate a translation.
- WHISPER $\rightarrow$ NLLB: We combined WHISPER (Medium, Large v1/v3) with NLLB (distilled-600M, distilled-1.3B, 3.3B).

CTC-based Cascades. These systems use a Connectionist Temporal Classification (CTC) ASR model. Unlike AED models, standard CTC outputs typically lack punctuation and casing, and strictly adhere to the acoustic evidence without language modeling interpretations. This setup represents a “prosody-poor” baseline. We combined WAV2VEC 2.0 (Baevski et al., 2020) or HUBERT (Hsu et al., 2021) encoders (Large, XL) with the NLLB MT models.

More details about the models and variants used in our analysis are available in §D.4 in the Appendix.

7.6.3 Metrics and Implementation

We generated hypotheses using beam search with a beam size of 5 for all models. For the estimation of the cascaded likelihood ($\mathcal{L}(y|s)$ in §7.5.1), we marginalized over the top-5 ASR hypotheses. To compute the *Contrastive Quality* score, we utilized xCOMET-XL (Rei et al., 2020), a state-of-the-art learned metric capable of reference-free quality estimation.

We report the results for both the Directional ($\mathcal{C}_D$) and Global ($\mathcal{C}_G$) conditions. To assess the statistical significance of our findings, we employed bootstrap resampling (Efron, 1979) with 10,000 resamples and a 95% confidence interval.

7.7 Results and Analysis

In this section, we present the results of our evaluation on the CONTRAPROST benchmark. We analyze the general prosody awareness of current state-of-the-art models, compare the performance of different architectures (E2E vs. Cascade), and examine the difficulty of specific prosodic categories.

Model Name	Contrastive Likelihood		Contrastive Quality		xCOMET
	Directional	Global	Directional	Global	
English → German					
SEAMLESSM4T-v2-LARGE	61.2	13.5	37.4	14.5	0.988
XLS-R-2B	59.3	4.6	31.1	7.3	0.980
ZEROSWOT-LARGE	60.6	9.7	29.2	8.7	0.990
SALMONN-13B	62.8	7.2	43.2	15.9	0.975
SEAMLESSM4T-v2-LARGE (Cascade)	60.2	12.9	31.1	10.4	0.991
WHISPER-v3-LARGE → NLLB-3.3B	60.7	5.8	23.1	5.5	0.992
HuBERT-XL → NLLB-3.3B	39.4	0.5	20.5	2.6	0.979
English → Spanish					
SEAMLESSM4T-v2-LARGE	64.9	13.4	37.9	11.0	0.982
XLS-R-2B	57.6	5.6	32.0	8.4	0.930
ZEROSWOT-LARGE	57.5	9.2	31.1	5.6	0.948
SALMONN-13B	61.3	3.6	39.6	12.3	0.967
SEAMLESSM4T-v2-LARGE (Cascade)	61.3	11.7	29.5	7.6	0.984
WHISPER-v3-LARGE → NLLB-3.3B	63.2	2.9	25.4	4.8	0.987
HuBERT-XL → NLLB-3.3B	41.8	0.2	20.8	2.4	0.968
English → Japanese					
SEAMLESSM4T-v2-LARGE	59.4	12.4	40.3	13.8	0.956
XLS-R-2B	60.0	4.6	27.4	7.0	0.950
ZEROSWOT-LARGE	58.8	7.9	23.6	7.9	0.970
SALMONN-13B	60.4	10.8	46.1	16.1	0.859
SEAMLESSM4T-v2-LARGE (Cascade)	59.4	9.1	31.0	8.7	0.961
WHISPER-v3-LARGE → NLLB-3.3B	59.8	4.9	21.5	5.3	0.960
HuBERT-XL → NLLB-3.3B	40.4	0.8	15.7	2.5	0.922
Average					
SEAMLESSM4T-v2-LARGE	61.8	13.1	38.5	13.1	0.975
XLS-R-2B	59.0	4.9	30.2	7.6	0.953
ZEROSWOT-LARGE	59.0	8.9	28.0	7.4	0.969
SALMONN-13B	61.5	7.2	43.0	14.8	0.934
SEAMLESSM4T-v2-LARGE (Cascade)	60.3	11.2	30.5	8.9	0.979
WHISPER-v3-LARGE → NLLB-3.3B	61.2	4.5	23.3	5.2	0.980
HuBERT-XL → NLLB-3.3B	40.5	0.5	19.0	2.5	0.956

Table 7.4: Contrastive Evaluation results on the CONTRAPROST benchmark for English-German, English-Spanish, and English-Japanese. We report the success rates (%) for both *Contrastive Likelihood* (internal confidence) and *Contrastive Quality* (generation correctness) under two conditions: **Directional** (accuracy on individual inputs) and **Global** (accuracy on resolving full contrastive pairs). Standard translation quality is measured with xCOMET-XL. Rows with a grey background denote cascaded architectures (ASR → MT), while white rows denote End-to-End models.

7.7.1 General Prosody Awareness

Table 7.4 presents the evaluation results for a representative selection of large models across all three language pairs (see Table D.7 in the Appendix for more models). We observe a distinct gap between the internal representations of the models and their generation capabilities.

Internal Awareness (Competence). Most ST models demonstrate a latent awareness of prosody. This is evidenced by the *Contrastive Likelihood* scores, which generally exceed the random baseline of 50%. For instance, on English-German, SALMONN achieves 62.8% and SEAMLESSM4T achieves 61.2%. One exception that stands out is the CTC-based cascade with HUBERT (scores around 40%). We analyze the architecture-related results further in §7.7.2.

Generation Performance (Execution). However, this internal awareness often fails to manifest in the final translation. The *Contrastive Quality* scores, which evaluate the actual output after beam search, are significantly lower than the likelihood scores. For example, while the XLS-R model shows a likelihood accuracy of 59.3% on eng-deu, its quality score drops to 31.1%. This discrepancy suggests a “washing out” effect: during decoding, the strong prior of the language model, which typically prefers high-frequency, neutral translations, overrides the subtler acoustic signals encoded in the source representation.

The Global Condition Challenge. The *Global Condition* ($\mathcal{C}_G$), which requires the model to correctly translate *both* sides of a prosodic minimal pair, proves extremely difficult. Success rates hover around 10–15% for the best models. This highlights that while models may occasionally resolve an ambiguity correctly by chance, they lack the robust disambiguation capabilities needed to consistently rely on prosody.

7.7.2 Architecture Analysis: E2E vs. Cascade

To systematically compare architectures, we performed a regression analysis on the Global Contrastive Quality scores of all 31 models. We utilized a mixed-effects model (Pinheiro

and Bates, 2006) to account for random effects arising from model families, such as the training data and hyperparameters. Specifically:

$$\text{score}_{ij} = \beta_0 + \beta_1 \cdot \text{size}_{ij} + \beta_2 \cdot \text{AED}_{ij} + \beta_3 \cdot \text{CTC}_{ij} + u_j + \epsilon_{ij} \quad (7.8)$$

where score_{ij} is the score of i -th model variant of the j -th model family, β_0 is the intercept, size is the log of the model size, AED and CTC are binary variables, u_j is the random effect for j -th model family, and ϵ_{ij} is a residual error term.

All the scores used for this analysis are available in Table D.7 in the Appendix. The results, visualized in Figure 7.5, yield three critical insights:

1. E2E Superiority. E2E models statistically outperform cascaded systems across all language pairs. The positive coefficient for E2E models in our regression confirms the theoretical advantage discussed in §7.1: by accessing the raw acoustic signal throughout the translation process, E2E models avoid the information bottleneck inherent in cascades.

2. The Importance of Cascade Type (AED vs. CTC). Not all cascades are equal.

- **CTC-based cascades** (e.g., HUBERT → NLLB) show a strong negative impact on performance. These ASR models typically output raw streams of characters without punctuation or casing, stripping away the surface-level proxies for prosody.
- **AED-based cascades** (e.g., SEAMLESSM4T in cascaded mode) perform significantly better. We attribute this to their “interpretative” nature; they tend to generate punctuation (commas for pauses, question marks for intonation) and select synonyms that better fit the context, effectively passing some prosodic information to the MT model via text.

3. Model Size is Not a Silver Bullet. Surprisingly, we found no significant correlation between model size and prosody awareness. Scaling up parameters improves general translation quality (as seen in standard benchmarks), but it does not automatically emerge prosodic understanding. This suggests that data composition, specifically the lack of prosody-rich training examples, is a harder constraint than model capacity.

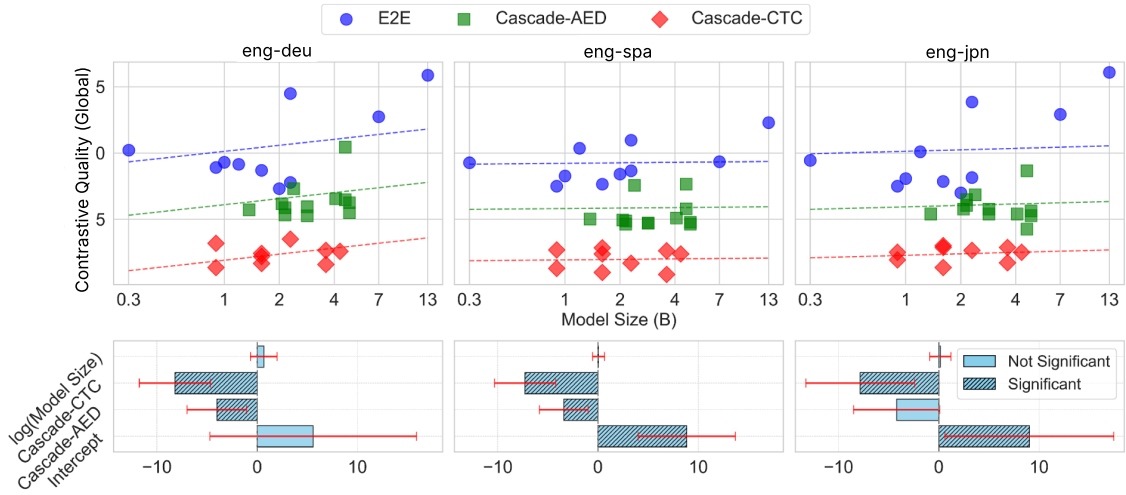


Figure 7.5: Results of the mixed-effects regression analysis modeling Global Contrastive Quality as a function of model architecture and size. **Top:** Scatter plot of Global Contrastive Quality against Model Size (log scale) for E2E models (blue circles), AED-based cascades (green squares), and CTC-based cascades (red diamonds), with fitted regression lines for each group. **Bottom:** Estimated regression coefficients with 95% confidence intervals for model size ($\log(\text{Size})$) and architecture types relative to the E2E baseline (Cascade-AED and Cascade-CTC), indicating the magnitude and significance of each factor’s impact.

7.7.3 Comparison with Standard Metrics

A key motivation for this work is the hypothesis that standard metrics do not adequately capture prosodic nuances. To verify this, we computed the Spearman rank correlation between our contrastive metrics and the standard reference-free metric, xCOMET-XL, across all 31 evaluated models.

As shown in Figure 7.6, there is a weak correlation between general translation quality, measured by xCOMET-XL, and prosody awareness, measured by Contrastive Quality. For instance, the WHISPER $\rightarrow$ NLLB cascade achieves one of the highest xCOMET scores (Table 7.4: 0.992 on eng-deu) yet performs poorly on the contrastive benchmark compared to E2E models. This discrepancy confirms that optimizing for general semantic coverage does not necessarily improve, and may even obscure, the system’s ability to leverage paralinguistic cues. It underscores the necessity of targeted benchmarks like CONTRAPROST to evaluate this specific dimension of translation.

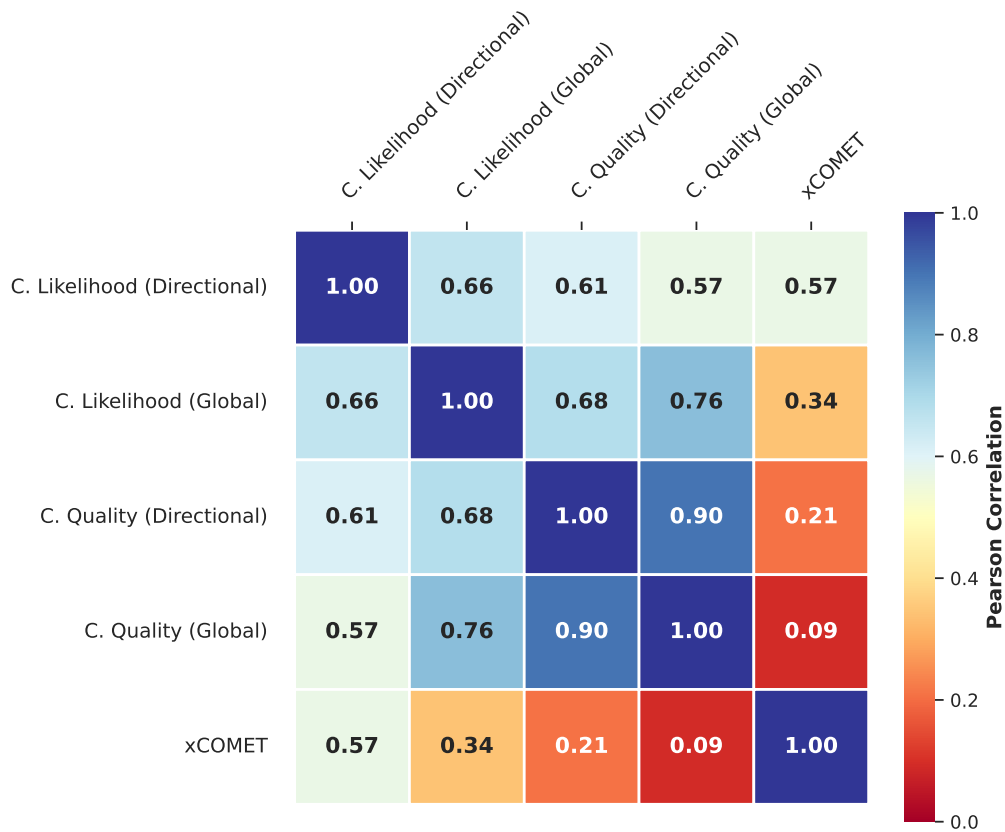


Figure 7.6: Correlation matrix comparing the proposed contrastive metrics (Likelihood and Quality in both Directional and Global conditions) with the standard reference-free metric xCOMET-XL. Values represent the Pearson correlation coefficients computed across all 31 evaluated model variants and three language pairs, illustrating the relationship between general translation quality and prosody awareness.

7.7.4 Analysis by Prosodic Category

Prosodic Categories. Figure 7.7 (Upper) breaks down model performance by specific prosodic categories for the English-German pair. The results indicate that *Intonation Patterns* is the only category models are able to solve consistently. This is likely because the distinction between a statement and a question is strongly correlated with punctuation (e.g., period vs. question mark). Since this is a standard feature in text training data, even cascaded systems perform relatively well here. In contrast, the other four categories prove much more difficult, with *Sentence Stress* being the most challenging. These categories rely on acoustic features like loudness, duration, and pitch contours that have no direct orthographic equivalent.

Model Comparisons. To rigorously compare model performance, we performed pairwise comparisons via bootstrap resampling, shown in Figure 7.7 (Lower). First, we find that the audio-LLM SALMONN is not statistically different from the standard E2E model SEAMLESSM4T. Second, comparing SEAMLESSM4T in both E2E and cascaded modes allows us to control for training data and architecture. The results show a clear advantage for the E2E configuration, providing empirical support for the theoretical benefit of direct audio access. Finally, comparing the two cascades, we observe a significant performance gain for the SEAMLESSM4T cascade over the WHISPER $\rightarrow$ NLLB system. We hypothesize that this is due to the multitasking nature of SEAMLESSM4T, which makes its ASR component more “interpretative”. Unlike standard ASR models that prioritize literal transcription, an interpretative ASR might generate synonyms or phrasing that better capture the prosodic intent (e.g., “ridiculed” instead of “laughed at”). Supporting this hypothesis, we observe a worse Word Error Rate (WER) for the SEAMLESSM4T ASR component (11%) compared to WHISPER (4%), suggesting it deviates more from the literal transcript to capture meaning.

The performance per category and model comparisons for the other language pairs are available in Figures D.1 and D.2 in the Appendix, where they confirm the findings of this section.

7.7.5 Language Dependencies

Finally, we examined whether prosody awareness varies by target language. Our regression analysis (see Figure 7.8) indicates significant differences, with models performing better on English-German compared to English-Spanish. We hypothesize that this may be linked to the expressivity or structural flexibility of the target language. Some languages may offer more lexical or syntactic tools to encode the nuances conveyed by English prosody, making the translation task inherently easier for the model.

7.8 Conclusions

In this chapter, we presented CONTRAPROST, a benchmark designed to systematically evaluate the prosody awareness of Speech Translation systems. By leveraging a semi-automated pipeline involving Large Language Models and Controllable Text-to-Speech,

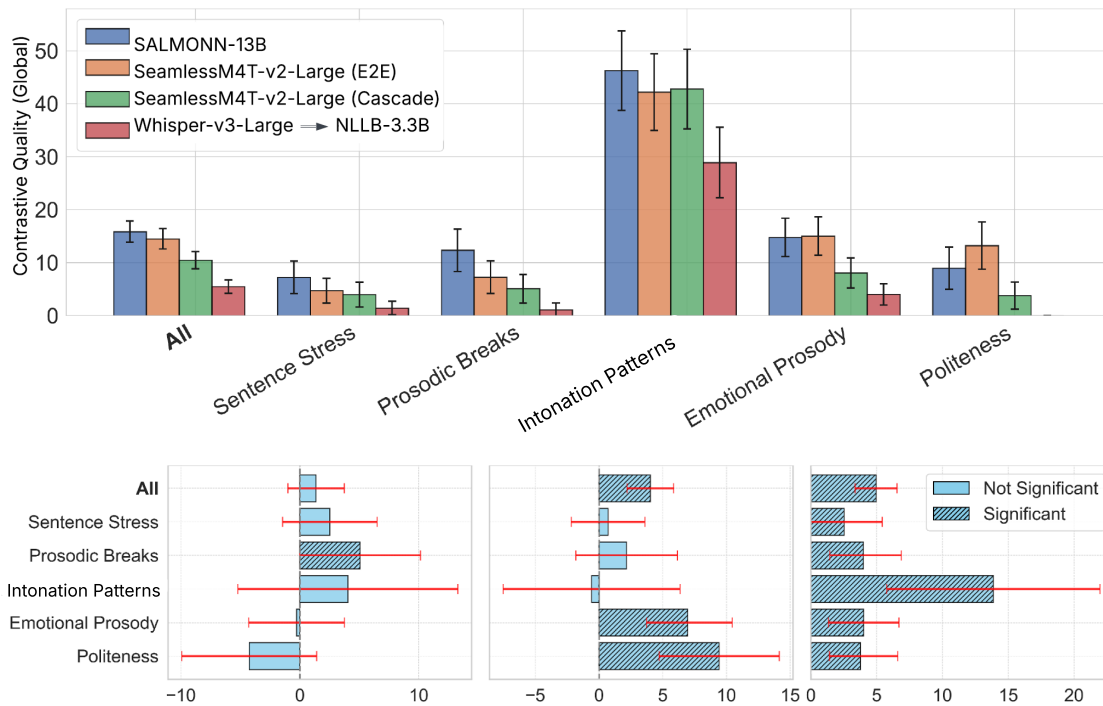


Figure 7.7: Detailed performance analysis for the English-German language pair. **Upper:** Global Contrastive Quality scores broken down by the five specific prosodic categories (Sentence Stress, Prosodic Breaks, Intonation Patterns, Emotional Prosody, Politeness) for selected high-performing models. Error bars represent 95% confidence intervals. **Lower:** Pairwise performance comparisons between key model configurations. Bars represent the difference in Global Contrastive Quality scores, with hatched patterns indicating statistically significant differences ($p < 0.05$) based on bootstrap resampling. Comparisons: (a) SALMONN-13B vs. SEAMLESSM4T-v2-LARGE, (b) SEAMLESSM4T-v2-LARGE (E2E) vs. SEAMLESSM4T-v2-LARGE (Cascade), (c) SEAMLESSM4T-v2-LARGE (Cascade) vs. WHISPER-v3-LARGE → NLLB-3.3B.

we addressed the scarcity of prosody-rich evaluation data and the insensitivity of standard metrics to paralinguistic nuances.

Summary of Conclusions. Our empirical analysis across 31 model variants and three language pairs yields several critical insights. First, we validated the theoretical hypothesis that End-to-End architectures possess an inherent advantage over cascaded systems in handling non-verbal information. E2E models consistently outperformed their cascaded counterparts on our contrastive benchmark, particularly in categories that rely on acoustic cues like stress and emotion rather than punctuation. Second, we found that while models often encode prosodic information in their internal representations (as evidenced by high Likelihood scores), this signal is frequently lost during the generation process

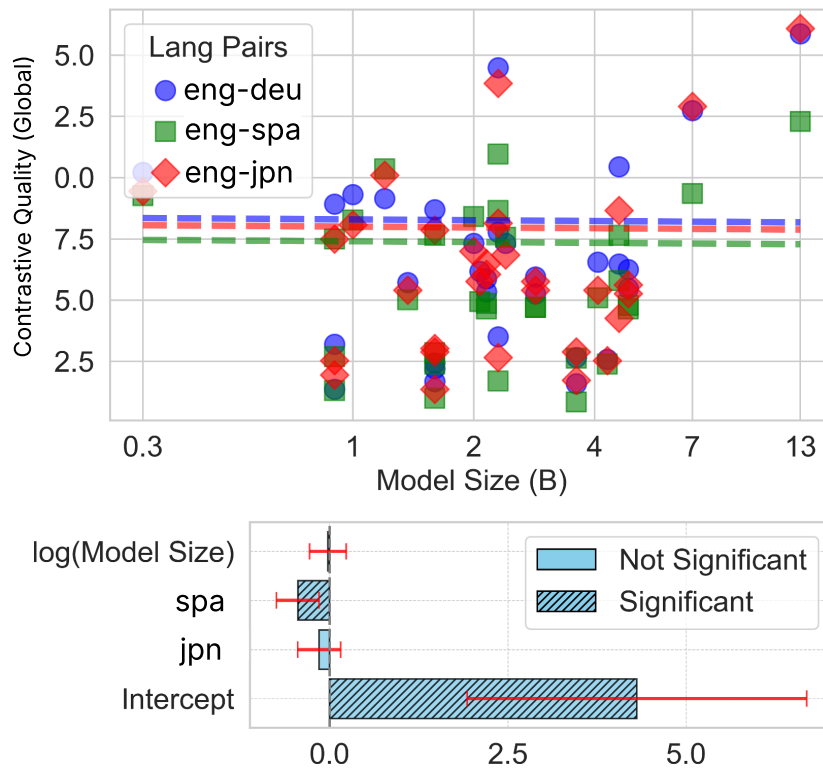


Figure 7.8: Regression analysis assessing the impact of the target language on prosody awareness. **Top:** Scatter plot of Global Contrastive Quality scores for all models, color-coded by target language (German, Spanish, Japanese). **Bottom:** Estimated regression coefficients for the target language factor, with English-German serving as the reference intercept. Bars indicate the magnitude of the performance difference for Spanish (spa) and Japanese (jpn) relative to German, with 95% confidence intervals.

(as evidenced by lower Quality scores). This suggests that the decoder’s language model prior often overrides the acoustic evidence. Third, we observed that cascaded systems can capture some prosodic information, but this is largely contingent on the “interpretative” capacity of the ASR component to encode prosody into surface features like punctuation or synonyms.

Limitations. We acknowledge certain limitations in our study. The reliance on synthetic speech, while necessary for scalability and control, may not fully capture the complexity and variability of natural spontaneous speech. Although we applied rigorous filtering, the automated generation process inevitably introduces some noise, such as instances where the synthesized prosody is not sufficiently prominent or the translations are overly explanatory. Furthermore, our contrastive evaluation focuses on the capability of the model

to resolve ambiguity in isolation. It does not quantify how frequently such ambiguity arises in open-domain translation scenarios.

Directions for future research. Despite these limitations, this work highlights a crucial missing dimension in current ST research. While previous chapters of this thesis focused on structural alignment (Ch. 3) and modality bridging (Ch. 5, Ch. 6), this chapter demonstrates that transferring the *content* is not enough; preserving the *manner* of speech is equally vital for accurate communication. The findings suggest that future work should explore explicit supervision signals, such as auxiliary prosodic losses or fine-tuning on prosody-rich data, to force models to better leverage the acoustic context they already represent. We discuss these future directions in the broader context of the thesis in the final chapter.

Part V

Closing

8 Conclusions

This dissertation investigated End-to-End (E2E) Speech Translation (ST) from the perspective of three persistent bottlenecks that limit its robustness and applicability in real-world settings: (i) the mismatch between sentence-level training data and continuous audio at inference time, (ii) the scarcity of paired speech-to-translation supervision, and (iii) the difficulty of evaluating whether E2E systems leverage information that is unique to speech, such as prosody. Across the main Parts II to IV, the thesis developed methods that (a) make E2E ST more resilient to realistic inputs, (b) reduce dependence on paired ST corpora by enabling effective cross-modal transfer from ASR and MT resources, and (c) provide targeted evaluation tools for measuring prosody awareness.

8.1 Bridging the Segmentation Gap

Part II addressed the *input processing bottleneck*: modern E2E ST models are typically trained on manually segmented, sentence-level audio, while practical deployments require translation from continuous streams. In Ch. 3, we introduced SHAS (Supervised Hybrid Audio Segmentation), motivated by the observation that acoustically defined boundaries (e.g., Voice Activity Detection) do not reliably coincide with semantically coherent translation units. SHAS uses self-supervised speech representations to train a Segmentation Frame Classifier that mimics human segmentation decisions, and a Probabilistic Divide-and-Conquer decoding algorithm that produces variable-length segments under length constraints. Empirically, SHAS substantially reduced the gap to oracle (manual) segmentation and demonstrated strong cross-lingual generalization, indicating that segmentation for translation can be learned from high-resource languages and transferred to others.

In Ch. 4, we further showed that segmentation is not only an inference-time necessity, but also an opportunity to make better use of scarce ST supervision. SEGAUGMENT turns re-segmentation into augmentation by generating multiple alternative segmentations of document-level speech under different length constraints and recovering the corresponding text via forced alignment and machine translation. This procedure increases contextual and positional diversity of training examples and, crucially, reduces the training–inference

mismatch: models trained with SEGAUGMENT can retain essentially the same translation quality when evaluated on automatic segmentation as when evaluated on manual segmentation. Furthermore, it also increases the utility of ST data, bringing overall improvements, particularly in low-resource settings. Taken together, SHAS and SEGAUGMENT establish that segmentation should be treated as a first-class component of the ST pipeline rather than a heuristic pre-processing step.

8.2 Mitigating Data Scarcity via Cross-Modal Alignment

Part III addressed the *training data bottleneck*: while ASR and MT benefit from abundant resources, paired ST data remains limited. In Ch. 5, we asked whether E2E ST is possible *without any parallel ST data* by directly connecting a pretrained speech encoder to a pretrained multilingual MT model. We proposed ZEROSWOT, a framework that bridges the modality gap through (i) Optimal Transport alignment between sequences and (ii) learned subword compression using CTC to match the granularity and length expected by the MT encoder. The resulting model significantly improves over prior zero-shot approaches and, when scaled, can reach or surpass strong supervised baselines on standard ST benchmarks. Beyond the performance gains, the study provided evidence that the combination of sequence-level OT and learned compression is an effective way to align modalities without requiring paired speech–translation examples.

In Ch. 6, we revisited a key limitation of subword-based interfaces: tokenizers and subword vocabularies impose structural rigidity that harms transfer to low-resource and unseen languages. Furthermore, acoustic representations require complex compression algorithms to be aligned with them (Ch. 5). Thus, we introduced CHARSONAR, a character-level adaptation of the SONAR embedding space, and demonstrated that moving the interface to characters simplifies cross-modal transfer from CTC-based ASR models and improves cross-lingual generalization, leading to state-of-the-art results. This chapter underscores a broader conclusion of the thesis: *the choice of representation granularity at the speech–text interface is decisive* for both modality transfer and language transfer. Character-level modeling provides a more universal, tokenizer-free bridge that is especially well matched to massively multilingual ASR encoders.

8.3 Evaluating Whether Speech Adds Meaning

Part **IV** addressed the *evaluation bottleneck*. A commonly cited advantage of E2E ST over cascades is the ability to access source-side acoustic cues throughout the translation process. However, typical ST benchmarks and metrics are largely insensitive to whether these cues influence the generated translation. In Ch. 7, we introduced CONTRAPROST, a benchmark and methodology designed to isolate the effect of prosody on meaning, using contrastive minimal pairs generated through a semi-automated pipeline with controllable TTS. The results showed a consistent pattern: while E2E systems tend to outperform cascades in prosody-sensitive settings, prosodic information is often weakly reflected in the final textual output even when it is present in internal representations. This finding highlights that access to speech is not sufficient by itself; architectures and objectives must encourage the model to *use* paralinguistic cues when they are semantically relevant.

8.4 Overall Takeaways

The thesis supports three overarching takeaways. First, robustness to real-world speech begins with robust segmentation: learned, translation-oriented segmentation can recover much of the performance lost when moving from curated sentence inputs to continuous streams, and segmentation-aware augmentation can further reduce brittleness. Second, data scarcity can be addressed by explicit cross-modal alignment that reuses powerful pretrained text models; moreover, representation choices (subwords vs. characters) strongly affect how well these methods scale to many languages. Third, progress in E2E ST should be measured not only by lexical adequacy, but also by whether models exploit uniquely acoustic information; targeted benchmarks such as CONTRAPROST are essential to make such capabilities measurable.

9 Retrospective: Speech Translation in the Era of Multimodal LLMs

The research presented in this dissertation spans a transformative period in Natural Language Processing and Artificial Intelligence. When the methodologies in this thesis were originally conceived between 2022 and 2024, the field of Speech Translation was defined by the structural dichotomy between the robust but error-prone *Cascade* paradigm and the emerging, albeit data-scarce, *End-to-End* (E2E) paradigm (Sperber and Paulik, 2020; Bentivogli et al., 2021). The central hypothesis driving this work was that E2E architectures could overcome their cascaded counterparts by solving their data scarcity and input processing bottlenecks. As we finalize this dissertation in 2026, the fundamental axioms of the field have shifted. The paradigm has evolved from *aligning* modalities, as explored in ZEROSWOT and CHARSONAR, to *interleaving* them within natively Multimodal Large Language Models (LLMs) and SpeechLLMs (Gaido et al., 2024). In this section, we critically re-examine the contributions of this thesis through the lens of this new landscape.

Input Processing: Memory Constraints and Modern Methods. In Ch. 3, we framed the “segmentation mismatch” primarily as a problem of fitting audio into the fixed context windows of standard Transformers (Vaswani et al., 2017). With the advent of foundation models capable of processing contexts exceeding one million tokens, such as Gemini (Gemini, 2025) or GPT-5 (OpenAI, 2025), one might hypothesize that the constraints necessitating SHAS would be rendered obsolete. However, empirical findings from 2026 challenge this “infinite context” assumption. Papi et al. (2025) report that SpeechLLMs (e.g., Qwen2-Audio (Chu et al., 2024)) processing continuous long-form audio without segmentation suffer from “extreme length degradation,” with performance drops exceeding 90% on document-level tasks. Consequently, explicit discretization of the acoustic signal remains critical to prevent semantic drift. Notably, SpeechLLMs capable of matching the robustness of traditional cascaded systems, like Voxtral (Liu et al., 2025), explicitly re-introduce a chunking mechanism (processing 30s segments) to handle acoustic inputs (Papi et al., 2025). Thus, rather than superseding segmentation, massive context windows have

highlighted its necessity: explicit discretization of the acoustic signal remains critical to prevent semantic drift and hallucination in long-form processing.

Advancements in 2025 have moved beyond the supervised mimicry of punctuation introduced in SHAS towards *pragmatic* segmentation policies. For instance, [Yang and Nakamura \(2025\)](#) proposed replacing the lightweight classifier of SHAS with LLMs fine-tuned via Direct Preference Optimization (DPO) ([Rafailov et al., 2024](#)). Crucially, they cite SHAS as the foundational pre-trained segmentation model but refine the objective: rather than predicting “gold” punctuation, the model learns to predict boundaries that maximize the trade-off between translation quality and latency. Furthermore, while SHAS focused on *when* to cut, recent advancements focus on *what* to separate. As a foundation model for general audio separation, SAM-AUDIO ([Shi et al., 2025](#)) unifies text, visual, and temporal prompting to isolate specific sounds within complex mixtures. This effectively upgrades the segmentation task from a pre-processing step to a semantic reasoning capability and allows systems to “focus” on a speaker or sound event in a way that static algorithms like SHAS or VAD could not.

Overall, audio segmentation remains an important part of ST pipelines, even in the presence of large SpeechLLMs, as shown by recent findings from IWSLT ([Abdulmumin et al., 2025](#)). Specifically, SHAS, proposed in 2022, remains a gold standard: it is used to segment evaluation data ([Abdulmumin et al., 2025](#)) and is employed by several teams in their systems ([Tan et al., 2025](#); [Wang et al., 2025](#)).

Data-Centric Robustness and the Long Tail. The data augmentation strategies proposed in SEGAUGMENT (Ch. 4) also require re-evaluation in the age of massive pre-training. Some of the original mechanisms we identified, such as the need for Positional Diversity, have been mitigated by architectural shifts like Rotary Positional Embeddings (RoPE) ([Su et al., 2024](#)), which allow models to extrapolate to unseen lengths more robustly than the absolute embeddings of 2023. Similarly, the Contextual Diversity provided by re-segmentation is partly addressed by the massive unsupervised pre-training of Speech Foundation Models (e.g., Whisper ([Radford et al., 2023](#)), USM ([Zhang et al., 2023c](#))). However, SEGAUGMENT has found new relevance in two critical areas. First, it addresses the “Length Adapter Bottleneck”. Models trained without robust length-adaptation strategies are prone to the degradation mentioned above. Recent work confirms that curriculum learning over diverse segment lengths, a core part of SEGAUGMENT, is essential for training robust adapters ([Du et al., 2025](#)). Second, and perhaps most importantly, the *long*

tail issue in ST persists. Despite the capabilities of flagship models, performance remains heavily skewed towards high-resource languages (Gaido et al., 2024). In low-resource scenarios where supervised data is scarce, fine-tuning large foundation models leads to rapid overfitting. Here, SEGAUGMENT can still serve as a vital regularizer, multiplying limited datasets into diverse acoustic views to ensure generalization in the long tail of speech translation.

The Importance of Modular Adaptation. Our strategy of leveraging pre-trained text models as an “anchor” for speech (Ch. 5, Ch. 6) has become the dominant architectural blueprint for the field. As detailed in recent surveys (Gaido et al., 2024), the standard SpeechLLM architecture in 2026 follows a similar pattern to the one proposed in Part III: a frozen, massive LLM backbone (in our case, a multilingual encoder) connected to a speech encoder via a lightweight adapter. While our work explored this through Optimal Transport and character-level bridging, the industry has standardized on similar modular adapters (e.g., Q-Formers (Li et al., 2023)) to bridge the “modality gap”. However, the *granularity mismatch* we identified in Ch. 6 remains a friction point. As models scale to thousands of languages, subword tokenizers are becoming a bottleneck and do not align easily with other modalities (e.g., speech), which further motivates the shift toward character-level or token-free approaches.

The “Listening vs. Reading” Paradox. In Ch. 7, we evaluated whether E2E models could preserve prosody, identifying a “competence–performance gap” where models encoded acoustic information but failed to generate it. Papi et al. (2025) reveals that SpeechLLMs struggle with a “Listening or Reading” paradox: because these models are heavily pre-trained on text, they often hallucinate fluent continuations or ignore the acoustic signal when it conflicts with strong text priors. Contrary to the 2024 hypothesis that “direct access to audio guarantees better prosody”, they find that cascaded systems remain surprisingly competitive for semantic correctness because the explicit transcription step grounds the model and reduces hallucination. This complicates our conclusion in Ch. 7; while E2E models are theoretically superior for paralinguistics, the modularity of cascades (or “soft cascades” like ZEROSWOT) offers a robustness that native SpeechLLMs are still fighting to match.

Generative Benchmarks. Finally, a retrospective on CONTRAPROST (Ch. 7) highlights a limitation not of methodology, but of tooling. Our benchmark relied on 2024-era LLM

and Text-to-Speech (TTS) generation, which introduced noise that required aggressive filtering. Today, with the generative capabilities of models like Gemini (Gemini, 2025) and highly expressive generative voice tools like ElevenLabs¹, the creation of synthetic, prosody-controlled evaluation sets has become trivial. Thus, although the methodology of CONTRAPROST (using synthetic minimal pairs to probe model behavior with respect to prosody) was correct, the benchmark could potentially be re-created on a much larger scale (language and prosody coverage) and with more faithful examples.

Summary: 2026 Status of Thesis Contributions. Table 9.1 maps each contribution of this thesis to its original context and its 2026 reassessment.

Contribution	2022–2024 Status	2026 Reassessment
SHAS (Ch. 3)	Novel semantic segmentation method; state-of-the-art on MuST-C/mTEDx	Still the gold-standard: used to segment evaluation data and by multiple teams in IWSLT 2025 (Abdulumumin et al., 2025)
SEGAUGMENT (Ch. 4)	Data augmentation via re-segmentation; closes training–inference gap	Finds renewed relevance for long-tail and low-resource settings; validates curriculum diversity for length-adapter training
ZEROSWOT (Ch. 5)	OT-based zero-shot ST; outperforms prior zero-shot methods	Blueprint adopted by SpeechLLM adapters; OT + CTC compression pattern now standard in modular speech–text alignment
CHARSONAR (Ch. 6)	Character-level modality bridging; state-of-the-art zero-shot ST	Directly motivates token-free and byte-level approaches; character interface alleviates subword bottleneck as models scale to more languages
CONTRAPROST (Ch. 7)	Benchmark for prosody preservation; reveals E2E competence–performance gap	Methodology validated; could scale to larger, higher-fidelity benchmarks with modern TTS tools

Table 9.1: 2026 Reassessment of the five main contributions of this thesis.

¹elevenlabs.io

10 Directions for Future Work

The retrospective analysis of Ch. 9 highlights that although the “engineering” bottlenecks of memory and data scarcity are being solved by scale, they are being replaced by “cognitive” bottlenecks, namely hallucination, semantic drift, and the loss of paralinguistic nuance. Based on the findings of this dissertation and the current state of the art in 2026, we propose three key directions for future research.

From Static Segmentation to Dynamic Auditory Reasoning. In Part II, we proposed SHAS to align inference segmentation with training conditions. However, as discussed in Ch. 9, the emergence of general-purpose audio foundation models such as SAM-AUDIO (Shi et al., 2025) suggests that segmentation should no longer be treated as a static pre-processing step. Instead, it should be framed as a dynamic process. Future work can explore concepts such as *Integrated Scene Analysis*, where the translation model actively queries an auxiliary reasoning module (Guo et al., 2025) to isolate relevant speakers or filter background noise based on the unfolding conversational context. Rather than feeding the model a fixed stream of 30-second chunks, the system could learn a policy, potentially via Direct Preference Optimization (DPO) (Yang and Nakamura, 2025), that decides *when* to listen, *when* to translate, and *which* acoustic source to attend to. This shifts the goal from “segmentation for robustness” to “segmentation for reasoning,” aligning with the broader trend of agentic AI.

Universal Token-Free Translation. In Ch. 6, we identified the “Subword Bottleneck” as a major impediment to zero-shot transfer. Although CHARSONAR alleviated this by moving to characters, the reliance on *any* discrete vocabulary is increasingly limiting for the long tail of languages. Given the success of our character-level alignment, the next logical step is *Token-Free Speech Translation*. Future research can investigate architectures that map continuous speech representations directly to byte-level representations (Pagnoni et al., 2025), bypassing the discrete tokenizer entirely. This approach would not only eliminate the bias of fixed vocabularies but also better align with the continuous nature of the speech signal. By removing the “modality gap” imposed by text-specific tokenizers,

we can move towards a truly universal translation engine capable of handling any written script or spoken dialect with equal native proficiency.

Overcoming the “Listening vs. Reading” Paradox via Preference Optimization. In Ch. 7, we demonstrated that E2E ST models possess the internal capacity to represent prosodic cues such as contrastive stress, yet this competence rarely manifests in the final translation output—a gap we termed the “competence–performance gap.” As further discussed in Ch. 9, the most critical open challenge arising from this finding is the tendency of SpeechLLMs to ignore acoustic nuance in favor of strong text priors, which is framed as the “Listening vs. Reading” paradox in Papi et al. (2025). To address this, future work must move beyond the standard next-token-prediction objective, which does not explicitly penalize the loss of paralinguistic information. A promising direction is to combine *Prosodic Chain-of-Thought* with DPO (Rafailov et al., 2024). We can train models to generate a “prosodic transcript” (e.g., explicit markers for tone, emotion, or emphasis) prior to translation, forcing the model to attend to the acoustic signal, similar to what Gállego et al. (2025) did with phonemes. Crucially, by applying DPO, we can optimize the model against preference pairs where “acoustically faithful” translations are ranked higher than “fluent but hallucinated” ones. This method effectively grounds the generation process in the audio, narrowing the gap between the internal competence we observed in Ch. 7 and the final translation performance.

Closing Remarks

E2E Speech Translation has matured rapidly through the use of pretrained foundation models, yet it still faces practical challenges at the boundaries between research benchmarks and real-world speech. By addressing segmentation, supervision scarcity, and prosody-sensitive evaluation in a unified framework, this dissertation contributes methods and insights that bring E2E ST closer to robust, scalable, and semantically faithful speech understanding and translation.

Bibliography

Idris Abdulmumin, Victor Agostinelli, Tanel Alumäe, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Fethi Bougares, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, William Chen, Raj Dabre, Yannick Estève, Marcello Federico, Mark Fishel, Marco Gaido, Dávid Javorský, Marek Kasztelnik, Fortuné Kponou, Mateusz Krubiński, Tsz Kin Lam, Danni Liu, Evgeny Matusov, Chandresh Kumar Maurya, John P. McCrae, Salima Mdhaffar, Yasmin Moslem, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Atul Kr. Ojha, John E. Ortega, Sara Papi, Pavel Pecina, Peter Polák, Piotr Połec, Ashwin Sankar, Beatrice Savoldi, Nivedita Sethiya, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Marco Turchi, Alex Waibel, Patrick Wilken, Rodolfo Zevallos, Vilém Zouhar, and Maike Züfle. 2025. **Findings of the IWSLT 2025 Evaluation Campaign**. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 412–481, Vienna, Austria (in-person and online). Association for Computational Linguistics.

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

P.D. Aguero, J. Adell, and A. Bonafonte. 2006. **Prosody Generation for Speech-to-Speech**

- Translation.** In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I.
- Ashkan Alinejad and Anoop Sarkar. 2020. **Effectively Pretraining a Speech Translation Decoder with Machine Translation Data.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8014–8020, Online. Association for Computational Linguistics.
- Duarte Alves, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins. 2023. **Steering Large Language Models for Machine Translation with Finetuning and In-Context Learning.** In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.
- Chantal Amrhein and Barry Haddow. 2022. **Don’t Discard Fixed-Window Audio Segmentation in Speech-to-Text Translation.** In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 203–219, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marceley 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, Changan Wang, and Shinji Watanabe. 2022. **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). Association for Computational Linguistics.
- 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, Changan Wang, and Matthew Wiesner. 2021. **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).

Association for Computational Linguistics.

Antonios Anastasopoulos and David Chiang. 2018. **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. Association for Computational Linguistics.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. **Findings of the IWSLT 2020 Evaluation Campaign**. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. **Common Voice: A Massively-Multilingual Speech Corpus**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Mikel Artetxe and Holger Schwenk. 2019. **Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond**. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. **Layer Normalization**.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2022. **XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale**. In *Interspeech 2022*, pages 2278–2282.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. **Unsupervised Speech Recognition**. In *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS ’21*, Red Hook, NY, USA. Curran Associates Inc.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. **wav2vec**

- 2.0: A Framework for Self-supervised Learning of Speech Representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019. **On Using SpecAugment for End-to-End Speech Translation**. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. **Neural Machine Translation by Jointly Learning to Align and Translate**. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA.
- Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. **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. Association for Computational Linguistics.
- Rainer Banse and Klaus R Scherer. 1996. **Acoustic Profiles in Vocal Emotion Expression**. *Journal of personality and social psychology*, 70(3):614.
- Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinesh Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Çelebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Chaghan Wang, Jeff Wang, Skyler Wang, and SEAMLESS Communication Team. 2025. **Joint Speech and Text Machine Translation for up to 100 Languages**. *Nature*, 637(8046):587–593.

- Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. **Cascade versus Direct Speech Translation: Do the Differences Still Make a Difference?** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2873–2887, Online. Association for Computational Linguistics.
- Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan. 2016. **Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation.** In *NIPS Workshop on end-to-end learning for speech and audio processing*, Barcelona, Spain.
- Dwight Bolinger. 1989. *Intonation and Its Uses*. Stanford University Press, Redwood City.
- Dwight L. Bolinger. 1961. **Contrastive Accent and Contrastive Stress.** *Language*, 37(1):83–96.
- 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. 2022. **On the Opportunities and Risks of Foundation Models.**

- Fabienne Braune and Alexander Fraser. 2010. **Improved Unsupervised Sentence Alignment for Symmetrical and Asymmetrical Parallel Corpora**. In *Coling 2010: Posters*, pages 81–89, Beijing, China. Coling 2010 Organizing Committee.
- Charles Brazier and Jean-Luc Rouas. 2024. **Conditioning LLMs with Emotion in Neural Machine Translation**. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 33–38, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. **Language Models are Few-Shot Learners**. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. **Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition**. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.
- Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. **VGGSound: A Large-scale Audio-Visual Dataset**. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725.
- Mingda Chen, Kevin Heffernan, Onur Çelebi, Alexandre Mourachko, and Holger Schwenk. 2023a. **xSIM++: An Improved Proxy to Bitext Mining Performance for Low-Resource Languages**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–109, Toronto, Canada. Association for Computational Linguistics.
- Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. 2023b. **BEATs: Audio Pre-training with Acoustic Tokenizers**. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Sirou Chen, Sakiko Yahata, Shuichiro Shimizu, Zhengdong Yang, Yihang Li, Chenhui

- Chu, and Sadao Kurohashi. 2024. **MELD-ST: An Emotion-aware Speech Translation Dataset**. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10118–10126, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.
- Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. **Revisiting Character-Based Neural Machine Translation with Capacity and Compression**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.
- Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. **Attention-based Models for Speech Recognition**. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15*, page 577–585, Cambridge, MA, USA. MIT Press.
- Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. **Qwen2-Audio Technical Report**.
- John Chung, Ece Kamar, and Saleema Amershi. 2023. **Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 575–593, Toronto, Canada. Association for Computational Linguistics.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. **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. Association for Computational Linguistics.
- Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. **w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training**. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250.
- Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. **Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation**. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady El-sahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinеш Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. 2023. **Seamless: Multilingual Expressive and Streaming Speech Translation**.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. **Unsupervised Cross-lingual Representation Learning at Scale**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. **FLEURS: FEW-Shot Learning Evaluation of Universal Representations of Speech**. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. **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. Association for Computational Linguistics.

Jonathan Culpeper. 2011. *“It’s not what you said, it’s how you said it!” Prosody and Impoliteness*, pages 57–84. De Gruyter Mouton, Berlin, New York.

Jonathan Culpeper, Derek Bousfield, and Anne Wichmann. 2003. **Impoliteness Revisited: With Special Reference to Dynamic and Prosodic Aspects**. *Journal of Pragmatics*, 35:1545–1579.

- Marco Cuturi. 2013. **Sinkhorn Distances: Lightspeed Computation of Optimal Transport**. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2292–2300, Red Hook, NY, USA. Curran Associates Inc.
- David Dale and Marta R. Costa-jussà. 2024. **BLASER 2.0: A Metric for Evaluation and Quality Estimation of Massively Multilingual Speech and Text Translation**. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16075–16085, Miami, Florida, USA. Association for Computational Linguistics.
- Maureen de Seyssel, Marvin Lavechin, Hadrien Titeux, Arthur Thomas, Gwendal Virlet, Andrea Santos Revilla, Guillaume Wisniewski, Bogdan Ludusan, and Emmanuel Dupoux. 2023. **ProsAudit, a Prosodic Benchmark for Self-supervised Speech Models**. In *Proc. INTERSPEECH 2023*, pages 2963–2967.
- Nicole Dehé. 2014. **Parentheticals in Spoken English : The Syntax-Prosody Relation**. Cambridge [u.a.] : Cambridge University Press.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **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. Association for Computational Linguistics.
- Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. **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, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mattia A. Di Gangi, Matteo Negri, Viet Nhat Nguyen, Amirhossein Tebbifakhr, and Marco Turchi. 2019b. **Data Augmentation for End-to-End Speech Translation: FBK@IWSLT '19**. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.
- Tu Anh Dinh, Danni Liu, and Jan Niehues. 2022. **Tackling Data Scarcity in Speech Translation Using Zero-Shot Multilingual Machine Translation Techniques**. In *ICASSP*

- 2022 - 2022 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6222–6226.
- Quoc Truong Do, Sakriani Sakti, and Satoshi Nakamura. 2017. **Toward Expressive Speech Translation: A Unified Sequence-to-Sequence LSTMs Approach for Translating Words and Emphasis**. In *Proc. Interspeech 2017*, pages 2640–2644.
- Linhao Dong, Shuang Xu, and Bo Xu. 2018. **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.
- Yexing Du, Youcheng Pan, Ziyang Ma, Bo Yang, Yifan Yang, Keqi Deng, Xie Chen, Yang Xiang, Ming Liu, and Bing Qin. 2025. **Making LLMs Better Many-to-Many Speech-to-Text Translators with Curriculum Learning**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12466–12478, Vienna, Austria. Association for Computational Linguistics.
- Paul-Ambroise Duquenne, Hongyu Gong, Benoît Sagot, and Holger Schwenk. 2022. **T-Modules: Translation Modules for Zero-Shot Cross-Modal Machine Translation**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5794–5806, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. **SONAR: Sentence-Level Multimodal and Language-Agnostic Representations**.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. **A Simple, Fast, and Effective Reparameterization of IBM Model 2**. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.
- Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024. **Are Character-level Translations Worth the Wait? Comparing ByT5 and mT5 for Machine Translation**. *Transactions of the Association for Computational Linguistics*, 12:392–410.
- B. Efron. 1979. **Bootstrap Methods: Another Look at the Jackknife**. *The Annals of Statistics*, 7(1):1 – 26.

- Paul Ekman and Wallace V Friesen. 1971. **Constants across Cultures in the Face and Emotion**. *Journal of personality and social psychology*, 17(2):124.
- Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021a. **Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online. Association for Computational Linguistics.
- Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Carlos Segura. 2021b. **Enabling Zero-Shot Multilingual Spoken Language Translation with Language-Specific Encoders and Decoders**. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 694–701.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. **Beyond English-Centric Multilingual Machine Translation**. *J. Mach. Learn. Res.*, 22(1).
- Qingkai Fang and Yang Feng. 2023. **Understanding and Bridging the Modality Gap for Speech Translation**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15864–15881, Toronto, Canada. Association for Computational Linguistics.
- Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. **STEMM: Self-learning with Speech-text Manifold Mixup for Speech Translation**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland. Association for Computational Linguistics.
- Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. **Are LLMs Breaking MT Metrics? Results of the WMT24 Metrics Shared Task**. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.
- Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso

- Poggio. 2015. **Learning with a Wasserstein Loss**. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2053–2061, Cambridge, MA, USA. MIT Press.
- D. B. Fry. 1955. **Duration and Intensity as Physical Correlates of Linguistic Stress**. *The Journal of the Acoustical Society of America*, 27(4):765–768.
- Peyré Gabriel and Cuturi Marco. 2019. **Computational Optimal Transport with Applications to Data Sciences**. *Foundations and Trends in Machine Learning*, 11(5-6):355–607.
- Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021a. **CTC-based Compression for Direct Speech Translation**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 690–696, Online. Association for Computational Linguistics.
- Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020a. **End-to-End Speech-Translation with Knowledge Distillation: FBK@IWSLT2020**. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online. Association for Computational Linguistics.
- Marco Gaido, Mattia A. Di Gangi, Matteo Negri, Mauro Cettolo, and Marco Turchi. 2020b. **Contextualized Translation of Automatically Segmented Speech**. In *Interspeech 2020*, pages 1471–1475.
- Marco Gaido, Matteo Negri, Mauro Cettolo, and Marco Turchi. 2021b. **Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation**. In *Proceedings of the 4th International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 55–62, Trento, Italy. Association for Computational Linguistics.
- Marco Gaido, Sara Papi, Matteo Negri, and Luisa Bentivogli. 2024. **Speech Translation with Speech Foundation Models and Large Language Models: What is There and What is Missing?** In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14760–14778, Bangkok, Thailand. Association for Computational Linguistics.
- Gerard I. Gállego, Ioannis Tsiamas, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2021. **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). Association for Computational Linguistics.
- Pengzhi Gao, Liwen Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2023. **Improving Zero-shot Multilingual Neural Machine Translation by Leveraging Cross-lingual Consistency Regularization**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12103–12119, Toronto, Canada. Association for Computational Linguistics.
- Pengzhi Gao, Ruiqing Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2024. **An empirical study of consistency regularization for end-to-end speech-to-text translation**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 242–256, Mexico City, Mexico. Association for Computational Linguistics.
- Gemini. 2025. **Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities**.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. **The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation**. *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. **Unsupervised Alignment of Embeddings with Wasserstein Procrustes**. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1880–1890. PMLR.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. **Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks**. In *Proceedings of the 23rd International Conference on Machine Learning, ICML ’06*, page 369–376, New York, NY, USA. Association for Computing Machinery.
- Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. **xCOMET: Transparent Machine Translation Evaluation through Fine-grained Error Detection**.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. **Conformer: Convolution-augmented Transformer for Speech Recognition**. In *Interspeech 2020*, pages 5036–5040.

Christine Gunlogson. 2002. **Declarative Questions**. In *Proceedings of Semantics and Linguistic Theory (SALT) XII*, pages 124–143, Ithaca, NY. CLC Publications.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. **DeepSeek-R1 Incentivizes Reasoning in LLMs through**

- Reinforcement Learning.** *Nature*, 645(8081):633–638.
- Gerard I. Gállego, Oriol Pareras, Martí Cortada Garcia, Lucas Takanori, and Javier Hernandez. 2025. **Speech-to-Text Translation with Phoneme-Augmented CoT: Enhancing Cross-Lingual Transfer in Low-Resource Scenarios.** In *Interspeech 2025*, page 31–35. ISCA.
- Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. **Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder.** In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.
- M. A. K. Halliday. 1967. **Notes on Transitivity and Theme in English. Part 1 and 2.** *Journal of Linguistics*, 3:199–244.
- Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. **Learning Shared Semantic Space for Speech-to-Text Translation.** In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online. Association for Computational Linguistics.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. **Deep Residual Learning for Image Recognition.** In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Dan Hendrycks and Kevin Gimpel. 2023. **Gaussian Error Linear Units (GELUs).**
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. **Distilling the Knowledge in a Neural Network.**
- Julia Hirschberg. 2017. **Pragmatics and Prosody (Chapter 28).** In *The Oxford Handbook of Pragmatics*. Oxford University Press.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. **Long Short-Term Memory.** *Neural Comput.*, 9(8):1735–1780.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Larousilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. **Parameter-Efficient Transfer Learning for NLP.** In *Proceedings of the 36th International Conference on*

- Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. **HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. **LoRA: Low-Rank Adaptation of Large Language Models**. In *International Conference on Learning Representations*.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. **Large Language Models Can Self-Improve**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- Zhichao Huang, Rong Ye, Tom Ko, Qianqian Dong, Shanbo Cheng, Mingxuan Wang, and Hang Li. 2023b. **Speech Translation with Large Language Models: An Industrial Practice**. *arXiv preprint arXiv:2312.13585*.
- Hirofumi Inaguma, Sravya Popuri, Ilia Kulikov, Peng-Jen Chen, Changan Wang, Yu-An Chung, Yun Tang, Ann Lee, Shinji Watanabe, and Juan Pino. 2023. **UnitY: Two-pass Direct Speech-to-speech Translation with Discrete Units**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15655–15680, Toronto, Canada. Association for Computational Linguistics.
- Sathish Indurthi, Mohd Abbas Zaidi, Nikhil Kumar Lakumarapu, Beomseok Lee, Hyojung Han, Seokchan Ahn, Sangha Kim, Chanwoo Kim, and Inchul Hwang. 2021. **Task Aware Multi-Task Learning for Speech to Text Tasks**. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7723–7727.
- Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. **Europarl-ST: A Multilingual Corpus for Speech Translation of Parliamentary Debates**. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

- Ray Jackendoff. 1972. *Semantic Interpretation in Generative Grammar*. MIT Press, Cambridge, Massachusetts.
- Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. **Perceiver: General Perception with Iterative Attention**. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR.
- Wouter Jansen, Michelle L. Gregory, and Jason M. Brenier. 2001. **Prosodic Correlates of Directly Reported Speech: Evidence from Conversational Speech**. In *Proc. ITRW on Prosody in Speech Recognition and Understanding*, page paper 14.
- F. Jelinek. 1976. **Continuous Speech Recognition by Statistical Methods**. *Proceedings of the IEEE*, 64(4):532–556.
- Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. **Leveraging Weakly Supervised Data to Improve End-to-end Speech-to-text Translation**. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184.
- 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. 2017. **Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation**. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. **MetricX-24: The Google Submission to the WMT 2024 Metrics Shared Task**. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.
- J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. **Libri-Light: A Benchmark for ASR with Limited or No Supervision**. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673.
- Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel

- Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2021. **Data Augmenting Contrastive Learning of Speech Representations in the Time Domain**. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 215–222.
- G. Klewitz and E. Couper-Kuhlen. 1999. **Quote-unquote? The Role of Prosody in the Contextualization of Reported Speech Sequences**. *Pragmatics*, 9(4):459–485.
- Philipp Koehn. 2004. **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. Association for Computational Linguistics.
- K.J. Kohler. 1991. **Prosody in Speech Synthesis: The Interplay between Basic Research and TTS Application**. *Journal of Phonetics*, 19(1):121–138. Speech Synthesis and Phonetics.
- Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. **Sentence Concatenation Approach to Data Augmentation for Neural Machine Translation**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149, Online. Association for Computational Linguistics.
- Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. **HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis**. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, Red Hook, NY, USA. Curran Associates Inc.
- Taku Kudo and John Richardson. 2018. **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. Association for Computational Linguistics.
- Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. **Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution**. In *International Conference on Learning Representations*.
- Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. **CTC-Segmentation of Large Corpora for German End-to-End Speech Recognition**. In *Speech and Computer*, pages 267–278, Cham. Springer International Publishing.

- J.D.R. Ladd. 1980. *The Structure of Intonational Meaning: Evidence from English*. Indiana University Press, Bloomington.
- Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2022. **Sample, Translate, Recombine: Leveraging Audio Alignments for Data Augmentation in End-to-end Speech Translation**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 245–254, Dublin, Ireland. Association for Computational Linguistics.
- Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2023. **Make More of Your Data: Minimal Effort Data Augmentation for Automatic Speech Recognition and Translation**. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Pauline Larrouy-Maestri, David Poeppel, and Marc D. Pell. 2024. **The Sound of Emotional Prosody: Nearly 3 Decades of Research and Future Directions**. *Perspectives on Psychological Science*, 0(0):17456916231217722. PMID: 38232303.
- Phuong-Hang Le, Hongyu Gong, Changan Wang, Juan Pino, Benjamin Lecouteux, and Didier Schwab. 2023. **Pre-training for Speech Translation: CTC meets Optimal Transport**. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. **Fully Character-Level Neural Machine Translation without Explicit Segmentation**. *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. **BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models**. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Xian Li, Changan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. **Multilingual Speech Translation from Efficient Finetuning of Pretrained Models**. 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 827–838, Online. Association for Computational Linguistics.

- Mark Y. Liberman and Richard Sproat. 1992. **The Stress and Structure of Modified Noun Phrases in English**. In *Lexical Matters*.
- Jindřich Libovický, Helmut Schmid, and Alexander Fraser. 2022. **Why don't people use Character-level Machine Translation?** In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2470–2485, Dublin, Ireland. Association for Computational Linguistics.
- Zheng Wei Lim, Ekaterina Vylomova, Trevor Cohn, and Charles Kemp. 2024. **Simpson's Paradox and the Accuracy-Fluency Tradeoff in Translation**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–103, Bangkok, Thailand. Association for Computational Linguistics.
- Alexander H. Liu, Andy Ehrenberg, Andy Lo, Clément Denoix, Corentin Barreau, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, Sanchit Gandhi, Soham Ghosh, Srijan Mishra, Thomas Foubert, Abhinav Rastogi, Adam Yang, Albert Q. Jiang, Alexandre Sablayrolles, Amélie Héliou, Amélie Martin, Anmol Agarwal, Antoine Roux, Arthur Darcet, Arthur Mensch, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Chris Bamford, Christian Wallenwein, Christophe Renaudin, Clémence Lanfranchi, Darius Dabert, Devendra Singh Chaplot, Devon Mizelle, Diego de las Casas, Elliot Chane-Sane, Emilien Fugier, Emma Bou Hanna, Gabrielle Berrada, Gauthier Delerce, Gauthier Guinet, Georgii Novikov, Guillaume Martin, Himanshu Jaju, Jan Ludziewski, Jason Rute, Jean-Hadrien Chabran, Jessica Chudnovsky, Joachim Studnia, Joep Barmentlo, Jonas Amar, Josselin Somerville Roberts, Julien Denize, Karan Saxena, Karmesh Yadav, Kartik Khandelwal, Kush Jain, Lélío Renard Lavaud, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Marie Pellat, Mathilde Guillaumin, Mathis Felardos, Matthieu Dinot, Maxime Darrin, Maximilian Augustin, Mickaël Seznec, Neha Gupta, Nikhil Raghuraman, Olivier Duchenne, Patricia Wang, Patryk Saffer, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Rémi Delacourt, Romain Sauvestre, Roman Soletskyi, Sagar Vaze, Sandeep Subramanian, Saurabh Garg, Shashwat Dalal, Siddharth Gandhi, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Thibault Schueller, Thibaut Lavril, Thomas Robert, Thomas Wang, Timothée Lacroix, Tom Bewley, Valeriia Nemychnikova, Victor Paltz, Virgile Richard, Wen-Ding Li, William Marshall, Xuanyu Zhang, Yihan Wan, and Yunhao Tang. 2025. **Voxtral**.
- Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021. **Improving**

- Zero-Shot Translation by Disentangling Positional Information.** 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 1259–1273, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020a. **Multilingual Denoising Pre-training for Neural Machine Translation.** *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. **End-to-End Speech Translation with Knowledge Distillation.** In *Proc. Interspeech 2019*, pages 1128–1132.
- Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020b. **Bridging the Modality Gap for Speech-to-Text Translation.**
- Steven R. Livingstone and Frank A. Russo. 2018. **The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English.** *PLOS ONE*, 13(5):1–35.
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled Weight Decay Regularization.** In *International Conference on Learning Representations*.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. **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. Association for Computational Linguistics.
- L. Mathias and W. Byrne. 2006. **Statistical Phrase-Based Speech Translation.** In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020a. **Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics.** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020b. **Tangled up in BLEU: Reeval-**

- uating the Evaluation of Automatic Machine Translation Evaluation Metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.
- Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. **Evaluating Machine Translation Output with Automatic Sentence Segmentation**. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.
- Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. **Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi**. In *Interspeech 2017*, pages 498–502.
- Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. 2021. **Audio-Visual Instance Discrimination with Cross-Modal Agreement**. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12470–12481.
- Marina Nespør and Irene Vogel. 1986. **Prosodic Phonology**. *Phonology*, 5(1):161–168.
- H. Ney. 1999. **Speech Translation: Coupling of Recognition and Translation**. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 1, pages 517–520 vol.1.
- Thai-Son Nguyen, Sebastian Stüker, Jan Niehues, and Alex Waibel. 2020. **Improving Sequence-To-Sequence Speech Recognition Training with On-The-Fly Data Augmentation**. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7689–7693.
- Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. **Data Augmentation by Concatenation for Low-Resource Translation: A Mystery and a Solution**. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 287–293, Bangkok, Thailand (online). Association for Computational Linguistics.
- NLLB. 2024. **Scaling Neural Machine Translation to 200 Languages**. *Nature*, 630:841–846.
- OpenAI. 2025. **OpenAI GPT-5 System Card**.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal

Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rameev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry,

- Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. **GPT-4 Technical Report**.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. **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. Association for Computational Linguistics.
- Siqi Ouyang, Rong Ye, and Lei Li. 2023. **WACO: Word-Aligned Contrastive Learning for Speech Translation**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3891–3907, Toronto, Canada. Association for Computational Linguistics.
- Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. **Byte Latent Transformer: Patches Scale Better Than Tokens**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.
- Giridhar Pamisetty and K. Sri Rama Murty. 2022. **Prosody-TTS: An End-to-End Speech Synthesis System with Prosody Control**. *Circuits Syst. Signal Process.*, 42(1):361–384.
- Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. **Librispeech: An ASR Corpus Based on Public Domain Audio Books**. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

- Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021. **Dealing with Training and Test Segmentation Mismatch: FBK@IWSLT2021**. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 84–91, Bangkok, Thailand (online). Association for Computational Linguistics.
- Sara Papi, Javier Garcia Gilabert, Zachary Hopton, Vilém Zouhar, Carlos Escolano, Gerard I. Gállego, Jorge Iranzo-Sánchez, Ahrii Kim, Dominik Macháček, Patricia Schmidtova, and Maike Züfle. 2025. **Hearing to Translate: The Effectiveness of Speech Modality Integration into LLMs**.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. **Bleu: a Method for Automatic Evaluation of Machine Translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition**. In *Interspeech 2019*, pages 2613–2617.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. **Instruction Tuning with GPT-4**.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. **MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Jose Pinheiro and Douglas M. Bates. 2006. **Mixed-effects Models in S and S-PLUS**. Statistics and Computing. Springer Science & Business Media, New York.
- Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D. McCarthy, and Deepak Gopinath. 2019. **Harnessing Indirect Training Data for End-to-End Automatic Speech Translation: Tricks of the Trade**. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.
- Maja Popović. 2015. **chrF: Character N-gram F-score for Automatic MT Evaluation**. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

- Maja Popović. 2017. **chrF++: Words Helping Character N-grams**. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Post. 2018. **A Call for Clarity in Reporting BLEU Scores**. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Tomasz Potapczyk and Pawel Przybyasz. 2020. **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. Association for Computational Linguistics.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. **Scaling Speech Technology to 1,000+ Languages**.
- Patti Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. 1991. **The Use of Prosody in Syntactic Disambiguation**. In *Proceedings of the Workshop on Speech and Natural Language*, HLT ’91, page 372–377, USA. Association for Computational Linguistics.
- Joel Pynte. 1996. **Prosodic Breaks and Attachment Decisions in Sentence Parsing**. *Language and Cognitive Processes*, 11(1-2):165–192.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. **Robust Speech Recognition via Large-scale Weak Supervision**. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. **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. Association for Computational Linguistics.

- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. **Beyond Accuracy: Behavioral Testing of NLP Models with CheckList**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. **Chat-GPT MT: Competitive for High- (but Not Low-) Resource Languages**. In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. **Learning Representations by Back-propagating Errors**. *Nature*, 323(6088):533–536.
- Hiroaki Sakoe and Seibi Chiba. 1978. **Dynamic Programming Algorithm Optimization for Spoken Word Recognition**. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.
- Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. 2021. **The Multilingual TEDx Corpus for Speech Recognition and Translation**. In *Interspeech 2021*, pages 3655–3659.
- Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, Christopher Klaiber, Pengwei Li, Daniel Licht, Jean Maillard, Alice Rakotoarison, Kaushik Ram Sadagopan, Guillaume Wenzek, Ethan Ye, Bapi Akula, Peng-Jen Chen, Naji El Hachem, Brian Ellis, Gabriel Mejia Gonzalez, Justin Haaheim, Prangthip Hansanti, Russ Howes, Bernie Huang, Min-Jae Hwang, Hirofumi Inaguma, Somya Jain, Elahe Kalbassi, Amanda Kallet, Ilia Kulikov, Janice Lam, Daniel Li, Xutai Ma, Ruslan Mavlyutov, Benjamin Peloquin, Mohamed Ramadan, Abinеш Ramakrishnan, Anna Sun, Kevin Tran, Tuan Tran, Igor Tufanov, Vish Vogeti, Carleigh Wood, Yilin Yang, Bokai Yu, Pierre Andrews, Can Balioglu, Marta R. Costa-jussà, Onur Celebi, Maha Elbayad, Cynthia Gao, Francisco Guzmán, Justine Kao, Ann Lee, Alexandre Mourachko, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Paden Tomasello, Chaghan Wang, Jeff Wang, and Skyler Wang. 2023. **SeamlessM4T: Massively Multilingual & Multimodal Machine Translation**.
- Rico Sennrich. 2017. **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. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **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. Association for Computational Linguistics.
- Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, Christoph Feichtenhofer, Piotr Dollár, Wei-Ning Hsu, and Ann Lee. 2025. **SAM Audio: Segment Anything in Audio**.
- Koustuv Sinha, Amirhossein Kazemnejad, Siva Reddy, Joelle Pineau, Dieuwke Hupkes, and Adina Williams. 2022. **The Curious Case of Absolute Position Embeddings**. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4449–4472, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A. Saurous. 2018. **Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron**. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4693–4702. PMLR.
- Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. 1999. **A Statistical Model-based Voice Activity Detection**. *IEEE Signal Processing Letters*, 6(1):1–3.
- Matthias Sperber, Ondřej Bojar, Barry Haddow, Dávid Javorský, Xutai Ma, Matteo Negri, Jan Niehues, Peter Polák, Elizabeth Salesky, Katsuhito Sudoh, and Marco Turchi. 2024. **Evaluating the IWSLT2023 Speech Translation Tasks: Human Annotations, Automatic Metrics, and Segmentation**. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6484–6495, Torino, Italia. ELRA and ICCL.
- Matthias Sperber and Matthias Paulik. 2020. **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. Association for Computational Linguistics.

- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. **RoFormer: Enhanced transformer with Rotary Position Embedding**. *Neurocomput.*, 568(C).
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. **Sequence to Sequence Learning with Neural Networks**. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.
- Haotian Tan, Ruhiyah Faradishi Widiaputri, Jan Meyer Saragih, Yuka Ko, Katsuhito Sudoh, Satoshi Nakamura, and Sakriani Sakti. 2025. **NAIST Simultaneous Speech Translation System for IWSLT 2025**. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 369–378, Vienna, Austria (in-person and online). Association for Computational Linguistics.
- Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. 2024. **SALMONN: Towards Generic Hearing Abilities for Large Language Models**. In *The Twelfth International Conference on Learning Representations*.
- Yun Tang, Hongyu Gong, Ning Dong, Changan Wang, Wei-Ning Hsu, Jiatao Gu, Alexei Baevski, Xian Li, Abdelrahman Mohamed, Michael Auli, and Juan Pino. 2022. **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. Association for Computational Linguistics.
- Yun Tang, Juan Pino, Xian Li, Changan Wang, and Dmitriy Genzel. 2021a. **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. Association for Computational Linguistics.
- Yun Tang, Juan Pino, Changan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. **A General Multi-Task Learning Framework to Leverage Text Data for Speech to Text Tasks**. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.
- Jörg Tiedemann and Santhosh Thottingal. 2020. **OPUS-MT – Building open translation**

- services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.
- Alexandra Torresquintero, Tian Huey Teh, Christopher G.R. Wallis, Marlene Staib, Devang S. Ram Mohan, Vivian Hu, Lorenzo Foglianti, Jiameng Gao, and Simon King. 2021. *ADEPT: A Dataset for Evaluating Prosody Transfer*. In *Proc. Interspeech 2021*, pages 3880–3884.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. *Llama 2: Open Foundation and Fine-Tuned Chat Models*.
- Ioannis Tsiamas, David Dale, and Marta R. Costa-jussà. 2025a. *Improving Language and Modality Transfer in Translation by Character-level Modeling*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20171–20187, Vienna, Austria. Association for Computational Linguistics.
- Ioannis Tsiamas, José A. R. Fonollosa, and Marta R. Costa-jussà. 2023a. *SegAugment: Maximizing the Utility of Speech Translation Data with Segmentation-based Augmentations*. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8569–8588, Singapore. Association for Computational Linguistics.
- Ioannis Tsiamas, Gerard I. Gállego, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022b. *Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation: UPC at IWSLT 2022*. In *Proceedings of the 19th International*

- Conference on Spoken Language Translation (IWSLT 2022)*, pages 265–276, Dublin, Ireland (in-person and online). Association for Computational Linguistics.
- Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022a. **SHAS: Approaching Optimal Segmentation for End-to-End Speech Translation**. In *Interspeech 2022*, pages 106–110.
- Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2023b. **Speech Translation with Foundation Models and Optimal Transport: UPC at IWSLT23**. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 397–410, Toronto, Canada (in-person and online). Association for Computational Linguistics.
- Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2024a. **Pushing the Limits of Zero-shot End-to-End Speech Translation**. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14245–14267, Bangkok, Thailand. Association for Computational Linguistics.
- Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2023c. **Efficient Speech Translation with Dynamic Latent Perceivers**. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Ioannis Tsiamas, Santiago Pascual, Chunghsin Yeh, and Joan Serrà. 2025b. **Sequential Contrastive Audio-Visual Learning**. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Ioannis Tsiamas, Matthias Sperber, Andrew Finch, and Sarthak Garg. 2024b. **Speech Is More than Words: Do Speech-to-Text Translation Systems Leverage Prosody?** In *Proceedings of the Ninth Conference on Machine Translation*, pages 1235–1257, Miami, Florida, USA. Association for Computational Linguistics.
- Jannis Vamvas and Rico Sennrich. 2021. **On the Limits of Minimal Pairs in Contrastive Evaluation**. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 58–68, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention Is All You Need**. In

- Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. **Prompting PaLM for Translation: Assessing Strategies and Performance**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.
- T. K. Vintsyuk. 1968. **Speech Discrimination by Dynamic Programming**. *Cybernetics*, 4(1):52–57.
- Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. **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. Association for Computational Linguistics.
- Michael Wagner. 2020. **Prosodic Focus**. In Daniel Gutzmann, Lisa Matthewson, Cecilia Meier, Hotze Rullmann, and Thomas E. Zimmermann, editors, *The Wiley Blackwell Companion to Semantics*. Wiley–Blackwell.
- Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020a. **CoVoST: A Diverse Multilingual Speech-To-Text Translation Corpus**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.
- Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020b. **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. Association for Computational Linguistics.
- Changhan Wang, Anne Wu, and Juan Pino. 2020c. **CoVoST 2 and Massively Multilingual Speech-to-Text Translation**.
- Chen Wang, Yuchen Liu, Boxing Chen, Jiajun Zhang, Wei Luo, Zhongqiang Huang, and Chengqing Zong. 2022. **Discrete Cross-Modal Alignment Enables Zero-Shot Speech Translation**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural*

- Language Processing*, pages 5291–5302, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Wenxuan Wang, Yingxin Zhang, Yifan Jin, Binbin Du, and Yuke Li. 2025. **NYA’s Offline Speech Translation System for IWSLT 2025**. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 206–211, Vienna, Austria (in-person and online). Association for Computational Linguistics.
- Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. **Sequence-to-Sequence Models Can Directly Translate Foreign Speech**. In *Interspeech 2017*, pages 2625–2629.
- Rachel Wicks and Kevin Duh. 2022. **The Effects of Language Token Prefixing for Multilingual Machine Translation**. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 148–153, Online only. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. **Transformers: State-of-the-Art Natural Language Processing**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Xueqing Wu, Yingce Xia, Jinhua Zhu, Lijun Wu, Shufang Xie, Yang Fan, and Tao Qin. 2021. **mixSeq: A Simple Data Augmentation Method for Neural Machine Translation**. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 192–197, Bangkok, Thailand (online). Association for Computational Linguistics.
- Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021a. **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. Association for Computational Linguistics.

- Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021b. **Self-Training and Pre-Training are Complementary for Speech Recognition**. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034.
- Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. **ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models**. *Transactions of the Association for Computational Linguistics*, 10:291–306.
- Jichen Yang, Kai Fan, Minpeng Liao, Boxing Chen, and Zhongqiang Huang. 2023. **Towards Zero-shot Learning for End-to-end Cross-modal Translation Models**. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13078–13087, Singapore. Association for Computational Linguistics.
- Yifan Yang, Zhesu Song, Jianheng Zhuo, Mingyu Cui, Jinpeng Li, Bo Yang, Yexing Du, Ziyang Ma, Xunying Liu, Ziyuan Wang, Ke Li, Shuai Fan, Kai Yu, Wei-Qiang Zhang, Guoguo Chen, and Xie Chen. 2025. **GigaSpeech 2: An Evolving, Large-Scale and Multi-domain ASR Corpus for Low-Resource Languages with Automated Crawling, Transcription and Refinement**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2673–2686, Vienna, Austria. Association for Computational Linguistics.
- Zeyu Yang and Satoshi Nakamura. 2025. **DPO-Tuned Large Language Models for Segmentation in Simultaneous Speech Translation**.
- Rong Ye, Mingxuan Wang, and Lei Li. 2021. **End-to-End Speech Translation via Cross-Modal Progressive Training**. In *Proc. Interspeech 2021*, pages 2267–2271.
- Rong Ye, Mingxuan Wang, and Lei Li. 2022. **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. Association for Computational Linguistics.
- Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. **Prompting Large Language Model for Machine Translation: A Case Study**. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.

- Biao Zhang, Barry Haddow, and Rico Sennrich. 2022a. [Revisiting End-to-End Speech-to-Text Translation From Scratch](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26193–26205. PMLR.
- Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2021a. [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. Association for Computational Linguistics.
- Hao Zhang, Nianwen Si, Yaqi Chen, Wenlin Zhang, Xukui Yang, Dan Qu, and Xiaolin Jiao. 2023b. [Tuning Large Language Model for End-to-end Speech Translation](#).
- Shucong Zhang, Erfan Loweimi, Peter Bell, and Steve Renals. 2021b. [On The Usefulness of Self-Attention for Automatic Speech Recognition with Transformers](#). In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 89–96.
- Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023c. [Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages](#).
- Ziqiang Zhang, Long Zhou, Junyi Ao, Shujie Liu, Lirong Dai, Jinyu Li, and Furu Wei. 2022b. [SpeechUT: Bridging Speech and Text with Hidden-Unit for Encoder-Decoder Based Speech-Text Pre-training](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1663–1676, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. 2018. [The Sound of Pixels](#). In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Jinming Zhao, Hao Yang, Gholamreza Haffari, and Ehsan Shareghi. 2022. [RedApt: An Adaptor for wav2vec 2 Encoding Faster and Smaller Speech Translation without Quality Compromise](#). In *Findings of the Association for Computational Linguistics*:

EMNLP 2022, pages 1960–1967, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Giulio Zhou, Tsz Kin Lam, Alexandra Birch, and Barry Haddow. 2024. **Prosody in Cascade and Direct Speech-to-Text Translation: A case study on Korean Wh-Phrases**. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 674–683, St. Julian's, Malta. Association for Computational Linguistics.

Yan Zhou, Qingkai Fang, and Yang Feng. 2023. **CMOT: Cross-modal Mixup via Optimal Transport for Speech Translation**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7873–7887, Toronto, Canada. Association for Computational Linguistics.

A Supplement to “Turning Segmentation into Augmentation”

A.1 Pseudocode for CTC-based Forced Alignment

In this section, we detail the algorithm used to recover textual transcriptions for the newly generated audio segments, as discussed in §4.4.3. This procedure is necessary because the original sentence-level transcripts align with the manual segmentation, not the synthetic boundaries produced by SEGAUGMENT. The complete process for a single document is formalized in Algorithm 2.

The procedure accepts the full document audio S , the original manual transcript X , and the set of new segmentation boundaries $\{\hat{b}_j^k\}$, where we drop the document index for ease of presentation. The process is divided into three main phases:

1. Alignment Preparation (Lines 1–3) First, the audio is passed through the CTC encoder $\mathcal{E}$ to obtain the frame-wise character probability matrix U . Simultaneously, the original transcript X must be normalized (X_{clean}) to match the output vocabulary of the CTC model. This normalization typically involves lowercasing and removing punctuation. The FORCEDALIGNMENT function then employs dynamic programming (e.g., Viterbi alignment) to map the characters in X_{clean} to the high-probability paths in U , resulting in precise start and end timestamps for every character.

2. Metadata Restoration (Lines 4–5) Since the alignment is performed on normalized text, the resulting timestamps correspond to stripped characters. To recover a usable transcript, we aggregate these character-level timestamps into word-level segments (JOINCHARS). We then map these normalized words back to their original surface forms from X via REVERSECLEAN, effectively restoring the original casing and punctuation (e.g., mapping aligned "hello" back to "Hello,").

3. Segmentation and Retrieval (Lines 6–10) Finally, the algorithm iterates through the new boundaries $\{\hat{b}\}_j^k$. For each segment $\hat{b}^j$, the function JOINWORDSWITHIN selects words whose timestamps fall within the segment’s start and end times. A POSTEDIT step performs minor heuristics to ensure grammatical correctness, such as capitalizing the first word of the new segment or removing trailing commas that were split from their following clauses.

Algorithm 2 CTC-based Forced Alignment for Transcript Recovery

Require: CTC Encoder $\mathcal{E}$, Manual Speech $\mathbf{S}$, Manual Transcript $\mathbf{X}$, New Segments $\{\hat{b}\}_j^k$

- 1: $\mathbf{U} \leftarrow \mathcal{E}(\mathbf{S})$ ▷ Extract CTC character probabilities
- 2: $\mathbf{X}_{clean} \leftarrow \text{CLEAN}(\mathbf{X})$ ▷ Normalize text to model vocab
- 3: $\text{chars} \leftarrow \text{FORCEDALIGNMENT}(\mathbf{X}_{clean}, \mathbf{U})$ ▷ Align clean text to audio
- 4: $\text{words} \leftarrow \text{JOINCHARS}(\text{chars})$ ▷ Merge chars into time-stamped words
- 5: $\text{words} \leftarrow \text{REVERSECLEAN}(\text{words})$ ▷ Restore casing and punctuation
- 6: $\hat{\mathbf{x}} \leftarrow []$
- 7: **for** $j \leftarrow 1$ **to** k **do**
- 8: $\hat{\mathbf{x}}^j \leftarrow \text{JOINWORDSWITHIN}(\hat{b}^j, \text{words})$ ▷ Filter words by time boundaries
- 9: $\hat{\mathbf{x}}^j \leftarrow \text{POSTEDIT}(\hat{\mathbf{x}}^j)$ ▷ Fix capitalization/detached punctuation
- 10: Append $\hat{\mathbf{x}}^j$ to $\hat{\mathbf{x}}$
- 11: **end for**
- 12: **return** $\hat{\mathbf{x}}$

A.2 Additional Experimental Details

S2T-TRANSFORMER Models. For all experiments, apart from §4.6.2 and from those in mTEDx spa-fra, we use s2t_transformer_s models implemented in FAIRSEQ (Wang et al., 2020b) (see §2.6.4). They have 12 encoder layers, 6 decoder layers, a dimensionality of 256, a feed-forward dimension of 2048, and 4 heads in the multi-head attention, with 31M parameters in total. We use GELU activations and pre-layernorm. The input is 80-dimensional log-Mel spectrograms, which are processed by a 2-layer convolutional network with 1024 inner channels, output dimension of 256, stride of 2, kernel size of 5, and GLU activations. We do not scale up the output of the convolutional network. The target vocabularies are learned with SentencePiece (Kudo and Richardson, 2018) and have a size of 8,000. We train with AdamW (Loshchilov and Hutter, 2019) with a learning rate of 0.002, a warm-up of 5,000 steps, and an inverse square root scheduler. We use a gradient accumulation of 6, making the effective batch size equal to 320 thousand tokens, apply SpecAugment (Park et al., 2019), and set the dropout to 0.1, applied to attention, activation, and output. The loss function is a standard cross entropy with label smoothing

of 0.1. We stop the training when the validation loss does not decrease for 10 epochs, and average the 10 best checkpoints according to validation BLEU.

The encoders of the ST models are pre-trained on the task of ASR using the same architecture. The only difference is the vocabulary size which is 5,000 and the learning rate which is 0.001. For the models trained without ASR pre-training (§4.6.5) we also used a learning rate of 0.001. For MuST-C, we pre-train using the ASR data of eng-deu, for mTEDx we use the spa-spa and por-por accordingly, and finally for CoVoST 2 the ASR data of eng-deu.

For mTEDx spa-fra we use an extra-small architecture (s2t_transformer_xs). It has 6 encoder layers and 3 decoder layers, a dimensionality of 256, a feed-forward dimension of 1024, 4 heads in the multi-head attention, and a vocabulary size of 3,000, having a total of 10M parameters. The learning rate is set to 0.001 (warm-up of 500), the batch size to 180 thousand tokens, and use a dropout of 0.2. We also share the weights of the embedding layer and output projection in the decoder. The same model is pre-trained on ASR, but with a dropout of 0.1. All other hyperparameters are the same as for the small models described before.

All our experiments were run on a cluster with 8 NVIDIA GeForce rtx 2080 ti. The running times of each experiment on a single GPU ranged from 12 to 36 hours.

w2v-MBART Models. For the experiments of §4.6.2, we use a strong baseline utilizing pre-trained models and a length adaptor (see §2.6.4). The encoder is composed of a 7-layer convolutional feature extractor and 24-layer Transformer encoder, while the decoder has 12 layers, and a vocabulary of size 250k, with 770M parameters in total. All the layers have an embedding dimensionality of 1024, a feed-forward dimensionality of 4098, GELU activations, 16 attention heads, and pre-layer normalization configuration. A strided 1D convolutional layer sub-samples the output of the encoder by 2 times. The encoder is initialized from WAV2VEC 2.0,¹ which is pretrained with 60K hours of non-transcribed speech from Libri-Light, and fine-tuned for ASR with 960 hours of labeled data from Librispeech. The decoder is initialized from MBART-50,² which is fine-tuned english-to-many multilingual machine translation. We fine-tune all the parameters of the model, apart from the feature extractor of the encoder and the embedding layer in the decoder. The inputs to the model are raw waveforms sampled at 16kHz, which are normalized to

¹fairseq/wav2vec/wav2vec2_vox_960h_new.pt

²fairseq/models/mbart50/mbart50.ft.1n.tar.gz

zero mean and unit variance. We train with AdamW using a base learning rate of 0.0005, with a warm-up for 2,000 steps and an inverse square root scheduler. In the encoder we use 0.1 activation dropout, time masking with probability of 0.2 and channel masking with probability of 0.1. In the decoder we use a dropout of 0.3, and attention dropout of 0.1. All other dropouts are not active. The loss function is a standard cross entropy with label smoothing of 0.2. We use gradient accumulation to have an effective batch size of 32M tokens, evaluate every 250 steps, and stop the training when the performance on the validation set does not improve for 20 evaluations. We average the 10 best checkpoints according to the validation BLEU, and generate with a beam search of 5.

Machine Translation Models. For the MT models used for text alignment in SEGAUGMENT (§4.4.3), we used medium-sized Transformers, with 6 encoder and decoder layers, dimensionality of 512, feed-forward dimension of 2048, and 8 heads in the multi-head attention. We train with AdamW using a learning rate of 0.002, with a warm-up for the first 2,500 updates, and the batch size is set to 14 thousand tokens. The vocabulary size is 8,000, and for regularization, we use a small dropout of 0.1 only at the outputs of each layer and label smoothing of 0.1. We stop training when the document-level BLEU does not increase for 20 epochs, average the 10 best checkpoints according to the same metric, and then generate with a beam of size 8.

A.3 Additional Experimental Results

ASR Pre-training. We present the results of ASR pre-training in Table A.1. We observe that including data from SEGAUGMENT improves WER across all ASR models. This indicates that the proposed method is also useful for ASR augmentation, likely due to the increased source-side contextual and positional diversity introduced during training (§4.8).

Ablations on Combinations of Real and Synthetic Data. We analyze the impact of different SEGAUGMENT data combinations on ST performance using the MuST-C v2.0 eng-deu benchmark (Table A.2). Comparing rows 2–6 against the baseline (row 1), we find that while individual synthetic datasets underperform compared to the original data, combining all synthetic versions (row 6) surprisingly surpasses the baseline by 1.1

Model	MuST-C	mTEDx		CoVoST
	eng-deu	spa-spa	por-por	eng-deu
S2T-TRANSFORMER	11.1	18.0 (19.7) ^{xs}	28.3	26.6
+ SEGAUGMENT	10.4	16.6 (18.1)^{xs}	23.8	23.8

Table A.1: WER(↓) scores for ASR models in test sets of MuST-C 2.0, mTEDx, and CoVoST 2. In **bold** is the best score overall. The superscript *xs* indicates the extra-small architecture configuration.

BLEU. This suggests that SEGAUGMENT generates high-quality data capable of training competitive models even in isolation. Furthermore, performance consistently improves as more synthetic variations are added, with the best results achieved by using the full set of augmentations alongside the original data (row 17). We also observe that combinations involving longer segments (rows 8–10, 13–14) tend to outperform those with shorter ones (rows 7, 11–12) when limited to fewer synthetic sets.

Applying SEGAUGMENT on ASR and MT Models. We investigate the impact of synthetic data on ASR and MT tasks using MuST-C v2.0 eng-deu (Table A.3). The inclusion of all synthetic subsets (row 6) yields a 0.7 reduction in WER and a 1.1 increase in BLEU, suggesting that SEGAUGMENT benefits both source-side acoustic modeling and target-side translation. Consistent with our ST findings, longer segments favor MT performance (rows 4–5), likely due to the need for broader context in translation (Voita et al., 2018), whereas shorter segments favor ASR (rows 2–3), aligning with the local nature of acoustic features (Zhang et al., 2021b).

More Automatic Segmentations. In Table A.4 we provide an extended version of Table 4.7, where we evaluate the models on four Automatic (SHAS) segmentations of the test set. Regarding the choice of automatic segmentation, we find that the *long* segmentation is the best when using SEGAUGMENT with special tokens (27.3 BLEU), *long* and *extra-long* are equally good without special tokens (26.9 BLEU) and the *medium* segmentation is best without SEGAUGMENT (23.7 BLEU). Using the special tokens reduces the performance on the manual segmentation by 0.3 BLEU (27.4 → 27.1) but increases it in the SHAS-*long* segmentation by 0.4 BLEU (26.9 → 27.3), thus bridging the gap with the manual segmentation.

	Original	SEGAUGMENT				BLEU
		<i>s</i>	<i>m</i>	<i>l</i>	<i>xl</i>	
(1)	✓					24.3
(2)		✓				22.6
(3)			✓			23.4
(4)				✓		22.4
(5)					✓	22.7
(6)		✓	✓	✓	✓	25.4
(7)	✓	✓				24.8
(8)	✓		✓			25.1
(9)	✓			✓		25.1
(10)	✓				✓	25.1
(11)	✓	✓	✓			25.6
(12)	✓		✓	✓		25.6
(13)	✓	✓			✓	25.8
(14)	✓			✓	✓	25.8
(15)	✓	✓	✓	✓		26.0
(16)	✓		✓	✓	✓	26.0
(17)	✓	✓	✓	✓	✓	26.2

Table A.2: BLEU(↑) scores on MuST-C v2.0 tst-COMMON (eng-deu) for models trained with different combinations of original and synthetic data.

A.4 SEGAUGMENT Data Statistics

In Table A.5 and Table A.6 detail the count and average duration of examples generated by SEGAUGMENT across all language pairs, considering only segments between 0.4s and 30s. For CoVoST 2, we prioritized length constraints over the classification threshold in the segmentation algorithm to ensure the synthetic data remained distinct from the manual segmentation. This approach yields segments that, while not always syntactically perfect sentences, provide greater diversity for training; consequently, no *large* or *extra-large* variations were generated for this dataset.

	Original	SEGAUGMENT				WER↓	BLEU↑
		<i>s</i>	<i>m</i>	<i>l</i>	<i>xl</i>		
(1)	✓					11.1	31.0
(2)	✓	✓				10.2	31.3
(3)	✓		✓			10.3	31.6
(4)	✓			✓		11.0	31.7
(5)	✓				✓	10.9	31.7
(6)	✓	✓	✓	✓	✓	10.4	32.1

Table A.3: WER and BLEU scores for ASR and MT models respectively on MuST-C v2.0 eng-deu tst-COMMON.

Lang. Pair	Model	Test Set Segmentation				
		Manual	Automatic			
			<i>s</i>	<i>m</i>	<i>l</i>	<i>xl</i>
eng-deu	S2T-TRANSFORMER	23.2	20.6	21.7	21.2	20.2
	+ SEGAUGMENT	25.0	23.1	24.3	24.5	24.7
	+ <i>special tok.</i>	25.1	23.1	24.5	25.2	25.1
eng-spa	S2T-TRANSFORMER	27.5	25.0	26.4	26.1	24.8
	+ SEGAUGMENT	29.3	27.3	28.7	29.0	28.9
	+ <i>special tok.</i>	29.2	26.6	28.6	29.4	29.1
eng-fra	S2T-TRANSFORMER	33.1	30.0	31.0	30.6	29.0
	+ SEGAUGMENT	35.7	32.3	34.1	34.9	34.8
	+ <i>special tok.</i>	35.6	31.4	33.8	35.3	35.1
eng-ita	S2T-TRANSFORMER	22.9	20.0	21.6	21.5	20.3
	+ SEGAUGMENT	25.6	22.7	24.2	24.4	24.8
	+ <i>special tok.</i>	25.0	22.3	24.3	25.1	24.5
eng-nld	S2T-TRANSFORMER	27.3	24.9	26.5	25.7	24.3
	+ SEGAUGMENT	30.3	27.5	29.2	29.7	29.7
	+ <i>special tok.</i>	29.4	26.9	26.9	29.8	29.3
eng-por	S2T-TRANSFORMER	28.7	25.8	27.1	26.9	25.4
	+ SEGAUGMENT	31.8	29.2	30.7	31.4	31.1
	+ <i>special tok.</i>	31.5	28.8	30.8	31.8	31.3
eng-ron	S2T-TRANSFORMER	22.2	19.9	20.8	20.5	18.7
	+ SEGAUGMENT	25.0	22.5	23.4	24.1	24.3
	+ <i>special tok.</i>	24.8	22.6	24.1	24.6	24.0
eng-rus	S2T-TRANSFORMER	15.3	13.4	14.7	14.6	13.8
	+ SEGAUGMENT	16.8	15.0	16.4	16.8	16.7
	+ <i>special tok.</i>	16.4	14.9	16.4	17.0	16.8
Average	S2T-TRANSFORMER	25.0	22.4	23.7	23.4	22.1
	+ SEGAUGMENT	27.4	24.9	26.4	26.9	26.9
	+ <i>special tok.</i>	27.1	24.6	26.2	27.3	26.9

Table A.4: BLEU(↑) scores of different methods on MuST-C v1.0 tst-COMMON evaluated on Manual (gold) and Automatic (SHAS) segmentations across different granularity lengths (*s*, *m*, *l*, *xl*). The best score for each column is highlighted. The best score overall for each language pair is in **bold**.

Dataset	Ver.	Pair	Original	SEGAUGMENT				
				<i>s</i>	<i>m</i>	<i>l</i>	<i>xl</i>	Concat
MuST-C	v1.0	eng-deu	225	433	253	120	76	882
		eng-spa	260	430	253	120	76	879
		eng-fra	269	421	253	120	76	871
		eng-ita	248	399	253	120	76	848
		eng-nld	244	433	253	120	76	882
		eng-por	201	433	253	120	76	882
		eng-ron	231	433	253	120	76	882
		eng-rus	260	419	253	120	76	868
	v2.0	eng-deu	249	401	231	109	70	812
mTEDx		spa-spa	101	150	88	43	27	308
		por-por	90	136	78	37	24	274
		spa-eng	36	53	32	15	10	110
		por-eng	30	47	27	13	8	94
		spa-fra	3.5	5.5	3	1.5	1	11
CoVoST 2		eng-deu	231	504	255	—	—	759

Table A.5: Number of examples (in thousands) in the Original and SEGAUGMENT datasets.

Dataset	Ver.	Pair	Original	SEGAUGMENT			
				<i>s</i>	<i>m</i>	<i>l</i>	<i>xl</i>
MuST-C	v1.0	eng-deu	6.3	3.8	6.9	15.2	24.0
		eng-spa	6.6	3.8	6.9	15.2	24.0
		eng-fra	6.3	3.8	6.9	15.2	24.0
		eng-ita	6.4	3.8	6.9	15.2	24.0
		eng-nld	6.2	3.8	6.9	15.2	24.0
		eng-por	6.5	3.8	6.9	15.2	24.0
		eng-ron	6.4	3.8	6.9	15.2	24.0
		eng-rus	6.5	3.8	6.9	15.2	24.0
	v2.0	eng-deu	6.3	3.8	6.8	15.2	23.9
mTEDx		spa-spa	5.9	3.8	6.9	15.2	23.9
		por-por	5.8	3.7	6.9	15.2	24.0
		spa-eng	6.1	3.9	6.9	15.3	23.8
		por-eng	5.9	3.7	6.8	15.2	24.0
		spa-fra	6.1	3.8	6.9	15.1	23.8
CoVoST 2		eng-deu	5.6	1.8	3.7	—	—

Table A.6: Average duration (in seconds) of examples in the Original and SEGAUGMENT datasets.

B Supplement to “Zero-Shot Speech Translation via Cross-Modal Alignment”

B.1 Dataset Details and Filtering

In this section, we provide a detailed description of the data processing and filtering pipelines applied to the training datasets used in Ch. 5.

MuST-C v1.0. For the training set of each language pair in MuST-C v1.0 (Di Gangi et al., 2019a), we perform text cleaning on both the transcription and translation. This includes space and punctuation normalization as well as the removal of non-UTF characters. We specifically remove speaker names (e.g., “Speaker 1:”) and non-speech events such as “(Laughing)”.

We filter out examples that contain fewer than 4 characters or have a transcription-to-translation ratio smaller than 0.5 or larger than 2.0. For the Zero-shot ST training, we strictly discard the translations. The transcriptions are further processed by replacing numbers and symbols with their spelled-out forms to match the acoustic content.

To create the final training set, we concatenate all eight language-pair train sets (eng-deu, eng-spa, eng-fra, eng-ita, eng-nld, eng-por, eng-ron, eng-rus). We then remove duplicates based on the triplet of (speaker, text, duration). Due to alignment errors during the original dataset creation, the same text is sometimes paired with different audio segments across different language pairs. In such cases of (speaker, text) duplicates, we retain the example that is closest to the average speaking rate of that speaker, calculated as the average words-per-second of all examples for that speaker in the concatenated dataset. Finally, we remove examples with extreme speaking rates (greater than 10 words per second).

We similarly construct a validation set, totaling 2,081 examples, for model selection. This is created by concatenating the individual dev sets (without filtering) and removing hard duplicates based on the (speaker, text, duration) triplet.

Dataset	Examples (K)	Hours
eng-deu	220	384
eng-spa	254	473
eng-fra	262	466
eng-ita	242	438
eng-nld	237	416
eng-por	196	359
eng-ron	226	406
eng-rus	250	458
MuST-C ASR (Total Raw)	1,888	3,402
↪ Duplicate removal	367	713
↪ Speaking ratio filtering	353	681

Table B.1: Number of examples (in thousands) and total duration of the MuST-C train set used to train ZEROSWOT. We show the initial size per language pair (eng-X) and the final size after concatenation and filtering.

Common Voice. We utilize version 8.0 of the Common Voice dataset (Ardila et al., 2020), specifically the train and valid splits. For the training split, we apply standard text normalization and replace numbers and symbols with their spelled-out forms.

LibriSpeech. LibriSpeech is only used as additional ASR training data for the experiments on FLEURS in the Appendix §B.6.1. For training, we use the train-clean-100, train-clean-360, and train-other-500 splits (Panayotov et al., 2015). For development, we use the dev-clean split.

Since LibriSpeech provides normalized, uppercase transcriptions without punctuation, we must restore them to a format compatible with the NLLB tokenizer. We restore casing and punctuation using a BERT-base model (Devlin et al., 2019) fine-tuned for punctuation restoration, specifically the rpunct tool.¹

CTC Label Construction. For all datasets, the target labels for the CTC loss are constructed as follows:

¹github.com/Felflare/rpunct

1. The transcription is tokenized using the NLLB SentencePiece model.²
2. We remove casing information.
3. All spaces are replaced with the special separator token <sep>.
4. Any character not present in the WAV2VEC 2.0 letter-based vocabulary³ is substituted with the <unk> token.

B.2 ZEROSWOT Model Configuration

In this section, we provide detailed specifications for the model architectures, training hyperparameters, and the specific configurations used for the ablation studies presented in Ch. 5.

B.2.1 Model Architecture

Acoustic Encoder. The acoustic encoder comprises a feature extractor with 7 strided convolutional layers that sub-sample the input signal by a factor of 320, followed by a stack of 24 Transformer layers with dimensionality $d = 1024$ (Baevski et al., 2020). Each layer utilizes a feed-forward dimension of 4096, 16 attention heads, GELU activations (Hendrycks and Gimpel, 2023), and a pre-layernorm configuration. Both the acoustic encoder and the linear projection layer of the CTC module (vocabulary size $|\mathcal{V}| = 30$) are initialized from the WAV2VEC 2.0 LARGE checkpoint⁴ which contains approximately 315M parameters. This checkpoint was pre-trained via self-supervised learning with self-training (Xu et al., 2021b) and subsequently fine-tuned on ASR tasks using LibriSpeech (Panayotov et al., 2015) and Libri-light (Kahn et al., 2020).⁵

Compression Adapter. The subword encoder within the compression adapter is a BERT-style Transformer encoder (Devlin et al., 2019) consisting of 3 layers with dimensionality $d = 1024$, matching the architectural format of the acoustic encoder layers. We add sinusoidal positional encodings to the chunked input before processing.

²github.com/facebookresearch/fairseq/tree/nllb

³dl.fbaipublicfiles.com/fairseq/wav2vec/dict.ltr.txt

⁴Wav2Vec 2.0 Large (LV-60) + Self Training / 960 hours / Libri-Light + Librispeech

⁵github.com/facebookresearch/fairseq/blob/main/examples/wav2vec/README.md

Text Branch. The text embedding layer utilizes a vocabulary of size $|\mathcal{B}| = 256k$. For the text encoder, we employ either the MEDIUM version (12 layers) or the LARGE version (24 layers). In the MEDIUM configuration, each layer has a model dimension of 1024, a feed-forward dimension of 4096, ReLU activation, 16 attention heads, and pre-layernorm configuration. The LARGE version differs only in the number of layers and the feed-forward dimension, which is increased to 8192. We initialize these components using dense, distilled versions of NLLB (NLLB, 2024), derived from the 54B parameter Mixture-of-Experts model: specifically, the 600M parameter and 1.3B parameter distilled checkpoints.⁶

Small Model Variant. The SMALL version of ZEROSWOT utilizes the WAV2VEC 2.0 BASE checkpoint⁷ (12 layers, dimensionality 768) and NLLB-600M model. The subword encoder in this configuration has 3 layers with dimensionality 768. A linear projection layer is used to map its output to $d = 1024$ to match the dimensionality of the NLLB text encoder. This model has approximately 110M trainable parameters and 0.7B parameters during inference.

B.2.2 Training Details

We finetune the parameters of the acoustic encoder, CTC module, and compression adapter. The text embedding and text encoder remain frozen throughout training. Optimization is performed using AdamW (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, a base learning rate of $3 \cdot 10^{-4}$, and an inverse square root scheduler with linear warmup (2k steps for MuST-C, 4k steps for Common Voice). The effective batch size is set to 32 million tokens. The input to the speech branch consists of raw waveforms sampled at 16kHz, while the text branch receives English text tokenized with the NLLB SentencePiece model.

Loss Implementation. The Optimal Transport loss is implemented using the geomloss library,⁸ utilizing the Sinkhorn distance (Cuturi, 2013) with default parameters. When applying the loss across multiple layers, we batch the computations for efficiency. Crucially, the losses for intermediate layers are applied after the first layer normalization of the

⁶github.com/facebookresearch/fairseq/tree/nllb

⁷Wav2Vec 2.0 Base / 960 hours / Librispeech

⁸kernel-operations.io/geomloss

subsequent layer (due to the pre-layernorm configuration), as applying losses to un-normalized representations led to training instability.

Regularization. We apply activation dropout of 0.1 in the acoustic encoder, dropout of 0.1 in the CTC module, and dropout of 0.1 in the compression adapter. Additionally, we employ WAV2VEC 2.0 masking on the speech signal (Baevski et al., 2020), masking 10 consecutive frames with a probability of 0.3 and 64 consecutive frequency channels with a probability of 0.2.

Optimization and Compute. To accelerate training, we extract the text encoder representations offline, allowing us to remove the frozen text branch from memory during the speech encoder training phase. We train until convergence, applying early stopping if the OT loss of the final layer on the development set does not improve for 10 epochs. Models trained on LibriSpeech or MuST-C typically converge in approximately 50k steps, while those on Common VOice require around 150k steps.

For models trained on Common Voice, we accelerate the process by training a *seed* model using the original NLLB version. We then adapt this pre-trained speech encoder to different NLLB versions. This adaptation phase requires only 30k steps (versus 150k for the seed model), utilizing a learning rate of $3 \cdot 10^{-5}$, a 1k step warmup, and cosine annealing.

All models are implemented in FAIRSEQ (Ott et al., 2019; Wang et al., 2020b) using memory-efficient-fp16 training. Training for 50k steps takes approximately 36 hours on a machine equipped with 8 NVIDIA RTX 3090 GPUs.

B.2.3 Compression Ablation Configurations

This section details the configurations for the ablation study on compression methods presented in §5.7.1. To ensure fair comparison, we maintain a consistent parameter count across all models by adding Transformer layers or MLPs where necessary. Architectural modifications are restricted to the compression module and all other components remain identical.

- **No Compression:** We add two large MLPs with residual connections at the output of the acoustic encoder. Each MLP consists of an up-projection to 8192, GELU activation, and down-projection back to 1024.
- **Length Adaptor:** Following [Li et al. \(2021\)](#), we employ 2 strided convolutional layers that sub-sample the acoustic encoder output by a factor of 4, followed by a single Transformer layer.
- **CTC-based Compression:** Following [Liu et al. \(2020b\)](#); [Gaido et al. \(2021a\)](#), we mean-pool consecutive representations with identical CTC labels (as described in §5.4.1) and process the result with 3 Transformer layers.
- **Tokenization Variants:** For models using our proposed compression adapter with different tokenization schemes (e.g., word vs. subword), we modify only the target labels $\mathbf{z}$ used for the CTC loss.

B.2.4 Supervised ST Fine-tuning Details

This section provides details for the supervised fine-tuning experiments of §5.6.4. We train for 20k steps with a learning rate of $4 \cdot 10^{-5}$, using a linear warmup for 1k steps, a fixed rate for 4k steps, and cosine annealing decay for the remainder.

For MEDIUM models, we fine-tune the parameters of the compression adapter, text encoder, and text decoder, keeping the acoustic encoder, speech embedder, and text embedding frozen. For LARGE models, we fine-tune only the compression adapter and apply Layer-Norm and Adapter (LNA) fine-tuning to the text encoder-decoder ([Li et al., 2021](#)). We note that more advanced fine-tuning parameter-efficient methods such as LoRA ([Hu et al., 2022](#)) could potentially yield further improvements. For inference, we average the 10 best checkpoints based on BLEU score on the MuST-C eng-deu dev set.

B.3 Machine Translation Models

Model Architecture. For the Machine Translation component, we utilize the NLLB-600M and NLLB-1.3B models. These are dense Transformer networks distilled from a 54B parameter Mixture-of-Experts (MoE) teacher model ([NLLB, 2024](#)), supporting many-to-many translation across 202 languages.

The NLLB-600M model consists of 12 layers in both the encoder and decoder, with a model dimensionality of 1024, a feed-forward dimension of 4096, 16 attention heads, ReLU activations, and a pre-layernorm configuration. The NLLB-1.3B model differs by having 24 layers in both the encoder and decoder, and an increased feed-forward dimension of 8192. Both variants share a multilingual vocabulary of 256k tokens, learned via SentencePiece (Kudo and Richardson, 2018).

Fine-tuning Hyperparameters. To adapt the generic NLLB models to the specific domains of MuST-C and CoVoST, we performed multilingual finetuning using the available transcription-translation pairs. We employed a batch size of 32k tokens for the 600M model and 64k tokens for the 1.3B model. Optimization was performed using AdamW (Loshchilov and Hutter, 2019) with 2k warm-up updates and an inverse square root scheduler.

We determined the optimal learning rate and dropout rate via a grid search over the values $\{7.5 \cdot 10^{-5}, 1 \cdot 10^{-4}, 2 \cdot 10^{-4}\}$ and $\{0.1, 0.2, 0.3\}$, respectively.⁹ Consistent with the original NLLB training configuration (NLLB, 2024), we applied a dropout of 0.1 to attention weights, 0.0 to activations, and used label smoothing of 0.1. Evaluation on the validation sets was performed every 500 steps, with early stopping triggered after 20 consecutive non-improving evaluations. The final model is obtained by averaging the 10 best checkpoints based on validation loss. Training was conducted with fp16 precision. For the 1.3B version, we utilized memory-efficient-fp16 (Ott et al., 2019) to fit the model on NVIDIA RTX 3090 GPUs. Convergence typically required 40k steps for MuST-C and approximately 100k steps for CoVoST.

B.4 Cascade Speech Translation Systems

In this section, we detail the training configurations for the Cascade ST systems used as baselines in the experiments of §5.6.3. These systems consist of an ASR model (either CTC-based or Attention-based) chained with an MT model (either NLLB-600M or NLLB-1.3B, optionally finetuned as described in §B.3).

⁹We ran 9 experiments on MuST-C for each NLLB version and transferred the optimal hyperparameters to CoVoST.

CTC ASR Models. The encoder architecture for the CTC-based ASR models is identical to the acoustic encoder used in ZEROSWOT (see §B.2.1). We initialize the encoder using the same pre-trained WAV2VEC 2.0 checkpoint and attach a randomly initialized CTC head. We then fine-tune the entire model using the CTC objective on either the MuST-C ASR or Common Voice datasets.

The data filtering and preprocessing pipelines are identical to those used for ZEROSWOT (see §B.1), utilizing the standard CTC labeling scheme described in Table 5.1 (Row 1). We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $1 \cdot 10^{-4}$, a 1k step warmup, and an inverse square root scheduler. The effective batch size is 32 million tokens. Regularization includes dropout of 0.1 on activations and the CTC module, along with WAV2VEC 2.0 masking (probability 0.5 for time masks of length 10; probability 0.25 for channel masks of size 64) (Baevski et al., 2020). We evaluate every 250 steps, stopping after 10 consecutive evaluations without improvement, and average the 10 best checkpoints. The final model has approximately 315 million parameters.

Attention-based Encoder-Decoder (AED) ASR Models. The encoder architecture remains identical to the CTC ASR model. The decoder consists of 6 Transformer layers with a dimensionality of 768, a feed-forward dimension of 3072, 8 attention heads, GELU activations (Hendrycks and Gimpel, 2023), and pre-layernorm configuration.

We initialize the encoder with a pre-trained WAV2VEC 2.0 LARGE checkpoint¹⁰ and train the full encoder-decoder model using cross-entropy loss on LibriSpeech. Subsequently, we fine-tune it on either MuST-C ASR or Common Voice. The target vocabulary size is 16k, learned jointly across all three ASR datasets using SentencePiece. Data filtering follows the procedures in §B.1.

Hyperparameters are consistent across the pre-training (LibriSpeech) and fine-tuning phases. We use AdamW with a learning rate of $2 \cdot 10^{-4}$, a 1k step warmup, inverse square root scheduler, and a batch size of 32 million tokens. Encoder masking and dropout match the CTC configuration, while the decoder uses a dropout of 0.1 and label smoothing of 0.1. We evaluate every 500 steps, apply early stopping after 10 non-improving updates, and average the 10 best checkpoints. The total parameter count is approximately 390 million.

¹⁰Wav2Vec 2.0 Large (LV-60) / 960 hours / Libri-Light + Librispeech

Inference. Cascade inference is performed sequentially: we first generate the transcript from the ASR model using beam search with a beam size of 5. This transcript is then passed to the MT model, which generates the final translation, also using a beam size of 5.

B.5 Evaluation

The evaluation of all models, either MT, ZEROSWOT, ASR, Cascade ST, or Supervised ST remains the same. We apply checkpoint averaging to the 10 best checkpoints according to the corresponding validation metric, and generate with a beam search of 5. We measure translation quality with BLEU (Papineni et al., 2002) using sacreBLEU (Post, 2018).¹¹ For comparing with SEAMLESSM4T, we follow their evaluation, and using a char tokenization for Mandarin Chinese, Japanese, Thai, Lao and Burmese (Seamless Communication, 2023a). We do not carry statistical significance testing since in almost all results we are comparing average BLEU scores across many language directions. Although BLEU as a metric has its limitations (Mathur et al., 2020b), we opted to use it for comparison with previous methods from the literature.

B.6 Additional Results

B.6.1 Massively Multilingual Evaluation on FLEURS

To verify the massively multilingual capabilities of our method, we evaluate ZEROSWOT on the FLEURS benchmark (Conneau et al., 2023), which is based on FLORES-101 (Goyal et al., 2022). We use the English-to-X portion of the dataset, covering 88 target languages. This evaluation tests whether the alignment learned by the speech encoder generalizes to the vast language coverage of the underlying NLLB model.

We evaluate two variants of our model: one trained solely on Common Voice (Ardila et al., 2020), and another trained with additional ASR data from MuST-C (Di Gangi et al., 2019a) and LibriSpeech (Panayotov et al., 2015) (“+ More ASR Data”). Both variants

¹¹BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

Model	Type	Medium	Large
NLLB	MT (upper bound)	22.5	24.8
SEAMLESSM4T-v1	Supervised E2E ST	19.2	21.5
Cascade (AED-ASR $\rightarrow$ NLLB)	Cascade ST	17.8	20.2
ZEROSWOT	Zero-shot ST	18.1	20.1
ZEROSWOT (+ More ASR Data)	Zero-shot ST	18.4	20.3

Table B.2: Average BLEU scores on the FLEURS test set (English $\rightarrow$ 88 languages). We compare ZEROSWOT against the NLLB text baseline, a strong cascade, and the supervised SEAMLESSM4T model. Results of SEAMLESSM4T-v1 taken from [Seamless Communication, 2023a](#). NLLB results are obtained with NLLB-600M and NLLB-1.3B. “+ More ASR Data” indicates training with additional ASR data from MuST-C and LibriSpeech. **Bold** indicates the best ST score in each size category.

are strictly testing out-of-domain knowledge since the evaluation test of FLEURS is of different domain than our ASR training data, and NLLB is not adapted. Thus, we test for massive multilinguality while operating in a pure zero-shot scenario with respect to task, languages and domains (Scenario A in Table 5.4).

We compare these against three baselines:

- **NLLB (MT):** The text-to-text translation performance, serving as an upper bound or oracle for the speech translation models.
- **Cascade (AED-ASR $\rightarrow$ NLLB):** A strong cascade system combining an Attention-based Encoder-Decoder ASR model trained on Common Voice with the NLLB translation model.
- **SEAMLESSM4T:** A state-of-the-art supervised End-to-End ST model trained on 400k hours of audio data ([Seamless Communication, 2023a](#)).

The results, summarized in Table B.2, demonstrate the robustness of our approach. Despite being a zero-shot model, ZEROSWOT achieves performance comparable to the strong supervised baseline SEAMLESSM4T, particularly in the LARGE configuration (20.3 vs 21.5 BLEU), even though SEAMLESSM4T utilizes orders of magnitude more training data. Furthermore, our method performs on par with the strong cascade system (18.1 vs 17.8 for Medium, 20.1 vs 20.2 for Large), confirming that the cross-modal alignment is as effective as explicit transcription for multilingual transfer. Adding more English ASR data yields consistent, albeit small, improvements (+0.3/+0.2 BLEU), suggesting that the quality of

the speech encoder’s representations can still be enhanced in the presence of more training data.

B.6.2 Extended Results in MuST-C and CoVoST

MuST-C. In Table B.3 we provide a summary of all obtained results in MuST-C, with our ST results using our ZEROSWOT and cascade models, as well as our MT results with the original and finetuned NLLB models. We also provide more results from the literature for both zero-shot and supervised ST models. Note on performance drop of CTC-based Cascade when moving from the original NLLB model (scenario B) to the finetuned one (scenario C). We hypothesize that the original NLLB can better handle the noisy English text produces by the CTC ASR model, due to the abundance of data it was trained on. When finetuning it using the MuST-C MT data, the model losses some of this capacity. In Table B.4 we provide the WER scores in MuST-C of the ASR models trained for the CTC-based and AED-based cascades (§5.6.3). Models trained on Common Voice correspond to out-of-domain training (Scenario A), while models trained on MuST-C ASR correspond to scenarios B and C.

CoVoST. In Table B.5 we present the extended results on CoVoST 2, including the ST results of ZEROSWOT, the supervised ST results of the baselines, and the MT results with the original NLLB models and our CoVoST-finetuned adapted variants.

B Supplement to “Zero-Shot Speech Translation via Cross-Modal Alignment”

Models	MuST-C Data			Params	BLEU								
	ASR	MT	deu		spa	fra	ita	nld	por	ron	rus	Average	
Machine Translation													
NLLB-MEDIUM (original, used in A, B)	-	✗	0.6	32.7	36.9	45.2	32.2	36.0	37.4	30.3	21.0	34.0	
NLLB-MEDIUM-FT This Work	-	✓	0.6	34.4	38.8	44.6	34.7	39.0	41.6	32.1	22.4	35.9	
NLLB-LARGE (original, used in A, B)	-	✗	1.3	34.6	38.6	46.8	33.7	38.2	39.6	31.8	23.2	35.8	
NLLB-LARGE-FT This Work	-	✓	1.3	35.3	39.9	45.8	36.0	40.6	43.1	32.6	23.9	37.2	
Zero-shot End-to-End Speech Translation													
MULTISLT (Escolano et al., 2021b)	✓	✓	0.05	6.8	6.8	10.9	-	-	-	-	-	-	
T-MODULES (Duquenne et al., 2022)	✓	✗	2	23.8	27.4	32.7	-	-	-	-	-	-	
DCMA (Wang et al., 2022)	✓	✗	0.15	22.4	24.6	29.7	18.4	22.0	24.2	16.8	11.8	21.2	
DCMA (Wang et al., 2022)	✓	✓	0.15	24.0	26.2	33.1	24.1	28.3	29.2	22.2	16.0	25.4	
SIMZEROCR (Gao et al., 2024)	✓	✓	0.15	25.1	27.0	34.6	-	-	-	-	15.6	-	
TOWARDS-ZERO-SHOT (Yang et al., 2023)	✓	✗	0.1	23.4	26.5	33.7	-	-	-	-	-	-	
TOWARDS-ZERO-SHOT (Yang et al., 2023)	✓	✓	0.1	26.5	29.5	35.3	-	-	-	-	-	-	
This Work Zero-shot End-to-End Speech Translation													
ZEROSWOT-SMALL (WAV2VEC 2.0-BASE / NLLB-MEDIUM)	(B)	✓	✗	0.1 / 0.7	27.0	31.6	35.6	26.6	30.9	31.5	25.1	17.9	28.3
ZEROSWOT-MEDIUM (WAV2VEC 2.0-LARGE / NLLB-MEDIUM)	(A)	✗	✗	0.35 / 0.95	24.8	30.0	32.6	24.1	28.6	28.8	22.9	16.4	26.0
	(B)	✓	✗	0.35 / 0.95	28.5	33.1	37.5	28.2	32.3	32.9	26.0	18.7	29.6
	(C)	✓	✓	0.35 / 0.95	30.5	34.9	39.4	30.6	35.0	37.1	27.8	20.3	31.9
ZEROSWOT-LARGE (WAV2VEC 2.0-LARGE / NLLB-LARGE)	(A)	✗	✗	0.35 / 1.65	26.5	31.1	33.5	25.4	29.9	30.6	24.3	18.0	27.4
	(B)	✓	✗	0.35 / 1.65	30.1	34.8	38.9	29.8	34.4	35.3	27.6	20.4	31.4
	(C)	✓	✓	0.35 / 1.65	31.2	35.8	40.5	31.4	36.3	38.3	28.0	21.5	32.9
This Work Cascaded Speech Translation with CTC ASR model													
WAV2VEC 2.0 → NLLB-MEDIUM	(A)	✗	✗	(0.3 + 0.6) / 0.9	21.9	24.4	27.5	19.3	23.7	24.0	19.8	14.2	21.8
	(B)	✓	✗	(0.3 + 0.6) / 0.9	26.1	28.1	32.6	22.9	27.9	27.9	23.5	16.6	25.7
	(C)	✓	✓	(0.3 + 0.6) / 0.9	25.1	26.8	30.1	20.9	27.6	28.5	20.7	17.8	24.7
WAV2VEC 2.0 → NLLB-LARGE	(A)	✗	✗	(0.3 + 1.3) / 1.6	24.7	27.1	30.3	21.6	26.8	27.4	22.9	16.9	24.7
	(B)	✓	✗	(0.3 + 1.3) / 1.6	27.8	29.9	34.4	24.6	29.6	30.0	25.6	18.5	27.5
	(C)	✓	✓	(0.3 + 1.3) / 1.6	27.0	28.4	31.9	22.6	29.5	31.4	22.1	19.0	26.5
This Work Cascaded Speech Translation with Attention-based Encoder-Decoder ASR model													
WAV2VEC 2.0-TRANSFORMER → NLLB-MEDIUM	(A)	✗	✗	(0.4 + 0.6) / 1	24.3	27.9	33.2	23.0	27.0	27.2	22.5	16.7	25.2
	(B)	✓	✗	(0.4 + 0.6) / 1	29.0	32.6	39.8	27.9	32.0	32.9	27.1	18.9	30.0
	(C)	✓	✓	(0.4 + 0.6) / 1	31.1	34.4	41.4	30.2	34.7	37.2	28.3	20.6	32.2
WAV2VEC 2.0-TRANSFORMER → NLLB-LARGE	(A)	✗	✗	(0.4 + 1.3) / 1.7	25.8	29.0	34.5	24.4	28.7	27.5	23.6	18.2	26.5
	(B)	✓	✗	(0.4 + 1.3) / 1.7	30.8	33.5	41.6	29.4	33.2	33.3	28.1	20.2	31.3
	(C)	✓	✓	(0.4 + 1.3) / 1.7	32.0	35.3	42.5	31.2	36.0	38.4	28.8	21.6	33.2
Supervised End-to-End Speech Translation													
CHIMERA (Han et al., 2021)		✓	0.15	27.1	30.6	35.6	25.0	29.2	30.2	24.0	17.4	27.4	
STEMM (Fang et al., 2022)		✓	0.15	28.7	31.0	37.4	25.8	30.5	31.7	24.5	17.8	28.4	
CONST (Ye et al., 2022)		✓	0.15	28.3	32.0	38.3	27.2	31.7	33.1	25.6	18.9	29.4	
STPT (Tang et al., 2022)		✓	0.15	29.2	33.1	39.7	-	-	-	-	-	-	
SPEECHUT (Zhang et al., 2022b)		✓	0.15	30.1	33.6	41.4	-	-	-	-	-	-	
CMOT (Zhou et al., 2023)		✓	0.15	29.0	32.8	39.5	27.5	32.1	33.5	26.0	19.2	30.0	
SIAMESE-PT (Le et al., 2023)		✓	0.25	27.9	31.8	39.2	27.7	31.7	34.2	27.0	18.5	29.8	
CRESS (Fang and Feng, 2023)		✓	0.15	29.4	33.2	40.1	27.6	32.2	33.6	26.4	19.7	30.3	
SIMREGCR (Gao et al., 2024)		✓	0.15	29.2	33.0	40.0	28.2	32.7	34.2	26.7	20.1	30.5	
LST-7B (Zhang et al., 2023b)		✓	7 / 7.3	29.2	33.1	40.8	-	-	-	-	-	-	
LST-13B (Zhang et al., 2023b)		✓	13 / 13.3	30.4	35.3	41.6	-	-	-	-	-	-	
This Work Experiments on Compression with ZEROSWOT-MEDIUM (B) from Table 5.10													
Without Compression		✓	✗	0.35 / 0.95	27.6	33.0	36.4	27.4	31.5	32.0	25.2	18.2	28.9
Length Adaptor (ds × 4)		✓	✗	0.35 / 0.95	27.8	32.4	36.7	27.9	31.4	31.6	25.4	18.3	28.9
Character-level Compr.		✓	✗	0.35 / 0.95	27.9	32.6	36.9	27.8	31.8	31.4	25.4	18.1	29.0
Compr. Adapter w/ Word Tok.		✓	✗	0.35 / 0.95	26.4	31.0	33.7	25.7	29.6	30.3	24.1	17.3	27.2
Compr. Adapter w/ Subword Tok.		✓	✗	0.35 / 0.95	27.3	32.5	35.4	26.4	31.0	31.6	25.0	18.3	28.4

Table B.3: Extended Results on MuST-C tst-COMMON.

Models	Type	ASR Data	Params	WER (↓)									Average
				deu	spa	fra	ita	nld	por	ron	rus		
WAV2VEC 2.0	CTC	Common Voice	0.3	42.5	42.5	42.5	42.6	42.6	42.5	42.5	42.5	42.5	
WAV2VEC 2.0	CTC	MuST-C	0.3	21.4	21.4	21.4	21.4	21.4	21.5	21.5	21.5	21.4	
WAV2VEC 2.0 - TRANSFORMER	AED	Common Voice	0.4	20.4	20.9	20.5	20.7	20.3	21.1	20.8	20.6	20.7	
WAV2VEC 2.0 - TRANSFORMER	AED	MuST-C	0.4	9.8	10.1	9.8	10.0	9.6	10.6	10.2	9.9	10.0	

Table B.4: Word-error-rate (WER) ASR Results on MuST-C eng→xxx tst-COMMON. Note that these are English ASR results, essentially evaluated on the audio-transcription pairs of each eng→xxx test set. AED stands for attention-based encoder-decoder.

Models	Params	ara	cat	cym	deu	est	fas	ind	jpn	lav	mon	slv	swe	tam	tur	zho	Average
<i>Machine Translation</i>																	
<i>Original Models, used in B</i>																	
NLLB-MEDIUM	0.6	20.0	39.0	26.3	35.5	23.4	15.7	39.6	21.8	14.8	10.4	30.3	41.1	20.2	21.1	34.8	26.3
NLLB-LARGE	1.3	23.3	43.5	33.5	37.9	27.9	16.6	41.9	23.0	20.0	13.1	35.1	43.8	21.7	23.8	37.5	29.5
<i>This Work Finetuned in CoVoST MT, used in C</i>																	
NLLB-MEDIUM-FT	0.6	28.5	46.3	35.5	37.1	31.5	29.2	45.2	38.4	29.1	22.0	37.7	45.4	29.9	23.0	46.7	35.0
NLLB-LARGE-FT	1.3	29.9	47.8	35.6	38.8	32.7	29.9	46.4	39.5	29.9	21.7	39.3	46.8	31.0	24.4	48.2	36.1
<i>This Work Zero-shot End-to-End Speech Translation</i>																	
ZEROSWOT-MEDIUM (B)	1.0	17.6	32.5	18.0	29.9	20.4	16.3	32.4	32.0	13.3	10.0	25.2	34.4	17.8	15.6	30.5	23.1
ZEROSWOT-MEDIUM (C)	1.0	24.4	38.7	28.8	31.2	26.2	26.0	36.0	46.0	24.8	19.0	31.6	37.8	24.4	18.6	39.0	30.2
ZEROSWOT-LARGE (B)	1.7	19.8	36.1	22.6	31.8	23.6	16.8	34.2	33.6	17.5	11.8	28.9	36.8	19.1	17.5	32.2	25.5
ZEROSWOT-LARGE (C)	1.7	25.7	40.0	29.0	32.8	27.2	26.6	37.1	47.1	25.7	18.9	33.2	39.3	25.3	19.8	40.5	31.2
<i>Supervised End-to-End Speech Translation</i>																	
XLS-R-0.3B	0.3	16.3	28.7	29.1	23.6	19.6	19.0	27.4	36.9	19.3	13.2	22.4	29.1	15.6	15.0	33.5	23.2
XLS-R-1B	1.0	19.2	32.1	31.8	26.2	22.4	21.3	30.3	39.9	22.0	14.9	25.4	32.3	18.1	17.1	36.7	26.0
XLS-R-2B	2.0	20.7	34.2	33.8	28.3	24.1	22.9	32.5	41.5	23.5	16.2	27.6	34.5	19.8	18.6	38.5	27.8
SEAMLESSM4T-V1-MEDIUM	1.2	20.8	37.3	29.9	31.4	23.3	17.2	34.8	37.5	19.5	12.9	29.0	37.3	18.9	19.8	30.0	26.6
SEAMLESSM4T-V1-LARGE	2.3	24.5	41.6	33.6	35.9	28.5	19.3	39.0	39.4	23.8	15.7	35.0	42.5	22.7	23.9	33.1	30.6
SEAMLESSM4T-V2-LARGE	2.3	25.4	43.6	35.5	37.0	29.3	19.2	40.2	39.7	24.8	16.4	36.2	43.7	23.4	24.7	35.9	31.7

Table B.5: Extended Results (BLEU scores) on CoVoST 2 test. All results are reported in BLEU. Results of previous works on supervised ST from XLS-R (Babu et al., 2022), SEAMLESSM4T-V1 (Seamless Communication, 2023a), and SEAMLESSM4T-V2 (Seamless Communication, 2023b).

C Supplement to “Character-level Modeling for Enhanced Modality Transfer”

C.1 Data and Languages

In Tables C.1 and C.2, we provide details about the languages and the amounts of data used in our experiments. The numbers for the MT data are in millions and indicate the amount remaining after filtering with BLASER-2.0-QE (Dale and Costa-jussà, 2024). The numbers for ASR are in thousands. The resourcefulness label (low, medium, or high) of each language is defined separately for each modality and indicates which of the three percentiles of the data distribution the language belongs to. An empty resourcefulness level means that no data are available; consequently, when evaluated, our model operates in a fully zero-shot setting.

Language	Code	FL+	MMS	Family	Subgrouping	Script	MT		ASR	
							#M / Res		#K / Res	
Aragonese	arg_Latn	✓	✗	Indo-European	Italic	Latin	0.1	new	0.0	-
Asturian	ast_Latn	✗	✓	Indo-European	Italic	Latin	0.2	low	0.4	-
Awadhi	awa_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.4	low	0.0	-
South Azerbaijani	azb_Arab	✗	✓	Turkic	Common Turkic	Arabic	0.3	low	0.0	-
North Azerbaijani	azj_Latn	✗	✓	Turkic	Common Turkic	Latin	9.4	med	0.1	low
Bashkir	bak_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	1.7	med	119.2	-
Belarusian	bel_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	11.5	med	347.6	high
Bhojpuri	bho_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.6	low	0.0	-
Bosnian	bos_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	21.8	high	0.0	-
Boro	brx_Deva	✓	✗	Sino-Tibetan	Tibeto-Burman	Devanagari	0.1	new	0.0	-
Bulgarian	bul_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	39.3	high	4.8	med
Catalan	cat_Latn	✗	✓	Indo-European	Italic	Latin	10.1	med	1146.2	high
Valencian	cat_Latn_vale1252	✓	✗	Indo-European	Italic	Latin	0.0	new	0.0	-
Czech	ces_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	52.3	high	20.1	med
Chuvash	chv_Cyrl	✓	✓	Turkic	Oghuric	Cyrillic	1.2	new	1.4	-
Central Kurdish	ckb_Arab	✗	✓	Indo-European	Iranian	Arabic	1.7	med	7.7	-
Crimean Tatar	crh_Latn	✗	✓	Turkic	Common Turkic	Latin	0.2	low	0.0	-
Dogri	dgo_Deva	✓	✓	Indo-European	Indo-Aryan	Devanagari	0.1	new	0.0	-
Greek	ell_Grek	✗	✓	Indo-European	Graeco-Phrygian	Greek	52.6	high	1.9	low
Estonian	est_Latn	✗	✓	Uralic	Finnic	Latin	16.9	high	3.2	med
Finnish	fin_Latn	✗	✓	Uralic	Finnic	Latin	32.6	high	2.1	low
French	fra_Latn	✗	✓	Indo-European	Italic	Latin	144.9	high	558.1	high
Friulian	fur_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-
Galician	glg_Latn	✗	✓	Indo-European	Italic	Latin	6.5	med	25.2	high
Konkani	gom_Deva	✓	✗	Indo-European	Indo-Aryan	Devanagari	0.1	new	0.0	-
Hindi	hin_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	35.6	high	4.7	med
Chhattisgarhi	hne_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Croatian	hrv_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	17.1	high	0.0	-
Hungarian	hun_Latn	✗	✓	Uralic	–	Latin	32.6	high	37.1	high
Italian	ita_Latn	✗	✓	Indo-European	Italic	Latin	95.7	high	169.8	high
Karakalpak	kaa_Latn	✓	✓	Turkic	Kipchak	Latin	0.3	new	0.0	-
Kashmiri	kas_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.1	low	0.0	-
Kazakh	kaz_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	5.6	med	0.5	low
Halh Mongolian	khk_Cyrl	✗	✓	Mongolic-Khitani	Mongolic	Cyrillic	0.5	low	2.2	low
Kyrgyz	kir_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	2.7	med	1.8	-
Northern Kurdish	kmr_Latn	✗	✓	Indo-European	Iranian	Latin	0.7	med	5.1	-
Ligurian	lij_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	1.6	-
Lithuanian	lit_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	14.0	high	7.3	med
Lombard	lmo_Latn	✗	✗	Indo-European	Italic	Latin	0.3	low	0.0	-
Latgalian	ltg_Latn	✗	✗	Indo-European	Balto-Slavic	Latin	0.3	low	3.7	-
Standard Latvian	lvs_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	2.8	med	11.4	med
Magahi	mag_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Maithili	mai_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.4	low	0.0	-
Marathi	mar_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	11.8	med	2.2	low

Table C.1: Language details (Part 1 of 2). FL+: FLORES+, Res: Resourcefulness level.

Language	Code	FL+	MMS	Family	Subgrouping	Script	MT		ASR	
							#M / Res		#K / Res	
Meadow Mari	mhr_Cyrl	✓	✓	Uralic	Finno-Ugric	Cyrillic	0.4	new	185.9	-
Macedonian	mkd_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	6.8	med	1.7	low
Erzya	myv_Cyrl	✓	✓	Uralic	Mordvinic	Cyrillic	0.1	new	1.2	-
Nepali	npi_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	4.5	med	0.3	low
Occitan	oci_Latn	✗	✓	Indo-European	Italic	Latin	0.2	low	0.3	low
Aranese	oci_Latn_aran1260	✓	✗	Indo-European	Italic	Latin	0.0	new	0.0	-
Southern Pashto	pbt_Arab	✗	✗	Indo-European	Iranian	Arabic	0.9	med	0.0	-
Western Persian	pes_Arab	✗	✓	Indo-European	Iranian	Arabic	15.0	high	28.9	high
Polish	pol_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	60.4	high	20.7	med
Portuguese	por_Latn	✗	✓	Indo-European	Italic	Latin	116.8	high	22.0	med
Dari	prs_Arab	✗	✗	Indo-European	Iranian	Arabic	0.9	med	0.0	-
Romanian	ron_Latn	✗	✓	Indo-European	Italic	Latin	59.9	high	5.1	med
Russian	rus_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	89.0	high	26.4	high
Sanskrit	san_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Sicilian	scn_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-
Slovak	slk_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	29.7	high	3.3	med
Slovenian	slv_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	20.8	high	1.4	low
Sindhi	snd_Deva	✓	✓	Indo-European	Indo-Aryan	Devanagari	0.0	new	0.0	-
Spanish	spa_Latn	✗	✓	Indo-European	Italic	Latin	202.0	high	336.8	high
Sardinian	srd_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.5	-
Serbian	srp_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	5.5	med	1.9	low
Silesian	szl_Latn	✗	✗	Indo-European	Balto-Slavic	Latin	0.4	low	0.0	-
Tatar	tat_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	2.1	med	9.3	-
Tajik	tgk_Cyrl	✗	✓	Indo-European	Iranian	Cyrillic	1.1	med	0.0	-
Turkmen	tuk_Latn	✗	✓	Turkic	Common Turkic	Latin	0.6	low	0.8	-
Turkish	tur_Latn	✗	✓	Turkic	Common Turkic	Latin	47.4	high	35.1	high
Tuvan	tyv_Cyrl	✓	✗	Turkic	Common Turkic	Cyrillic	0.2	new	0.0	-
Uyghur	uig_Arab	✗	✓	Turkic	Common Turkic	Arabic	0.8	med	9.7	-
Ukrainian	ukr_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	12.2	med	25.1	med
Northern Uzbek	uzn_Latn	✗	✓	Turkic	Common Turkic	Latin	4.1	med	48.5	high
Venetian	vec_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-

Table C.2: Language details (Part 2 of 2). FL+: FLORES+, Res: Resourcefulness level.

C.2 Additional Results

Detailed Text Translation Results. In Table C.3 we present the per-language text translation results for the models of Table 6.3, including NLLB-1.3B (NLLB, 2024), the original SONAR (Duquenne et al., 2023), as well as our SONAR and CHARSONAR models trained on 63 (only base) and 75 (base and new) languages. Overall, CHARSONAR models outperform their subword-based counterparts, and for the majority of languages, they additionally surpass NLLB-1.3B.

Cross-lingual Similarity Results. Table 6.8 provides the cross-lingual similarity error rates for our text encoders. It compliments text translation results of Table 6.3, and confirms our findings, that CHARSONAR outperforms the subword-based models, especially for low-resource and new languages.

Text Results Grouped by Linguistic Family and Script. In Tables C.5 and C.6, we present the per linguistic family and script results of our text encoders in translation and cross-lingual similarity search. We observe that for base languages that are already supported by the teacher SONAR (Duquenne et al., 2023), the CHARSONAR encoder outperforms the subword-based encoders in all categories, apart from the single group that contains the Greek language, and only in cross-lingual similarity search. For the new languages, we notice that our method performs better in all categories for translation, but the subword-based model is better for the Turkic and Uralic families, and Cyrillic script in cross-lingual similarity search.

Data Filtering Thresholds. In Figure C.1 we present our ablation for deciding the BLASER-2.0-QE (Dale and Costa-jussà, 2024) data filtering threshold. To speed-up experimentation and use less data, we filtered with 4.5 for the initial exploration, but for the main experiments we used a threshold of 4.

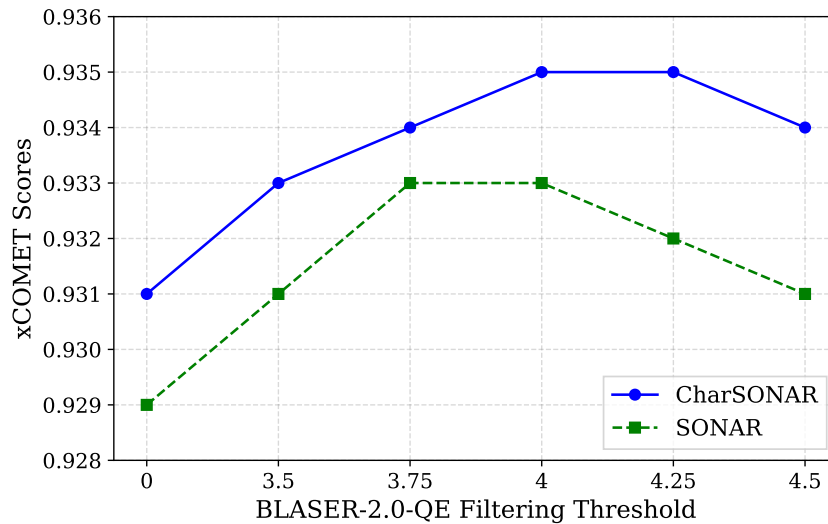


Figure C.1: xCOMET scores vs. BLASER-2.0-QE filtering threshold for CHARSONAR and SONAR in FLORES dev. Results with the ablation group of 15 languages including the ones from the Uralic family and the ones using the Cyrillic script. xCOMET scores are average of xxx→eng for all the 15 languages in the group.

Detailed Speech Translation Results. In Table C.7, we present the per-language speech translation results for the models of Table 6.7. We also add results from the original SONAR Speech Encoders (Duquenne et al., 2023), which were excluded from the main table since they do not support all the languages with which we experiment. Furthermore, Table C.7 includes more than 33 languages, since for ease of presentation, in the main results we presented the ones that were both supported by WHISPER and our models. We indicate the languages not taken into account for the results of Table 6.7. Our analytic results confirm the efficacy of our method, where with the Dual Adapter trained on Common Voice and FLEURS, outperforms all other models in the majority of the languages. Furthermore we also notice that for several high-resource languages, a cascade of WHISPER and CHARSONAR is very competitive. On the other hand, the previous state-of-the-art model, SEAMLESSM4T, obtains the best results in only 4 languages.

	NLLB	SONAR	SONAR-63	CHARSONAR-63	SONAR-75	CHARSONAR-75
arg_Latn	0.867 [†]	0.860 [†]	0.910 [†]	<u>0.912[†]</u>	0.931	0.917
ast_Latn	0.918	0.909	0.920	0.928	0.920	0.927
awa_Deva	0.918	0.861	0.894	0.896	0.898	0.898
azb_Arab	0.632	0.513	0.698	0.698	0.681	0.696
azj_Latn	0.910	0.741	0.863	0.868	0.865	0.870
bak_Cyrl	0.916	0.900	0.915	0.920	0.916	0.918
bel_Cyrl	0.928	0.905	0.922	0.926	0.920	0.926
bho_Deva	0.907	0.879	0.902	0.908	0.904	0.906
bos_Latn	0.961	0.953	0.960	0.960	0.959	0.960
brx_Deva*	0.208 [†]	0.203 [†]	<u>0.217[†]</u>	0.216 [†]	0.841	0.851
bul_Cyrl	0.954	0.952	0.952	0.953	0.952	0.954
cat_Latn	0.954	0.949	0.952	0.956	0.952	0.956
cat_Latn_vale1252	0.903 [†]	0.886 [†]	0.952 [†]	<u>0.955[†]</u>	0.952 [†]	0.955 [†]
ces_Latn	0.954	0.953	0.955	0.956	0.955	0.956
chv_Cyrl	0.240 [†]	0.251 [†]	<u>0.274[†]</u>	0.268 [†]	0.851	0.855
ckb_Arab	0.881	0.872	0.882	0.889	0.884	0.890
crh_Latn	0.907	0.890	0.909	0.920	0.914	0.918
dgo_Deva*	<u>0.606[†]</u>	0.567 [†]	0.575 [†]	0.590 [†]	0.895	0.899
ell_Grek	0.940	0.932	0.938	0.938	0.937	0.939
est_Latn	0.940	0.935	0.942	0.945	0.942	0.945
fin_Latn	0.942	0.936	0.945	0.946	0.944	0.947
fra_Latn	0.965	0.961	0.961	0.961	0.959	0.961
fur_Latn	0.931	0.930	0.937	0.938	0.936	0.938
glg_Latn	0.957	0.951	0.954	0.957	0.954	0.957
gom_Deva*	0.511 [†]	0.477 [†]	0.607 [†]	<u>0.635[†]</u>	0.869	0.880
hin_Deva	0.939	0.935	0.934	0.935	0.934	0.938
hne_Deva	0.911	0.902	0.913	0.915	0.912	0.916
hrv_Latn	0.947	0.949	0.954	0.954	0.954	0.954
hun_Latn	0.946	0.940	0.945	0.948	0.945	0.947
ita_Latn	0.957	0.954	0.951	0.954	0.952	0.954
kaa_Latn	0.349 [†]	0.359 [†]	0.698 [†]	<u>0.767[†]</u>	0.917	0.924
kas_Deva	0.657	0.600	0.674	0.704	0.681	0.710
kaz_Cyrl	0.918	0.908	0.912	0.917	0.909	0.917
khk_Cyrl	0.852	0.838	0.863	0.873	0.865	0.876
kir_Cyrl	0.908	0.898	0.911	0.915	0.909	0.915
kmr_Latn	0.784	0.776	0.789	0.804	0.789	0.800
lij_Latn	0.907	0.896	0.912	0.914	0.910	0.914
lit_Latn	0.930	0.921	0.933	0.938	0.934	0.937
lmo_Latn	0.866	0.833	0.884	0.900	0.885	0.898
ltg_Latn	0.888	0.866	0.900	0.917	0.900	0.916
lvs_Latn	0.928	0.915	0.932	0.938	0.932	0.937
mag_Deva	0.931	0.927	0.926	0.935	0.929	0.930
mai_Deva	0.930	0.881	0.907	0.905	0.906	0.906
mar_Deva	0.929	0.920	0.920	0.925	0.921	0.927
mhr_Cyrl*	0.262 [†]	0.278 [†]	0.268 [†]	<u>0.307[†]</u>	0.896	0.901
mkd_Cyrl	0.946	0.942	0.946	0.952	0.947	0.951
myv_Cyrl	0.245 [†]	0.243 [†]	0.251 [†]	<u>0.259[†]</u>	0.851	0.846
npi_Deva	0.926	0.881	0.900	0.901	0.900	0.901
oci_Latn	0.956	0.952	0.958	0.958	0.957	0.959
oci_Latn_aran1260	0.505 [†]	0.500 [†]	0.569 [†]	<u>0.576[†]</u>	0.566 [†]	0.571 [†]
pbt_Arab	0.866	0.855	0.870	0.872	0.872	0.874
pes_Arab	0.924	0.918	0.922	0.926	0.925	0.926
pol_Latn	0.950	0.946	0.946	0.949	0.948	0.948
por_Latn	0.963	0.960	0.961	0.962	0.960	0.962
prs_Arab	0.901	0.900	0.908	0.910	0.908	0.910
ron_Latn	0.964	0.961	0.963	0.964	0.964	0.964
rus_Cyrl	0.950	0.942	0.944	0.946	0.944	0.946
san_Deva	0.749	0.702	0.731	0.737	0.732	0.742
scn_Latn	0.896	0.877	0.904	0.915	0.905	0.914
slk_Latn	0.955	0.951	0.953	0.956	0.952	0.955
slv_Latn	0.944	0.943	0.948	0.951	0.948	0.951
snd_Deva*	0.466 [†]	0.464 [†]	<u>0.512[†]</u>	0.497 [†]	0.860	0.869
spa_Latn	0.950	0.949	0.946	0.948	0.946	0.948
srd_Latn	0.909	0.902	0.917	0.918	0.915	0.916
srp_Cyrl	0.949	0.943	0.950	0.954	0.950	0.954
szl_Latn	0.930	0.920	0.934	0.942	0.933	0.940
tat_Cyrl	0.927	0.915	0.925	0.927	0.925	0.928
tgk_Cyrl	0.915	0.903	0.915	0.925	0.916	0.923
tuk_Latn	0.893	0.882	0.898	0.910	0.904	0.912
tur_Latn	0.946	0.940	0.943	0.946	0.945	0.945
tyv_Cyrl	0.281 [†]	0.313 [†]	0.366 [†]	<u>0.381[†]</u>	0.880	0.884
uig_Arab	0.863	0.853	0.859	0.868	0.859	0.867
ukr_Cyrl	0.948	0.942	0.945	0.948	0.946	0.949
uzn_Latn	0.931	0.914	0.918	0.924	0.920	0.926
vec_Latn	0.920	0.907	0.931	0.933	0.930	0.936

Table C.3: Text Translation xCOMET scores in FLORES+ devtest xxx→eng. * indicates translation is evaluated on dev split (some newly added languages in FLORES+ do not have a devtest split). † indicates that the result is zero-shot. Underlined is the best among the zero-shot for each language, if any. In **bold** is the best for supervised results for each language, if any.

Model	Low (21)	Med (21)	High (21)	All (63)	New (12)
<i>Previous Works (trained on the 200 base languages)</i>					
NLLB-1.3B (NLLB, 2024)	-	-	-	-	-
SONAR (Duquenne et al., 2023)	13.1	10.1	7.3	10.2	52.7 [†]
<i>This work (trained on 75 languages; 63 base + 12 new)</i>					
SONAR-75	9.0	7.7	5.8	7.5	12.6
CHARSONAR-75	8.4	7.2	5.5	7.0	12.3
Δ	0.6	0.5	0.3	0.5	0.3
<i>This work (trained only on 63 base languages)</i>					
SONAR-63	8.8	7.7	5.8	7.5	42.9 [†]
CHARSONAR-63	8.3	7.2	5.5	7.0	<u>42.2[†]</u>
Δ	0.5	0.5	0.3	0.5	0.7

Table C.4: Cross-lingual similarity search xsim++ (↓) error rates on FLORES devtest (xxx→eng), grouped by resourcefulness level (Low/Med/High) and New languages. † indicates zero-shot evaluation. In **bold** is the best overall and underlined is the best zero-shot. Δ refers to the difference between CHARSONAR-N and SONAR-N models.

	Grouped by Linguistic Family				Grouped by Script		
	Indic	Romance	Turkic	Uralic	Devanagari	Latin	Cyrillic
# Languages	4	3	3	2	4	4	4
xCOMET (↑)							
SONAR (Duquenne et al., 2023)	0.428	0.749	0.305	0.260	0.428	0.651	0.269
SONAR-75	0.866	0.813	0.882	0.869	0.866	0.839	0.867
CHARSONAR-75	0.875	0.814	0.888	0.874	0.875	0.842	0.871
xsim++ (↓)							
SONAR (Duquenne et al., 2023)	53.8	29.3	63.5	70.2	53.8	36.1	68.6
SONAR-75	8.7	21.6	9.6	10.5	8.7	17.7	11.0
CHARSONAR-75	8.2	20.9	10.1	10.8	8.2	16.9	11.8

Table C.5: Text translation (xCOMET ↑) and text retrieval (xsim++ ↓) results per linguistic family and script for the 12 newly added languages. Results in FLORES+ devtest xxx→eng.

	Grouped by Linguistic Family								Grouped by Script				
	Balto-Slavic	Romance	Turkic	Indic	Iranian	Uralic	Mongolic	Greek	Latin	Cyrillic	Devanagari	Arabic	Greek
# Languages	16	15	11	10	6	3	1	1	34	12	10	6	1
xCOMET (↑)													
SONAR	0.934	0.926	0.850	0.849	0.871	0.937	0.838	0.932	0.917	0.916	0.849	0.818	0.932
SONAR-75	0.942	0.935	0.884	0.870	0.881	0.944	0.864	0.936	0.929	0.925	0.870	0.852	0.936
CHARSONAR-75	0.946	0.940	0.892	0.877	0.887	0.946	0.876	0.939	0.934	0.930	0.877	0.860	0.939
xsim++ (↓)													
SONAR	8.1	8.0	13.4	13.5	10.7	7.1	13.3	9.2	8.3	9.7	13.5	16.1	9.2
SONAR-75	6.2	5.5	10.0	9.7	8.2	5.7	11.0	6.7	6.1	7.7	9.7	11.6	6.7
CHARSONAR	5.8	5.2	9.4	9.1	7.7	5.3	10.1	6.9	5.7	7.1	9.1	11.0	6.9

Table C.6: Text translation (xCOMET ↑) and text retrieval (xsim++ ↓) results per linguistic family and script for the 63 base training languages. Results in FLORES devtest xxx→eng.

Language	E2E ST			CHARSONAR Cascades		S-CHARSONAR (CV)			+ FLEURS	
	SONAR	WHISPER	SEAMLESSM4T	w/ MMS	w/ WHISPER	MLP	V-ALIGN	DUAL	V-ALIGN	DUAL
ast_Latn*	-	-	0.752	0.821	-	0.204	0.816	0.223	0.840	0.849
azj_Latn	-	0.642	0.837	0.762	0.777	0.197	0.788	0.194	0.788	0.764
bel_Cyrl	0.823	0.676	0.885	0.864	0.842	0.856	0.864	0.868	0.874	0.886
bos_Latn	0.882	0.829	0.919	0.899	0.911	-	-	-	-	-
bul_Cyrl	0.852	0.816	0.902	0.884	0.891	0.879	0.891	0.898	0.892	0.907
cat_Latn	0.886	0.874	0.929	0.893	0.929	0.903	0.900	0.922	0.913	0.930
ces_Latn	0.873	0.811	0.910	0.889	0.904	0.888	0.890	0.904	0.902	0.911
ckb_Arab*	-	-	0.766	0.786	-	0.692	0.800	0.798	0.802	0.818
ell_Grek	-	0.747	0.868	0.860	0.874	0.216	0.866	0.866	0.872	0.889
est_Latn	0.805	0.629	0.898	0.908	0.902	0.877	0.910	0.912	0.918	0.914
fin_Latn	0.799	0.768	0.887	0.900	0.926	0.229	0.902	0.904	0.906	0.905
fra_Latn	0.870	0.891	0.910	0.884	0.935	0.888	0.906	0.914	0.914	0.922
glg_Latn	-	0.830	0.917	0.897	0.908	0.859	0.908	0.913	0.906	0.910
hin_Deva	0.770	0.746	0.856	0.856	0.834	0.854	0.876	0.875	0.880	0.882
hrv_Latn*	0.866	0.816	0.895	0.903	0.912	-	-	-	-	-
hun_Latn	-	0.724	0.878	0.867	0.886	0.870	0.866	0.882	0.884	0.892
ita_Latn	0.892	0.898	0.927	0.915	0.943	0.914	0.924	0.926	0.932	0.939
kaz_Cyrl	-	0.349	0.846	0.844	0.774	0.198	0.849	0.198	0.866	0.876
khk_Cyrl	-	0.207	0.748	0.731	0.342	0.229	0.732	0.213	0.763	0.764
kir_Cyrl*	-	-	0.854	0.837	-	0.197	0.855	0.853	0.861	0.865
lit_Latn	0.766	0.579	0.832	0.880	0.853	0.858	0.881	0.885	0.884	0.897
lvs_Latn	0.848	0.589	0.885	0.899	0.888	0.879	0.905	0.909	0.908	0.910
mar_Deva	0.734	0.503	0.821	0.812	0.684	0.734	0.836	0.843	0.840	0.846
mkd_Cyrl	0.887	0.808	0.917	0.919	0.912	0.246	0.927	0.925	0.928	0.928
npi_Deva	0.675	0.538	0.826	0.790	0.652	0.196	0.810	0.798	0.801	0.799
oci_Latn	-	0.483	0.568	0.747	0.583	0.202	0.707	0.707	0.795	0.805
pes_Arab	0.810	0.666	0.887	0.867	0.851	0.875	0.890	0.884	0.887	0.899
pol_Latn	0.860	0.856	0.893	0.888	0.923	0.876	0.888	0.899	0.898	0.908
por_Latn	0.878	0.906	0.897	0.902	0.941	0.885	0.908	0.917	0.918	0.922
ron_Latn	0.856	0.867	0.909	0.895	0.919	0.855	0.900	0.904	0.906	0.905
rus_Cyrl	0.878	0.893	0.912	0.883	0.934	0.883	0.898	0.908	0.903	0.910
slk_Latn	0.885	0.822	0.914	0.911	0.924	0.876	0.909	0.917	0.919	0.921
slv_Latn	0.843	0.672	0.879	0.871	0.852	0.201	0.871	0.200	0.883	0.875
snd_Deva*	0.360	0.360	0.443	0.698	0.423	-	-	-	-	-
spa_Latn	0.888	0.893	0.908	0.899	0.934	0.912	0.917	0.920	0.922	0.930
srp_Cyrl	0.891	0.856	0.924	0.924	0.929	0.235	0.922	0.913	0.927	0.928
tgk_Cyrl*	-	0.523	0.858	0.874	0.658	-	-	-	-	-
tur_Latn	0.743	0.827	0.888	0.878	0.924	0.868	0.885	0.903	0.899	0.909
ukr_Cyrl	0.858	0.865	0.912	0.890	0.931	0.887	0.903	0.904	0.911	0.915
uzn_Latn	0.736	0.326	0.846	0.753	0.527	0.798	0.839	0.843	0.846	0.854

Table C.7: Speech Translation xCOMET scores on FLEURS test xxx→eng. In **bold** is the best for each language. The 6 languages with * were not part of the main results of Table 6.7, since they were not supported either by our models or by WHISPER. **E2E ST**: Original SONAR w2v-BERT Speech Encoders with SONAR decoder. (Duquenne et al., 2023), WHISPER-LARGE-V3 (Radford et al., 2023), SEAMLESSM4T-LARGE-V2 (Barrault et al., 2025). **S-CHARSONAR (CV)**: Our proposed model trained on Common Voice Data with the three different Adapters. **(+ FLEURS)**: Our proposed model trained on Common Voice and FLEURS.

D Supplement to “Towards Prosody-aware Evaluation: Speech Is More Than Words”

D.1 Examples for In-context Learning

In Tables D.1 to D.3, we present examples that were used for in-context learning when generating new examples with GPT-4 (§7.4.1).

1.1 Contrastive Stress	
Sentence x	She didn’t give the book to John.
Prosody p^a	She didn’t give the *BOOK* to John.
Interpretation i^a	Something else was given to John.
Prosody p^b	She didn’t give the book to *JOHN*.
Interpretation i^b	The book was given to someone else.
1.2 New vs. Given Information	
Sentence x	The committee decided to postpone the meeting.
Prosody p^a	The *COMMITTEE* decided to postpone the meeting.
Interpretation i^a	Given: Someone decided to postpone the meeting; New: It was the committee who decided.
Prosody p^b	The committee decided to *POSTPONE* the meeting.
Interpretation i^b	Given: The committee decided something; New: The decision was to postpone it.
1.3 Relational vs. Descriptive Adjectives	
Sentence x	They are German teachers.
Prosody p^a	They are *GERMAN* teachers.
Interpretation i^a	Teachers who teach the German language. (Relational)
Prosody p^b	They are German *TEACHERS*.
Interpretation i^b	Teachers who are German. (Descriptive)
1.4 Focus-Sensitive Operators	
Sentence x	I only introduced John to Maria at yesterday’s party.
Prosody p^a	I only introduced *JOHN* to Maria at yesterday’s party.
Interpretation i^a	John was the only person I introduced to Maria.
Prosody p^b	I only introduced John to *MARIA* at yesterday’s party.
Interpretation i^b	Maria was the only person I introduced John to.

Table D.1: Examples in the category *Sentence Stress* that were used for in-context learning.

2.1 Direct vs. Indirect Statements	
Sentence x	Alex announced Jamie will meet the manager.
Prosody p ^a	Alex *ANNOUNCED* Jamie will meet the manager.
Interpretation i ^a	(Direct Statement)
Prosody p ^b	Alex announced Jamie will meet the manager.
Interpretation i ^b	(Indirect Statement)
2.2 Restrictive vs. Non-Restrictive Phrases	
Sentence x	The students who were talking were sent out.
Prosody p ^a	The students who were *TALKING* were sent out.
Interpretation i ^a	Only the students who were talking were actually sent out. (Restrictive)
Prosody p ^b	The *STUDENTS* who were talking were sent out.
Interpretation i ^b	All students were sent out, and the fact they were talking is additional information. (Non-restrictive)
2.3 Attachment Ambiguities	
Sentence x	Paula phoned her friend from Alabama.
Prosody p ^a	Paula phoned her friend from *ALABAMA*.
Interpretation i ^a	Paula called her friend while she was in Alabama. (VP Attachment)
Prosody p ^b	Paula phoned her *FRIEND* from Alabama.
Interpretation i ^b	Paula phoned her friend who is from Alabama. (NP Attachment)
2.4 Phrasal Verbs	
Sentence x	John laughed at the party.
Prosody p ^a	John *LAUGHED* at the party.
Interpretation i ^a	John laughed while he was at the party. (Literal)
Prosody p ^b	John *LAUGHED AT* the party.
Interpretation i ^b	John made fun of the party. (Idiomatic)
2.5 Complementizer vs. Parenthetical	
Sentence x	We only suspected they all knew that a burglary had been committed.
Prosody p ^a	We only *SUSPECTED* they all knew that a burglary had been committed.
Interpretation i ^a	The suspicion was that they all knew about the burglary. (Complementizer)
Prosody p ^b	We only suspected they all *KNEW* that a burglary had been committed.
Interpretation i ^b	They all knew that we only suspected that a burglary had been committed. (Parenthetical)
2.6 Modifier Scope	
Sentence x	This collar is dangerous to younger dogs and cats.
Prosody p ^a	This collar is dangerous to *YOUNGER* dogs and cats.
Interpretation i ^a	Younger refers to both dogs and cats. (Broad Scope)
Prosody p ^b	This collar is dangerous to *YOUNGER* dogs and *CATS*.
Interpretation i ^b	Younger refers only to dogs. (Narrow Scope)

Table D.2: Examples in the category *Prosodic Breaks* that were used for in-context learning.

3. Intonation Patterns	
Sentence x	You can solve this problem
Prosody p ^a	You *CAN* solve this problem.
Interpretation i ^a	Encouraging or asserting the person’s ability to solve this problem.
Prosody p ^b	You _can_ solve this problem?
Interpretation i ^b	Questioning the person’s ability to solve this problem.
4. Emotional Prosody (Happy/Sad)	
Sentence x	The surgery went as expected.
Prosody p ^a	<happy> The surgery went *AS EXPECTED*!
Interpretation i ^a	The surgery’s successful outcome aligns with hopes and predictions, leading to joy and relief.
Prosody p ^b	<sad> The surgery went _as expected_ ...
Interpretation i ^b	The expected outcome was not favorable, leading to a somber tone.
4. Emotional Prosody (Fearful/Angry)	
Sentence x	Can we talk about this later?
Prosody p ^a	<fearful> Can we... talk about this... later?
Interpretation i ^a	Indicates hesitation or fear about the topic, or the situation in general.
Prosody p ^b	<angry> Can we *TALK* about this later!?
Interpretation i ^b	Implies urgency or frustration, and a demand for immediate attention.
5. Politeness	
Sentence x	Can you move your car?
Prosody p ^a	<polite> Can you _move_ your car?
Interpretation i ^a	A polite request to move the car.
Prosody p ^b	<impolite> Can you *MOVE* your *CAR*?!
Interpretation i ^b	A rude demand to move the car, with an aggressive tone.

Table D.3: Examples in the categories *Intonation Patterns*, *Emotional Prosody*, and *Politeness*, that were used for in-context learning.

D.2 Quality Assessment for TTS Candidates

Here we present the objectives we defined for assessing the quality of the generated speech candidates for each contrastive example. The objective is applied only to candidates that had $WER = 0$ using WHISPER. If all candidates are invalid for a prosodic case, the whole example is removed. We also defined some threshold levels for the objectives after trial and error, in order to remove examples where the best candidate was below it.

Sentence Stress. We use forced-alignment with WAV2VEC 2.0 (Baevski et al., 2020) to obtain the segment for each word in the signal, and extract their loudness, pitch, and duration features. Then we define the stress level stress for a word w as the weighted sum of these three features. Finally, we select the best candidate according to a simple objective

$\text{obj}_{\text{stress}}$ that has three goals: (1) maximize the stress of the target word ($\text{stress}_{\text{tgt}}$), (2) minimize the stress of the target word of the contrastive case ($\text{stress}_{\text{foil}}$), and (3) minimize the average stress of the rest.

$$\text{stress}_w = \lambda_1 \text{loud}_w + \lambda_2 \text{pitch}_w + \lambda_3 \text{dur}_w \quad (\text{D.1})$$

$$\text{obj}_{\text{stress}} = 2 \cdot \text{stress}_{\text{tgt}} - \text{stress}_{\text{foil}} - \frac{1}{n-1} \sum_{w \neq \text{tgt}} \text{stress}_w \quad (\text{D.2})$$

where we used $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, and $\lambda_3 = 0.2$. Note that in the sentence stress examples, there is always exactly one target word in each contrastive prosodic case.

Prosodic Breaks. Likewise, after forced-alignment, we measure the duration dur of each gap l between the words in the utterance, and define a similar objective $\text{obj}_{\text{break}}$ as:

$$\text{obj}_{\text{break}} = 2 \frac{1}{|\text{tgt}|} \sum_{l \in \text{tgt}} \text{dur}_l - \frac{1}{|\text{foil}|} \sum_{l \in \text{foil}} \text{dur}_l - \frac{1}{n - |\text{tgt}|} \sum_{l \notin \text{tgt}} \text{dur}_l \quad (\text{D.3})$$

In this category, there can be 0 to 2 breaks in each prosodic case, which could be shared between the two prosodic cases. In the objective, we consider only the ones that are not common in the two cases.

Intonation Patterns. We use teacher-forcing with WHISPER to extract the punctuation probabilities given the transcription text without the ending punctuation. The probability of the sentence to be a statement is the sum of the probabilities of the tokens “.” and “!”, while the probability of a question is the probability of the token “?”. Thus, the objective $\text{obj}_{\text{inton}}$ for a statement is defined as:

$$\text{obj}_{\text{inton}} = P(. \mid \mathbf{s}, \mathbf{t}_{<n}) + P(! \mid \mathbf{s}, \mathbf{t}_{<n}) - P(? \mid \mathbf{s}, \mathbf{t}_{<n}) \quad (\text{D.4})$$

where $\mathbf{s}$ is the speech signal and $\mathbf{t}_{<n}$ are the tokens of the transcription, excluding the final one, which corresponds to the punctuation in all cases of this category. The negative objective $-\text{obj}_{\text{inton}}$ is used for a case that is a question.

Emotional Prosody. We employ an emotion classifier¹ which is based on a finetuned WAV2VEC 2.0 on the RAVDESS dataset (Livingstone and Russo, 2018), and define the

¹huggingface.co/ehcalabres/wav2vec2-lg-XLS-R-en-speech-emotion-recognition

objective as:

$$\text{obj}_{\text{emo}} = P(e_{\text{tgt}} \mid \mathbf{s}; \theta) - P(e_{\text{foil}} \mid \mathbf{s}; \theta) \quad (\text{D.5})$$

where θ are the parameters of the classifier, e_{tgt} is the target emotion label, and e_{foil} is the emotion label of the other prosodic case.

Politeness. To the best of our knowledge, there is no open-sourced audio classifier to detect politeness levels. Thus, we re-purpose the emotion classifier and define the probabilities of politeness and impoliteness as a weighted sum of the 8 available emotion classes:

$$P(\text{polite}) = \frac{\sum_e w_e P(e \mid \mathbf{s})}{\sum_e w_e} \quad (\text{D.6})$$

and similarly for impolite. We used the weighted scheme displayed in Table D.4, which was obtained by prompting GPT-4.

Emotion	Politeness	Impoliteness
Happy	0.3	-0.1
Calm	0.3	-0.2
Neutral	0.2	0.1
Surprised	0.1	0.1
Sad	0.0	0.2
Disgust	-0.1	0.3
Angry	-0.2	0.4
Fearful	-0.1	0.0

Table D.4: Weighting scheme for Politeness and Impoliteness labels based on the emotion classifier.

D.3 Data

D.3.1 Data Statistics

Table D.5 provides detailed statistics for each category and subcategory throughout the generation stages. The quality constraints of the cTTS system, where prosody was not always effectively encoded in the speech, necessitated the removal of a large percentage of

examples before the translation stage. Additionally, many examples were removed because the oracle translations for both contrastive cases were identical.

Category / Subcategory	Initial	Generated	Synthesised	Translated		
				deu	spa	jpn
Contrastive Stress (General)	200	199	183	87	76	97
Relational/Descriptive Adjectives	200	199	147	42	33	51
Contrastive Stress (Noun-Phrase)	200	199	124	37	36	39
New/Given Information	200	197	146	51	65	91
Focus-sensitive Operators	200	181	118	60	42	64
Sentence Stress	<u>1000</u>	<u>975</u>	<u>718</u>	<u>277</u>	<u>252</u>	<u>342</u>
Complementizer/Paranetical	200	200	171	59	46	73
VP/NP Attachment	200	200	66	23	18	20
Modifier Scope	200	200	200	83	107	81
Restrictive/Nonrestrictive	200	199	177	65	82	40
Direct/Indirect	200	198	154	41	25	70
Phrasal Verbs	42	42	17	5	1	5
Prosodic Breaks	<u>1042</u>	<u>1039</u>	<u>785</u>	<u>276</u>	<u>279</u>	<u>289</u>
Intonation Patterns	<u>300</u>	<u>263</u>	<u>174</u>	<u>173</u>	<u>173</u>	<u>173</u>
Sad-Happy	200	200	1	1	1	1
Neutral-Angry	200	199	185	123	111	119
Neutral-Happy	200	198	161	81	97	81
Disgust-Angry	200	198	18	4	5	3
Disgust-Sad	200	198	-	-	-	-
Neutral-Surprised	200	198	43	33	35	30
Disgust-Neutral	200	197	7	2	5	5
Happy-Angry	200	197	138	50	65	72
Sad-Surprised	200	197	3	2	2	2
Sad-Neutral	200	196	4	3	2	2
Sad-Angry	200	196	5	1	4	4
Disgust-Surprised	200	196	4	2	2	1
Disgust-Happy	200	195	10	5	7	6
Happy-Surprised	200	195	52	34	27	21
Angry-Surprised	200	193	68	32	34	30
Emotional Prosody	<u>3000</u>	<u>2953</u>	<u>699</u>	<u>373</u>	<u>397</u>	<u>377</u>
Politeness	<u>400</u>	<u>375</u>	<u>387</u>	<u>212</u>	<u>193</u>	<u>206</u>
Total	5742	5605	2763	1311	1294	1387

Table D.5: Number of Examples by Category and Subcategory throughout the generation pipeline.

D.3.2 Overly Explanatory Examples

Table D.6 presents two examples where GPT-4, acting as an oracle translator (§7.4.2), proposed overly explanatory translations in the emotional prosody category. Both are in line with the emotion of the speaker, but they contain new information not originally present in the source. These were removed during the filtering stage due to the excessive word-length ratio between the two cases.

Example 1: <i>This will only take a minute.</i>		
A	(neutral)	Das dauert nur eine Minute. (This will only take a minute.)
B	(angry)	Das dauert nur eine Minute, also machen Sie keinen Aufstand. (This will only take a minute so don't make a fuzz about it.)
Example 2: <i>Our case was dismissed.</i>		
A	(neutral)	Unser Fall wurde abgewiesen. (Our case was dismissed.)
B	(sad)	Unser Fall wurde abgewiesen und das macht mich fassungslos. (Our case was dismissed which is just perplexing.)

Table D.6: Examples of overly explanatory translations proposed by GPT-4. These were filtered out of the final benchmark.

D.4 Evaluated Speech Translation Models

This section details the model families and specific versions used in our experiments. We evaluated a total of 31 ST model variants. All models are available in the Hugging Face Transformers Library (Wolf et al., 2020). For inference, we used the default generation parameters and a beam size of 5.

1. **SEAMLESSM4T** (Barrault et al., 2025), with its two versions v1 (Seamless Communication, 2023a) and v2 (Seamless Communication, 2023b) are a family of unified encoder-decoder models. They are both multilingual (supporting up to 100 languages) and multimodal (accepting speech or text input/output), capable of performing ASR, TTS, MT, ST, and Speech-to-Speech translation. The architecture comprises a text encoder, text decoder, speech encoder, and speech decoder, where specific components are activated based on the input and output modalities. The text encoder-decoder is based on NLLB (NLLB, 2024), the speech encoder on

a Conformer-based (Gulati et al., 2020) w2v-BERT (Chung et al., 2021), and the speech decoder on a unit decoder (Inaguma et al., 2023) with a HiFi-GAN vocoder (Kong et al., 2020). The original version includes a Medium (1.2B)² and a Large (2.3B)³ variant, while v2 offers a LARGE variant (2.3B)⁴. For the cascaded ST setup, we first utilize the model in ASR mode, followed by the same model in MT mode.

2. **XLS-R** (Babu et al., 2022) is a multilingual E2E ST model based on WAV2VEC 2.0 (Baevski et al., 2020), trained via self-supervised learning on a large speech corpus covering 128 languages. For ST, the encoder is coupled with the decoder from MBART-50 (Liu et al., 2020a) and finetuned on paired speech-translation data. We use the following versions, which are finetuned on the English-to-15 subset of CoVoST 2 (Wang et al., 2020c): 300M⁵, 1B⁶, and 2B⁷.
3. **ZEROSWOT** (introduced in Ch. 5) is a zero-shot E2E ST model that aligns a WAV2VEC 2.0 encoder with an NLLB decoder . It achieves this by compressing the speech representation into subword units and applying Optimal Transport (Frogner et al., 2015) alignment, utilizing only ASR data. The versions used here employ an NLLB model finetuned on the text data of CoVoST 2, while the ZEROSWOT alignment was trained on Common Voice (Ardila et al., 2020). The MEDIUM version⁸ contains 1B parameters, and the LARGE version⁹ contains 1.7B parameters.
4. **SALMONN** (Tang et al., 2024) is a general-purpose audio LLM capable of various speech and audio tasks, including ST. It is built upon the Vicuna LLM (Peng et al., 2023) and incorporates two encoders: one from WHISPER (Radford et al., 2023) and one from BEATs (Chen et al., 2023b). The concatenated output representations from these encoders are processed by a Q-former (Li et al., 2023) and fed to the LLM, which is finetuned using LoRA (Hu et al., 2022). We evaluate the 7B¹⁰ and 13B¹¹ versions. To translate speech, we use the prompt recommended in the original paper: “Listen to the speech and translate it into {Target Language}”.

²huggingface.co/facebook/seamless-m4t-medium

³huggingface.co/facebook/seamless-m4t-large

⁴huggingface.co/facebook/seamless-m4t-v2-large

⁵huggingface.co/facebook/wav2vec2-xls-r-300m-en-to-15

⁶huggingface.co/facebook/wav2vec2-xls-r-1b-en-to-15

⁷huggingface.co/facebook/wav2vec2-xls-r-2b-en-to-15

⁸huggingface.co/johntsi/ZeroSwot-Medium-cv-covost2-en-to-15

⁹huggingface.co/johntsi/ZeroSwot-Large-cv-covost2-en-to-15

¹⁰huggingface.co/tsinghua-ee/SALMONN-7B

¹¹huggingface.co/tsinghua-ee/SALMONN

5. **WHISPER** → **NLLB** represents an AED-based cascade. WHISPER (Radford et al., 2023) is an encoder-decoder model performing ASR and many-to-English ST. We utilize three versions for this cascade: WHISPER-MEDIUM¹², WHISPER-LARGE¹³, and the V3 LARGE version¹⁴. While we primarily report results with v3, we also discuss v1 to avoid bias, as it was used during the filtering process. NLLB (NLLB, 2024) is a massively multilingual MT model covering 200 languages. We used two distilled versions of the 54B MoE model, distilled-600M¹⁵ and distilled-1.3B¹⁶, as well as the dense 3.3B model¹⁷. Evaluating all combinations resulted in 9 cascade variants.
6. **CTC** → **NLLB** represents a CTC-based cascade. We employ three different CTC encoders. The first is the LARGE version (300M) of WAV2VEC 2.0¹⁸, finetuned on Libri-Light (Kahn et al., 2020) and Librispeech (Panayotov et al., 2015) with self-training (Xu et al., 2021b). The second is the LARGE version (300M) of HUBERT (Hsu et al., 2021)¹⁹, finetuned on Librispeech. The third is the XL version of HUBERT²⁰ (1B parameters). Combined with the three NLLB models, this yields 9 CTC-based cascade variants.

D.5 Additional Results

In Table D.7, we present the Global Contrastive Quality scores for all 31 ST models across the three language pairs. These results form the basis for the regression analysis presented in Figure 7.5 in §7.7 of the main text.

In Figures D.1 and D.2, we compare the four representative models for Spanish and Japanese, mirroring the analysis performed for German in Figure 7.7 in the main text. Overall, the findings and observations coincide with those for German.

¹²huggingface.co/openai/whisper-medium

¹³huggingface.co/openai/whisper-large

¹⁴huggingface.co/openai/whisper-large-v3

¹⁵huggingface.co/facebook/nllb-200-distilled-600M

¹⁶huggingface.co/facebook/nllb-200-distilled-1.3B

¹⁷huggingface.co/facebook/nllb-200-3.3B

¹⁸huggingface.co/facebook/wav2vec2-large-960h-lv60-self

¹⁹huggingface.co/facebook/hubert-large-ls960-ft

²⁰huggingface.co/facebook/hubert-xlarge-ls960-ft

Model	Model Type	Size (B)	Contrastive Quality (Global)			
			eng-deu	eng-spa	eng-jpn	Average
SEAMLESSM4T-V1-MEDIUM	E2E	1.2	9.1	10.4	10.1	9.9
SEAMLESSM4T-V1-LARGE	E2E	2.3	7.8	8.7	8.2	8.2
SEAMLESSM4T-V2-LARGE	E2E	2.3	14.5	11.0	13.9	13.1
XLS-R-300M	E2E	0.3	10.2	9.3	9.5	9.6
XLS-R-1B	E2E	1.0	9.3	8.3	8.1	8.6
XLS-R-2B	E2E	2.0	7.3	8.4	7.0	7.6
ZEROSWOT-MEDIUM	E2E	0.9	8.9	7.5	7.5	8.0
ZEROSWOT-LARGE	E2E	0.9	8.7	7.7	7.9	8.1
SALMONN-7B	E2E	7.0	12.7	9.4	12.9	11.7
SALMONN-13B	E2E	13.0	15.9	12.3	16.1	14.8
SEAMLESSM4T-V1-MEDIUM	Cascade-AED	2.4	7.3	7.6	6.9	7.2
SEAMLESSM4T-V1-LARGE	Cascade-AED	4.6	6.5	5.8	4.3	5.5
SEAMLESSM4T-V2-LARGE	Cascade-AED	4.6	10.5	7.7	8.7	8.9
WHISPER-V1-MEDIUM → NLLB-600M	Cascade-AED	1.4	5.7	5.0	5.4	5.4
WHISPER-V1-MEDIUM → NLLB-1.3B	Cascade-AED	2.1	6.2	5.0	5.8	5.6
WHISPER-V1-MEDIUM → NLLB-3.3B	Cascade-AED	4.1	6.6	5.1	5.4	5.7
WHISPER-V1-LARGE → NLLB-600M	Cascade-AED	2.2	5.9	4.9	6.1	5.6
WHISPER-V1-LARGE → NLLB-1.3B	Cascade-AED	2.9	6.0	4.7	5.8	5.5
WHISPER-V1-LARGE → NLLB-3.3B	Cascade-AED	4.9	6.3	4.6	5.6	5.5
WHISPER-V3-LARGE → NLLB-600M	Cascade-AED	2.2	5.3	4.6	6.5	5.5
WHISPER-V3-LARGE → NLLB-1.3B	Cascade-AED	2.9	5.3	4.7	5.4	5.1
WHISPER-V3-LARGE → NLLB-3.3B	Cascade-AED	4.9	5.5	4.8	5.3	5.2
WAV2VEC 2.0 → NLLB-600M	Cascade-CTC	0.9	1.4	1.3	2.0	1.5
WAV2VEC 2.0 → NLLB-1.3B	Cascade-CTC	1.6	1.7	1.0	1.4	1.3
WAV2VEC 2.0 → NLLB-3.3B	Cascade-CTC	3.6	1.6	0.9	1.7	1.4
HUBERT → NLLB-600M	Cascade-CTC	0.9	3.2	2.7	2.5	2.8
HUBERT → NLLB-1.3B	Cascade-CTC	1.6	2.2	2.4	2.9	2.5
HUBERT → NLLB-3.3B	Cascade-CTC	3.6	2.7	2.6	2.9	2.7
HUBERT-XL → NLLB-600M	Cascade-CTC	1.6	2.4	2.9	3.0	2.8
HUBERT-XL → NLLB-1.3B	Cascade-CTC	2.3	3.5	1.7	2.7	2.6
HUBERT-XL → NLLB-3.3B	Cascade-CTC	4.3	2.6	2.4	2.5	2.5

Table D.7: Contrastive Quality (Global) scores for English-German, English-Spanish, and English-Japanese, including their averages.

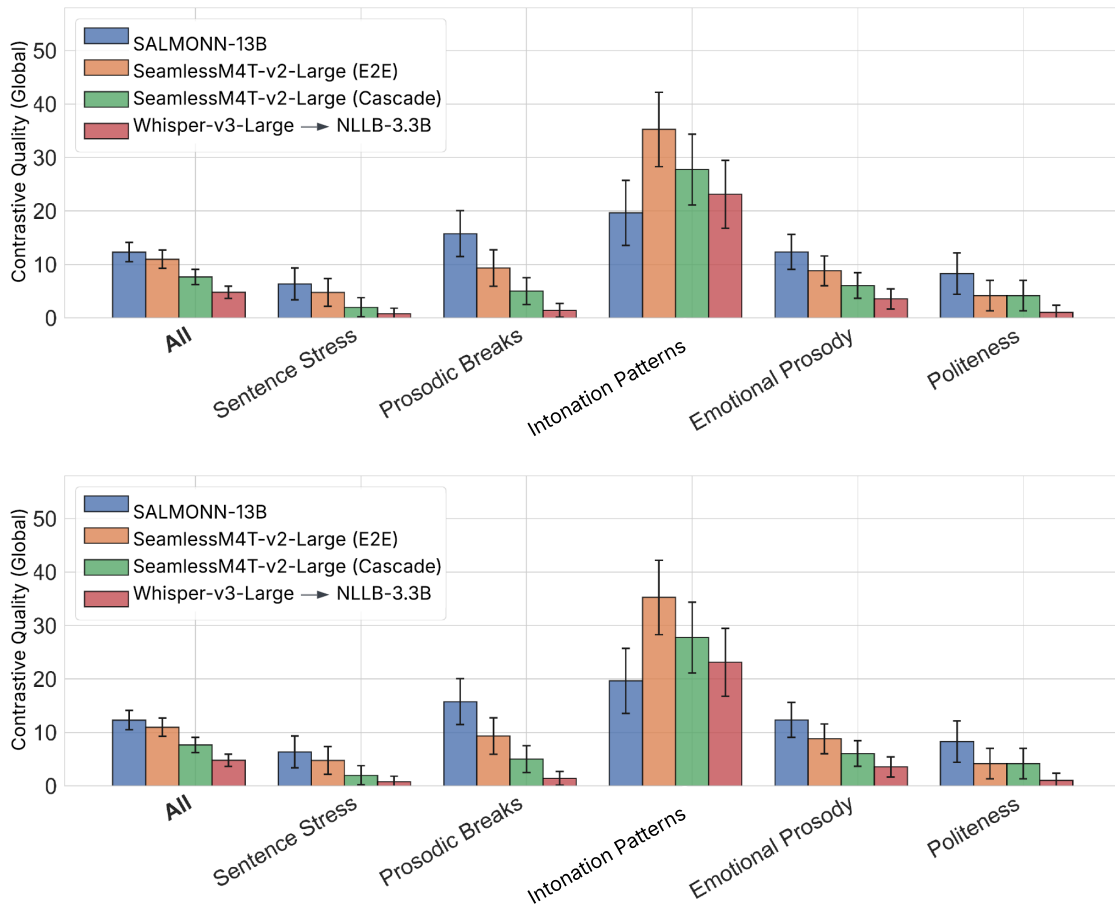


Figure D.1: Upper: Model performance per category (eng-spa). Lower: Model performance comparisons (eng-spa): (a) SALMONN-13B vs. SEAMLESSM4T-v2-LARGE, (b) SEAMLESSM4T-v2-LARGE (E2E) vs. SEAMLESSM4T-v2-LARGE (Cascade), (c) SEAMLESSM4T-v2-LARGE (Cascade) vs. WHISPER-v3-LARGE → NLLB-3.3B.

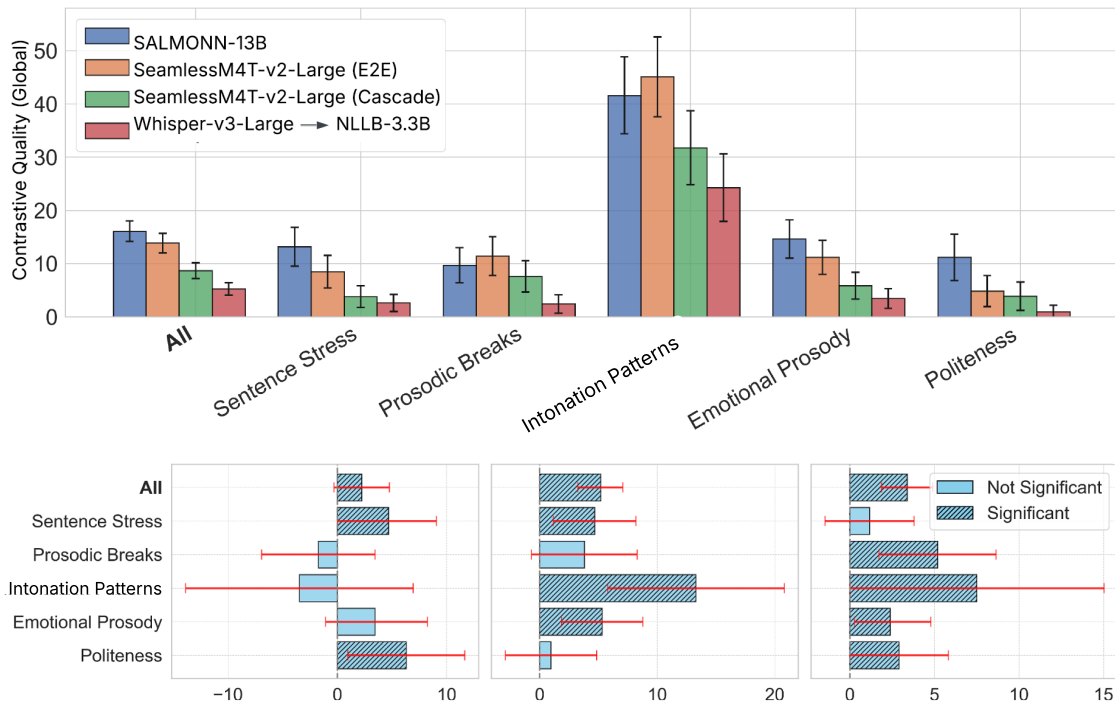


Figure D.2: Upper: Model performance per category (eng-jpn). Lower: Model performance comparisons (eng-jpn): (a) SALMONN-13B vs. SEAMLESSM4T-v2-LARGE, (b) SEAMLESSM4T-v2-LARGE (E2E) vs. SEAMLESSM4T-v2-LARGE (Cascade), (c) SEAMLESSM4T-v2-LARGE (Cascade) vs. WHISPER-v3-LARGE → NLLB-3.3B.

E Additional Main-Author Publications

In this chapter we provide details for the main-author papers written through the course of this doctoral thesis, that were not included in the main text.

E.1 Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation

This section is based on our work “Pretrained Speech Encoders and Efficient Fine-tuning Methods for Speech Translation”, published in **IWSLT 2022** (Tsiamas et al., 2022b).

Efficient Adaptation of Foundation Models. In this work, we addressed the computational and data challenges of training competitive End-to-End Speech Translation (E2E ST) systems by leveraging large-scale pre-trained models. Participating in the IWSLT 2022 Offline Speech Translation task (Anastasopoulos et al., 2022), we proposed a system architecture initialized with self-supervised speech encoders (comparing WAV2VEC 2.0 (Baevski et al., 2020) and HUBERT (Hsu et al., 2021)) coupled with a pre-trained multilingual MBART-50 decoder (Liu et al., 2020a). To mitigate the prohibitive computational cost of fully fine-tuning these massive architectures, we adopted and extended efficient fine-tuning strategies. We employed the LayerNorm and Attention (LNA) method (Li et al., 2021) and proposed complementing it with scaled parallel adapters (Houlsby et al., 2019) inserted into the frozen feed-forward layers. This methodological innovation allowed us to adapt the models to the specific translation task while updating only a fraction of the total parameters, effectively balancing model plasticity with training efficiency.

Encoder Comparison and Segmentation Robustness. Our experimental analysis revealed that the HUBERT encoder consistently outperformed WAV2VEC 2.0, providing a more robust foundation for the ST task. Furthermore, we addressed the input processing bottleneck inherent in real-world evaluation scenarios. By applying our previously proposed Supervised Hybrid Audio Segmentation (SHAS) method (see Ch. 3) to the

unsegmented IWSLT test sets, we achieved substantial improvements of up to 5 BLEU compared to the provided baseline segmentation. We also investigated the utility of Knowledge Distillation (KD) (Hinton et al., 2015) from a Machine Translation teacher; however, we observed that KD yields diminishing returns when the performance gap between the MT teacher and the ST student is narrow. Ultimately, our best submission, an ensemble of models utilizing HUBERT and parallel adapters, achieved competitive results on the English-German, English-Japanese, and English-Chinese tracks, demonstrating that parameter-efficient adaptation of foundation models, combined with optimal segmentation, is a viable path for high-performance speech translation.

E.2 Speech Translation with Foundation Models and Optimal Transport

This section is based on our work “Speech Translation with Foundation Models and Optimal Transport”, published in **IWSLT 2023** (Tsiamas et al., 2023b).

Siamese Pretraining and Cross-Modal Alignment. In this contribution to the IWSLT 2023 Offline Speech Translation task (Agarwal et al., 2023), we investigated methods to bridge the representation gap between independently pre-trained foundation models. We constructed an E2E system by coupling a WAV2VEC 2.0 encoder (Baevski et al., 2020) with an MBART-50 decoder (Liu et al., 2020a). The core contribution of this work, which serves as a precursor to the ZEROSWOT methodology presented in Ch. 5, is an extended “Siamese” pretraining strategy. We employed Optimal Transport (OT) (Frogner et al., 2015; Le et al., 2023) to align the acoustic representations with the semantic space of the frozen text encoder before supervised fine-tuning. Unlike ZEROSWOT, which relies solely on this alignment for zero-shot inference, this work utilizes OT as a robust pre-training objective to maximize transfer learning from the Machine Translation foundation model, effectively initializing the speech encoder to produce “text-like” embeddings.

Data Augmentation and Segmentation. To further enhance the system’s robustness to data scarcity and segmentation mismatches, we integrated several data-centric innovations proposed throughout this dissertation. We generated synthetic training data using SEGAUGMENT (see Ch. 4), creating diverse views of the training utterances to prevent

overfitting to specific segmentation patterns. Additionally, we incorporated Knowledge Distillation (Hinton et al., 2015) from the fine-tuned MT model to guide the ST training (Liu et al., 2019). At inference time, we processed the unsegmented test sets using SHAS (see Ch. 3), tuning the segmentation parameters specifically for the TED and ACL domains. The combination of OT-based alignment, robust data augmentation, and optimal segmentation allowed our single models and ensembles to achieve very competitive performance, validating the efficacy of aligning speech and text modalities via optimal transport in a supervised setting.

E.3 Efficient Speech Translation with Dynamic Latent Perceivers

This section is based on our work “Efficient Speech Translation with Dynamic Latent Perceivers”, published in **ICASSP 2023** (Tsiamas et al., 2023c).

Linear Complexity with Perceiver Encoders. While the Transformer architecture (Vaswani et al., 2017) has become the standard for E2E ST (Wang et al., 2020b), its self-attention mechanism suffers from quadratic complexity $\mathcal{O}(n^2)$ with respect to the sequence length n . This poses a significant efficiency bottleneck for speech signals, which are inherently much longer than text. In this work, we proposed the *Speech-to-Text Perceiver* (S2T-PERCEIVER), an architecture that mitigates this issue by replacing the standard encoder with a Perceiver (Jaegle et al., 2021). The model utilizes a cross-attention layer to project the variable-length input acoustic features into a fixed-length latent space of size m , followed by deep processing via self-attention solely within this latent space. This design effectively decouples the computational cost from the input length, reducing the complexity to $\mathcal{O}(m \cdot n)$, which is linear with respect to the speech duration.

Dynamic Latent Access for Flexible Inference. To further enhance the efficiency and flexibility of the model, we introduced *Dynamic Latent Access* (DLA), a novel training and inference strategy. DLA allows the model to be trained with a large latent capacity while only accessing a random subset of latents at each training step, thereby increasing expressive power without incurring additional computational overhead. Crucially, this training regime enables a flexible inference mechanism where the number of active

latents can be adjusted dynamically based on the available computational budget. We proposed a diversity-based selection algorithm to identify the most informative latents during inference, allowing the S2T-PERCEIVER to scale down to $0.6\times$ the FLOPS of a standard S2T-Transformer while maintaining competitive translation quality on the MuST-C benchmark (Di Gangi et al., 2019a).

E.4 Sequential Contrastive Audio-Visual Learning

This section is based on our work “Sequential Contrastive Audio-Visual Learning”, published in ICASSP 2025 (Tsiamas et al., 2025b) and developed during a research internship at Dolby Laboratories between November 2023 and February 2024.

Fine-grained Alignment Beyond Speech Translation. In this work, we explored the broader applicability of cross-modal alignment beyond the text–speech domain. We identified a critical bottleneck in standard Contrastive Audio-Visual Learning (CAV) (Morgado et al., 2021): the temporal aggregation of representations into global embeddings, which obscures fine-grained sequential information. To overcome this, we introduced *Sequential Contrastive Audio-Visual Learning* (SCAV), a framework that aligns modalities in their native, non-aggregated sequential space. We experimented with various sequential distance metrics, including Dynamic Time Warping (Vintsyuk, 1968) and Optimal Transport (Wasserstein distance) (Frogner et al., 2015), to compute the alignment cost between audio and visual sequences. This approach effectively extends the alignment principles discussed in Ch. 5 to the audio-visual domain by leveraging sequential geometry to capture synchronicity and semantic correspondence without temporal compression.

Retrieval Performance and Efficiency. Our experiments on the VGGSound (Chen et al., 2020) and Music (Zhao et al., 2018) datasets demonstrated that SCAV yields substantial improvements in retrieval tasks, outperforming standard aggregation-based methods by up to $3.5\times$ in recall. Notably, the method excelled in zero-shot retrieval scenarios, validating the robustness of the learned representations. To address the computational cost of sequential comparisons, we further introduced a hybrid retrieval mechanism that combines the efficiency of global embeddings with the precision of sequential distances. While this work targets audio-visual representation learning, it reinforces the fundamental hypothesis of

Part **III** of this thesis: that bridging the modality gap requires robust, geometry-aware alignment strategies capable of capturing the rich temporal dynamics inherent in multimodal data.