UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

# *Learning multilingual and multimodal representations with language-specific encoders and decoders for machine translation*

## Carlos Escolano Peinado

**ADVERTIMENT** La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del repositori institucional UPCommons (http://upcommons.upc.edu/tesis) i el repositori cooperatiu TDX (http://www.tdx.cat/) ha estat autoritzada pels titulars dels drets de propietat intel·lectual **únicament per a usos privats** emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei UPCommons o TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a UPCommons (*framing*). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA** La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del repositorio institucional UPCommons (http://upcommons.upc.edu/tesis) y el repositorio cooperativo TDR (http://www.tdx.cat/?locale-attribute=es) ha sido autorizada por los titulares de los derechos de propiedad intelectual **únicamente para usos privados enmarcados** en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio UPCommons No se autoriza la presentación de su contenido en una ventana o marco ajeno a UPCommons (*framing)*. Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING** On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the institutional repository UPCommons (http://upcommons.upc.edu/tesis) and the cooperative repository TDX (http://www.tdx.cat/?locale- attribute=en) has been authorized by the titular of the intellectual property rights **only for private uses** placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading nor availability from a site foreign to the UPCommons service. Introducing its content in a window or frame foreign to the UPCommons service is not authorized (framing). These rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author.

Universitat Politècnica de Catalunya

PhD Thesis

# Learning Multilingual and Multimodal Representations with Language-Specific Encoders and Decoders for Machine Translation

Carlos Escolano Peinado

Advisors:
Marta Ruiz Costa-Jussà
Josè Adrián Rodriguez Fonollosa

2021

*"Just because you can explain it*
*doesn't mean it's not still a miracle."*

TERRY PRATCHETT, *Small Gods*

# Abstract

Multilingual Machine Translation is the task that focuses on methods to translate between several pairs of languages in a single system. It has been widely studied in recent years due to its ability to easily scale to more languages, even between pairs never seen together during training (zero-shot translation). Several architectures have been proposed to tackle this problem with varying amounts of shared parameters between languages. Current state-of-the-art systems focus on a single sequence-to-sequence architecture where all languages share the complete set of parameters, including the token representation. While this has proven convenient for transfer learning, it makes it difficult to incorporate new languages to the trained model as all languages depend on the same parameters.

What all proposed architectures have in common is enforcing a shared presentation space between languages. Specifically, during this work, we will employ as representation the final output of the encoders that the decoders will use to perform cross-attention. Having a shared space reduces noise as similar sentences at semantic level produce similar vectorial representations, helping the decoders process representations from several languages. This semantic representation is particularly important for zero-shot translation. The representation similarity to the languages pairs seen during training is key to reducing ambiguity between languages and obtaining good translation performance.

Our contributions focus on studying several methods to obtain a common multilingual representation without parameter sharing. Firstly, we propose a training method that enforces a common representation for bilingual training and a procedure to extend it to new languages. Secondly, we propose another training method that allows this representation to be learned directly on multilingual data and can be equally extended to new languages. Thirdly, we show that the proposed multilingual architecture is not limited only to textual languages. We extend our method to new data modalities by adding speech encoders, performing Spoken Language Translation, including Zero-Shot, to all the supported languages.

Our main results show that the common intermediate representation is achievable in this scenario, matching the performance of previously shared systems while allowing the addition of new languages or data modalities efficiently without negative transfer learning to the previous languages or retraining the system.

# Acknowledgements

A Ph.D. is never an easy task, involving a great deal of commitment and some bumps around the road. Independently of the results and experiments contained in this thesis, there is an even larger set of ideas and possible improvements that didn't make the cut or lead to negative results. In addition to this process, I surely did not expect that coping with a pandemic would be part of this process, adding an extra layer of uncertainty, virtual meetings, and conferences to this journey.

Despite all this, this journey has been a great opportunity to grow and learn not only about the topic of the thesis but about myself and the world of research. During these years, I've had a lot of opportunities to work with incredible people and learn about their views on research and life in general. As no adventure like this can be done alone, I would like to take this opportunity to thank everyone that has contributed to turning this thesis into a reality.

To my supervisors, Marta Ruiz Costa-Jussà and José Adrián Rodriguez Fonollosa, whose constant help and guidance made this thesis possible. I arrived at the department more than six years ago as a bachelor student without ever considering research as a career path, and during these years, they helped me discover this world of research and NLP.

To Mikel Artexte, whose ability to analyze a problem at first glance, along with the discussion with my supervisors, sparked some of the ideas discussed throughout this thesis.

To my labmates at UPC, especially Bardia, Casimiro, Christine, Magdalena, and Noé, who shared this adventure with me from the beginning, were of great help at times when all experiments seemed to fail. Going to ACL together was undoubtedly one of the highlights of this journey. Also to Gerard and Jordi, whom I met as students and grew to become colleagues during these years. And to my office mates Ferran and Andreu, for making waiting for experiments to run a much easier task during the first part of this process.

To Xavier Giró, Nuria Castell, and Alberto Abello, who gave me the opportunity to teach at their courses and discover that teaching is a great way to learn in-depth about a topic.

To Telefónica Research and Amazon Barcelona, for the opportunity to do a research internship and see different ways of doing research from the industry's perspective.

To my friends, Sara and David, and especially Joaquín, who were there during this journey to help me disconnect from NLP problems when I needed it most, or as David knows, to listen to me rehearsing my first lecture on a train and not even complaining. And to little Leo, watching him grow, even remotely, has helped brighten some long days preparing this document.

To my family, my parents Carlos and Maria Clara, and my sister Sofía, who are always there to support me and always believed in me doing this Ph.D., even before I did myself. Especially during the pandemic and lockdown months.

Finally, I would like to thank the VEU and Machine Translation research groups at UPC, as well as the different research projects that have funded this work through the years. To the LUNAR project by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947657). To the Spanish Ministerio de Ciencia e Innovación, the Agencia Estatal de Investigación through the ADAVOICE project (PID2019-107579RB-I00 (agreement AEI/10.13039/501100011033)) and the Ministerio de Economıa y Competitividad y el Fondo Europeo de Desarrollo Regional throught the DEEPVOICE (TEC2015-69266-P) and AMALEU (EUR2019-103819) projects. And to the Google Faculty Research awards 2018 and 2019, awarded to my supervisor, Marta Ruiz Costa-Jussà.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Motivation

Machine Translation (MT) is the area of research focused on the automatic translation of languages. For 60 years, linguists and computer scientists have worked to produce more reliable systems that kept the meaning of the translated sentence and generated text following the same structure and idioms that a native speaker would employ. Over the years, it has become a pervasive technology in our everyday lives with multiple commercial solutions and millions of everyday users. Its importance can be even more evident during a crisis where people from different parts of the world interact with limited resources. An example of this phenomenon is the 2010 Haiti earthquake when a Haitian Creole-English system [Lewis, 2010], deployed in under a week, helped volunteers from all around the world communicate with the local population.

Since the beginning of the field the idea of developing common language representations has been an objective [Vauquois, 1968] where a common interlingua serves as bridge between all languages as any other language could be translated to or from it. Current state-of-the-art neural machine translation (NMT) systems forget this objective by following standalone architectures that focus on a fixed set of languages. Each model is evaluated for its own task, and in order to perform a new task, a new model is trained from scratch or fine-tuned, harming or completely forgetting the original one.

From the current NMT approaches, the ones that more resemble this interlingua objective are multilingual NMT systems where a single system can translate between several pairs of languages. These systems are trained on a combination of parallel data from all desired language pairs with the objective of learning a common representation that allows knowledge transfer and even zero-shot translation between non-observed directions during the supervised training. Several architectures have been proposed in the last years seeking better cross-lingual mappings and transfer learning between languages. From the initial language-specific encoders and decoders models, [Firat et al., 2016b, Firat et al., 2016a] where each language used its own set of parameters to fully shared universal encoder-decoders architectures [Ha et al., 2016, Johnson et al., 2017], where all languages share the same parameters, creating a stronger dependency between language.

## 1.2 Objectives

In this thesis, we argue that systems should not be considered standalone systems for a limited amount of tasks but as a flexible platform that can be incrementally extended to new tasks in the future, benefiting from the previous knowledge encoded in the model. From this general idea, we devise the four main objectives covered throughout this work:

1. **Explore the capabilities of language-specific encoder-decoder architectures**: Recover the language-specific encoder-decoder approach and analyze its benefits and limitations compared to current state-of-the-art architectures. To achieve this objective we study the obtained cross-lingual representations of all our methods (Sections 3.4.3, 4.6.6, and 5.4.2) as well as different data conditions (Section 4.6.4) and fine-tuning (Section 4.6.5)

2. **Learning cross-lingual representations without parameter sharing**: The models should learn a common representation for all languages, ensuring the performance of any new language projected into this space. To enforce the modularity of the approaches, encoders, and decoders should be language-specific, reducing the dependencies between languages. To achieve these objectives, we propose both a bilingual method (Section 3.1) based on auxiliary tasks and two multilingual methods (Sections 4.2 and 4.3).

3. **Training new languages on a previous cross-lingual representation**: New language encoders/decoders should be able to learn the cross-lingual mapping of the system (Objective 2) while being more efficient than training a new

model from scratch. To achieve this, we propose an incremental training approach (Sections 3.2 and 4.2) training only the new modules to the system.

4. **Training new modalities modules on a previous cross-lingual representation**: Encoders for new data modalities should be able to learn the cross-lingual mapping of the system (Objective 2) without modification of the previous models. To achieve this, we propose a method to incrementally train Speech encoders (Section 5.3) for the task of Spoken Language Translation.

## 1.3 Contributions

This thesis focuses on four main aspects of multilingual machine translation: Supervised translation, zero-shot translation, language addition, and cross-lingual mappings. We highlight this as the main empirical findings throughout this work:

- When data between sufficient languages, language-specific encoder-decoder architectures can compare to or even outperform universal encoder-architecture ones in terms of supervised performance and transfer learning (Section 4.6).

- Systems can be efficiently extended to new tasks and modalities just by training in combination with previously frozen modules, without modification of the previous modules (Sections 3.4, 4.6, and 5.4).

- Adapter modules efficiently bridge data modalities even on non-pretrained systems (Section 5.4).

During this thesis, we also propose four methods based on language-specific encoder-decoders architecture ranging from bilingual machine translation to multilingual machine translation and spoken language translation:

- **Multilinguality by Incremental Training**: A method to train a bilingual system that can be incrementally extended to new languages while allowing zero-shot translation to previous languages on the system (Sections 3.1, and 3.2).

- **Multilingual Joint Training**: Two methods to train multilingual systems without parameter sharing while allowing the languages to be incrementally added (Sections 4.2, and 4.3).

- **Multimodality by Incremental Training**: A method to incrementally train Speech encoders to a preexisting multilingual, multilingual NMT system, allowing to perform zero-shot spoken language translation (Sections 5.1, and 5.3).

## 1.4 Outcomes of the Thesis

Main publications discussed throughout this thesis:

- [Escolano et al., 2019a] **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2019a). From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Compu-tational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy.Association for Computational Linguistics.

- [Escolano et al., 2020b] **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (2020b). Training multilingual machine translation by alternately freezing language-specific encoders-decoder. *Arxiv Preprint.*

- [Escolano et al., 2020a] **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2020a). The TALP-UPC system description for WMT20 news translation task:Multilingual adaptation for low resource MT. In *Proceedings of the Fifth Confer-ence on Machine Translation*, pages 134–138, Online. Association for Computa-tional Linguistics.

- [Escolano et al., 2020c] **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Segura, C. (2020c). Enabling zero-shot multilingual spoken language translation with language-specific encoders and decoders. *Accepted to IEEE-ASRU 2021*

- [Escolano et al., 2021b] **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J.

A. R.(2021). From bilingual to multilingual neural machine translation by incremental training. In *Journal of the Association for Information Science and Technology* p. 190-203

- [Escolano et al., 2021a] **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (2021). 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

- [Escolano et al., 2022] **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2022). Multilingual machine translation: Deep analysis of language-specific encoder-decoders. *Accepted to the Journal of Artificial Intelligence Research*

Industry Internships during the Ph.D:

- **Telefónica Research**: 07/2020-08/2020 & 12/2020-02/2021. Working on Spoken Language Translation. Publication: [Escolano et al., 2020c]

- **Amazon**: 09/2020-11/2020. Working on Keyword Extraction.

The following patent includes work described in this thesis:

- [pat, ] **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R. Multilngual Translator. 2021. Patent applied for a multilingual and multimodal translation system.

Publications during the PhD.not directly related to the work presented in this thesis:

- [Vila et al., 2018] Vila, L. C., **Escolano**, C., Fonollosa, J. A., and Costa-Jussa, M. R.(2018). End-to-end speech translation with the transformer. In *IberSPEECH*,pages 60–63.

- [Casas et al., 2019] Casas, N., Fonollosa, J. A. R., **Escolano, C.**, Basta, C., andCosta-jussà, M. R. (2019). The TALP-UPC machine translation systems forWMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: SharedTask Papers, Day 1)*, pages 155–162, Florence, Italy. Association for Computational Linguistics.

- [Escolano et al., 2019b] **Escolano, C.**, Costa-jussà, M. R., Lacroux, E., and Vázquez,P.-P. (2019b). Multilingual, multi-scale and multi-layer visualization of intermedi-ate representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages151–156, Hong Kong, China. Association for Computational Linguistics.

- [Armengol-Estapé et al., 2020] [Armengol-Estapé et al., 2020] Armengol-Estapé, J., Costa-jussà, M. R., and **Escolano, C.** (2020). Enriching the transformer with linguistic factors for low-resource machine translation. *Under Review*

- [Costa-jussà et al., 2020] Costa-jussà, M. R., **Escolano, C.**, Basta, C., Ferrando,J., Batlle, R., and Kharitonova, K. (2020). Gender bias in multilingual neuralmachine translation: The architecture matters. *Arxiv Preprint.*

- [Gállego et al., 2021] Gállego, G. I., Tsiamas, I., **Escolano, C.**, Fonollosa, J. A. R.,and Costa-jussà, M. R. (2021). End-to-end speech translation with pre-trainedmodels and adapters: UPC at IWSLT 2021. In *Proceedings of the 18th Interna-tional Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119,Bangkok, Thailand (online). Association for Computational Linguistics.

## 1.5 Thesis outline

In this section, we present an outline of the chapters of this thesis, each with its related papers. Text and figures from such papers are reused and adapter through this thesis.

In chapter 2 we introduce the most relevant background concepts required throughout this work and their current state-of-art. The chapter is organized into two main blocks focusing on deep learning and natural language processing topics covered during this thesis.

In chapter 3 we propose a bilingual NMT system that can be extended to new languages without retraining while allowing zero-shot translation. [Escolano et al., 2019a, Escolano et al., 2021b]

In chapter 4, we propose two methods to train multilingual NMT while maintaining the ability to train new languages incrementally. We also perform an extensive comparison of our model with the state-of-the-art universal encoder-decoder architecture. [Escolano et al., 2020a, Escolano et al., 2020b, Escolano et al., 2021a, Escolano et al., 2022]

In chapter 5, we propose a method to extend the method from the previous chapter to the task on spoken language translation by incrementally training a Speech decoder. We also analyze the impact of Adapter modules on the task and in the learned encoding representation. [Escolano et al., 2020c]

In chapter 6, we present the final conclusions of this thesis, reflecting on the different contributions and research objectives.

# 2 Literature Review

In this chapter, we are going to discuss some of the theoretical concepts upon the work described in this thesis is built. Section 2.1 will discuss the main Deep Learning topics applied, focusing on the architectures employed and their importance for this work. Section 2.2 will provide an introduction to Natural Language Processing and the different tasks we have cover in our experiments.

## 2.1 Deep Learning

We name Deep Learning, the area of machine learning focused on models based on neural networks. The name "Deep" comes from the fact that these models usually stack several layers to perform their tasks. Each layer transforms the data until arriving at the desired representation space instead of performing a single transformation over the input data. For example, a given input space can become linearly separable by applying transformations, allowing the model to classify the data.

This section will focus on generative models that given input data, map the data into a latent representation space, and produce synthetic data conditioned on this representation. This model vary from generating a reconstruction of the input data (Autoencoders, section 2.1.1) to generating completely new sequences (section 2.1.3 ) such in Machine Translation. We'll also focus on the latent space and how to adapt it for different tasks or domains (sections 2.1.2 and 2.1.4 ).

### 2.1.1 Autoencoders

From those models, the first one we are going to discuss is Autoencoders [Lecun, 1987, Bourlard and Kamp, 1988]. An Autoencoder is a system that, given some input data $x$, computes its reconstruction $y$ without requiring additional labeling. The objective of these techniques is not to copy the data directly but to compute latent representations, called codes, that encode the most important features of the input data. Learned codes could be used as dimensionality reduction of the original data or feature selection for other models. We can identify two main components:

- **Encoder:** Given input data $x$ compute a latent representation $h$ that represents the most important features of the data defined as: $h = f(x)$

- **Decoder:** Given a latent representation $h$ generates a new data point $y$ on the same space of the original input data. We can define this process as: $y = g(h)$

During training, the objective is to reduce the distance between the generated $y$ and the original $x$. As stated, the model's objective is learning codes that represent the input data, not a perfect reconstruction. In order to improve this representation, several variants of this process have been proposed through the years. Undercomplete autoencoders reduce the dimensionality of the latent representation in order to create an information bottleneck, inducing the model to focus on the most salient features. Sparse autoencoders [Hinton, 1984] also achieve this bottleneck, limiting the neuron's activation by adding a regularization term to the reconstruction objective. In order to prevent the model from just copying the data, Denoising Autoencoders [Vincent et al., 2008] add noise in the form of random noise or permutation to the input data and try to reconstruct the original data without noise.

All the mentioned methods follow the same principle of learning representations of the input data. Another variant is Variational Autoencoders(VAE) [Kingma and Welling, 2014] that represent inputs as a probability distribution instead of points in a learned space. Once the model is trained, this allows the generation of synthetic data points in the distribution by sampling the learned probability distribution.

### 2.1.2 Common Representation Learning

Representation learning and Autoencoders are useful in diverse scenarios, but it is not uncommon to have access to more than one view of our data. Video input that contains both images and audio or parallel text in two languages are examples of this. Using Autoencoders, we could learn features from each data source, but those would not benefit from any knowledge transfer between the different views.

Common Representation learning seeks to learn a shared representation from different views, Obtaining a representation that focuses on the mutual information between views, discarding view-specific details. There are several methods for this task, but two main families can be defined:

- **Multiview Representation Alignment:** Given two or more views, compute a mapping between the view to a shared space. Examples of this technique are Canonical Correlation Analysis (CCA) [Hotelling, 1936] And Partial Least Squares [Wold, 1982].

- **Deep Representation Fusion:** Given two or more views, learn a shared representation from scratch. All views are required to compute the representation to capture the most salient features from all of them. Examples of this technique are MultiModal Deep Autoencoders [Ngiam et al., 2011].

In this section, we will focus on Representation Fusion as it is the closest to the topic we want to discuss. As mentioned, the main objective of these methods is mapping different views of the same data in a common representation space. This space allows the model to represent the data even from a partial set of the original views. Different approaches have been proposed to achieve this goal. Distance-based methods [Li et al., 2003] compute a distance metric between the representations, usually in combination with additional performance metrics to enforce the model's reconstruction performance. Another popular approach is correlation-based methods [Andrew et al., 2013] that try to maximize the correlation between the different views. This measure could be a less restrictive constraint over the space, as distances may not perform well on high dimensional spaces [Aggarwal et al., 2001]. For pre-computed representations, applying an orthogonal transformation [Artetxe et al., 2017] has also been studied, as this linear transformation preserves the length vector and angles from the original space, and therefore the relation between elements in the same space.

A common characteristic of these methods is the trade-off between the two objectives of the system, reconstruction of the input data, and the mapping between the representation space. Some works [Chandar et al., 2016, Jaques et al., 2017, Silberer and Lapata, 2014] have tried to solve this problem by combining the previous measures, reconstruction, and distance/correlation, with an additional task of cross-reconstruction. For this task, given input data from one of the available views to generate a different view. On implementations with view-specific encoders and decoders, e.g., CorrNets [Chandar et al., 2016] this additional task contributes to the mapping as it trains the model to decode the views from different input views or modalities.

### 2.1.3 Sequence-to-sequence models

In previous sections, we have discussed methods to learn representations based on the reconstruction of the input data. However, representation learning is required for other tasks that focus on the generation of new target data. This section will focus on Sequence-to-Sequence generation as it is the framework used in our experiments. We can consider a sequence a series of elements $x = \{x_1, x_2, ..., x_n\}$ , each with its individual set of features, positioned in a determined order. Order is important, as we can extract information from the features of each element in the sequence and the context learned from the surrounding elements in the sequence. An example of sequences in this sense would be text, given the sentence *The boy fell, but he was ok.* each of the words would be an element of the sequence, and we used their position in it to understand the meaning of the whole sentence. Not only is the order important to create grammatical sentences, but also, we can understand that *he* references "the boy" thanks to the context provided by the previous elements in the sentence.

Since the first proposed model [Sutskever et al., 2014] the most common approach has followed an Encoder-Decoder architecture. This model showed the first competitive results on Machine Translation by an end-to-end neural system. It is based on Recurrent Neural Networks (RNN) [Rumelhart et al., 1986] for both encoder and decoder. Following the basic RNN framework, given a sequence $x$ each step of the sequence:

$$c_t = f(Wc_{t-1} + Ux_t + b) \tag{2.1}$$

$$h_t = Vc_t + a \tag{2.2}$$

where $W$, $U$ and $V$ are weight matrices, $b$ and $a$ are bias vectors, $f()$ is an activation function, $c_{t_1}$ is the context vector from the previous element in the sequence and $x_t$ is the current element in the sequence. Equation 2.1 shows that for each element the context vector is updated by a transformation learned from the previous context and the current element. The output, Equation 2.2, is computed from the context vector, including the current element. The most common implementations on this principle for Sequence-to-Sequence are Long Short Term Memories (LSTM) [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Units [Cho et al., 2014].

As in the previous section, the Sequence-to-Sequence model consists of an Encoder-Decoder architecture as follows:

- **Encoder:** A RNN layer processes each element of the sequence one by one, computing an output hidden representation and a context vector that serves as additional input for the next element of the sequence. This context vector keeps information of all the previous elements. The context vector from the last element is used as the decoder's initial context.

- **Decoder:** A RNN that generates one element of the sequence at a time in an auto-regressive manner given the context vector from the encoder and the last decoded element from the output sequence. As new elements are generated, the context vector is updated to include the current state of the output sequence. In order to decode the following output from the output sequence and not the current one, all elements are shifted right by one position by including a *Beginning of sentence (BOS)* element to the decoder as the initial output element and *End of sentence (EOS)* token as the last element to predict by all sequences. The output is mapped into the number of tokens and normalized with a softmax function when predicting discrete outputs.

Figure 2.1 shows this architecture omitting all the specific elements of the problem or modality. An important aspect of this architecture for this work is the decoupling of the source encoding and the decoding. The target does not modify source representation, which is beneficial in terms of efficiency. The cost of processing the source sequence grows linearly with the number of elements and allows the combination of different encoders and decoders for a multiview task without any modification.

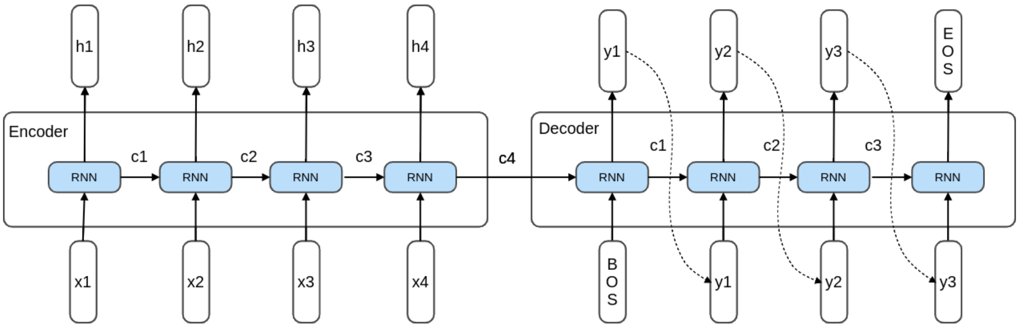

**Figure 2.1:** Sequence to Sequence architecture

All the process is fully differentiable, allowing the system to be trained in an end-to-end fashion. An optimization of this process is the use of Teacher Forcing [Williams and Zipser, 1989]. During training, the golden truth element from the target sequence is used instead of feeding the last decoded element to the decoder. This modification allows the network to compute the elements and removes the dependency from the previous decoded elements. At inference time, the decoded elements are used. The possible discrepancies between the real and learned distributions may lead to generation problems.

Even though the model was an important breakthrough on sequence generation tasks, it also shows some limitations that other works have addressed. Representing a full sequence into a single vector may create an information bottleneck, as the full sequence is encoded in a fixed size vector independently of the number of elements. Additionally, this context vector is modified as new tokens are generated. Especially on long sentences, this may lead to a loss of context information and repetition of the generated tokens. [Bahdanau et al., 2015] proposed two improvements over this architecture. First, the use of bidirectional RNN on the encoder to consider the context from the whole sequence. These layers consist of two RNN, one that processes the sequence from left-to-right and another from right-to-left. Finally, the output of both layers is concatenated.

The second proposed modification is the addition of Attention. Instead of computing a single representation for the entire source sequence, this mechanism aims to compute specific representations for each of the elements of the target sequence. This allows the system to generate the next token conditioned on a combination of the most important features of the source sentence. Given a set of encoder hidden representations $h = \{h_1, h_2, ..., h_n\}$ and the decoder representation of the last generated token $s_t$ a set of scores is computed for each hidden representation based on their importance or similarity concerning $s_t$. Several approaches have been proposed to compute these scores, feed-forward networks [Bahdanau et al., 2015] and the dot product between $h$ and $s$, and a softmax function is applied to normalize the scores $\alpha$. The attention representation is computed as the weighted sum of the hidden states $h$ by $\alpha$

#### 2.1.3.1 Transformer

Other alternatives have been proposed to the recurrent approach, from convolutional [Gehring et al., 2017] to attention-based Transformer architectures [Vaswani et al., 2017]. This section will focus on the latter, as it is the architecture employed in this work. The Transformer architecture follows the encoder-decoder architecture based on applying attention to the sequence features. Two different applications of attention can be found in this model:

- **Self-attention:** Attention computed over the same sequence to transform the data emphasizing its most salient features.

- **Cross-attention:** Attention computed over the source and target sequence to emphasize the most salient features from the source sequence given the current decoded target elements.

Attention is computed by splitting the elements features in $n$ fixed heads and performing attention independently for each of them, in a process called Multi-head attention. Each head computes attention as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{2.3}$$

Where $Q$ is the query matrix, $K$ is the key matrix, and $V$ is the value matrix. Relating this method to the attention mechanism from the RNN framework, the query matrix represents the current sequence, keys represent the set of hidden states from the input sequence used to compute the attention scores, and the values are also the hidden states from the input sequence over which the scores are applied after normalizing them, scaling by the key's dimensionality $d_k$ and applying a softmax function. Attention values are not summed as this attention is applied in parallel over all sequence elements without any recurrence. This lack of recurrence also removes the notion of order between elements, as attention is computed between all possible combinations between keys and queries. All elements are attended in the same way without considering their distance in the sequence order. To provide this information, positional encodings are added to both the encoder source and target decoder representations. Figure 2.2 shows how multi-head attention is performed. Before the attention at each head, queries, keys, and values are independently transformed. After attention, the outputs of all heads are concatenated and combined by a linear transformation.

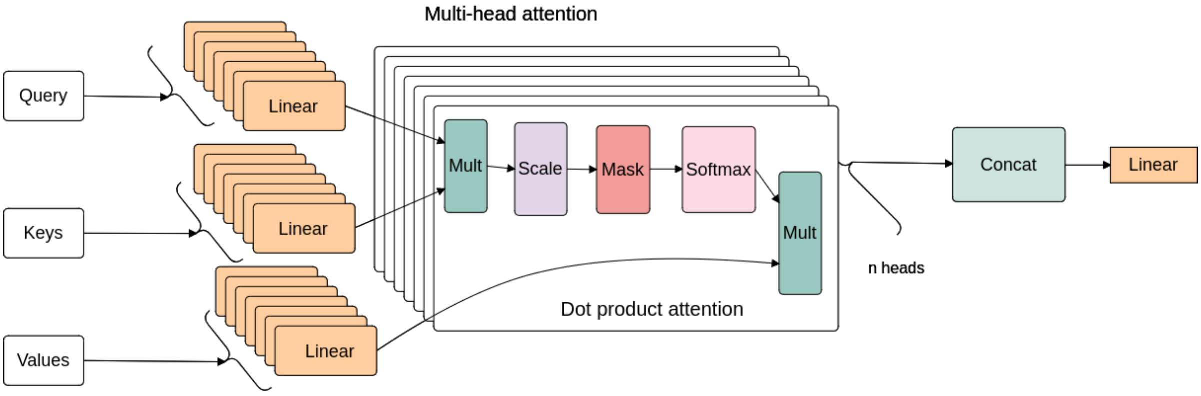

**Figure 2.2:** Multi-head attention

Figure 2.1.3.1 shows the complete Transformer architecture as was initially proposed for machine translation. The encoder consists of a stack of multi-head self-attention blocks followed by layer normalization and a feed-forward layer. Previous works [Geva et al., 2020] showed that feed-forward layers helped to combine the output of the different heads and acted as memories emphasizing patterns found in the data.

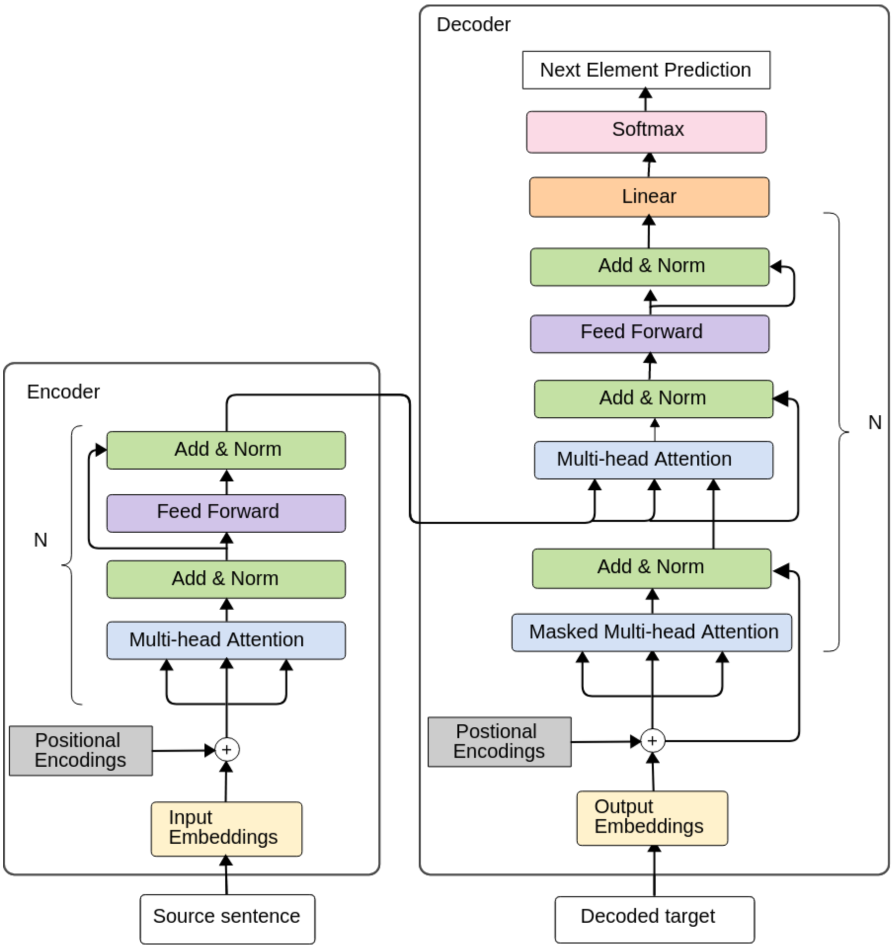

**Figure 2.3:** Transformer architecture

The Decoder, also based on a stack of multi-head attention layers. Instead of just self-attention, the first step of self-attention is applied to the previously decoded tokens. To prevent attending to elements not decoded yet, the upper triangle of the matrix is masked before the softmax normalization. The output of this attention block is used as queries for a step of cross-attention. In this case, the same computation is applied, but queries come from the previous step, and keys and values are the encoder's output. As in the recurrent architecture, this step aims at finding the most salient elements of the encoder hidden representation, according to the currently decoded elements of the target sequence.

Concerning model training, as in the RNN architecture, the model is trained end-to-end by optimizing a generation objective. As the model, by definition, can process sequences in parallel, teacher forcing is applied to allow the training of the complete sentence as a single block, speeding up training. Additionally, without recurrence, the model does not have dependencies between elements, allowing model parallelism between several GPUs, which allows the use of bigger batch sizes, contributing to model convergence. At inference time, elements are generated in an autoregressive manner, generating the next element and feeding the current decoded sentence as the Decoder's input.

### 2.1.4 Transfer Learning

As discussed before, models can benefit from jointly training different views from the same data, but there are other techniques and scenarios in which we can obtain knowledge training between our models, improving their generalization performance. According to the training method, we can distinguish two main approaches:

**Transductive Transfer Learning** consists of sharing knowledge between a previously trained model to train a new model for another task or domain. Several approaches have been proposed on this principle. Knowledge Distillation [Hinton et al., 2015] has been used to train a student model to learn the output distribution of a teacher model trained on another domain with more data available [Gaido et al., 2020]. Fine-tuning is another popular alternative to leverage the knowledge from a previous model into a new task or domain. This approach has gained momentum as large pretrained language models become more popular and show their ability to perform several tasks. For NLP-related tasks, models like ELMO [Peters et al., 2018a] based on RNN or BERT [Devlin et al., 2019] based on Transformer self-attention are commonly fine-tuned to new languages or domains. While fine-tuning improves the objective task, it is also subject to catastrophic forgetting of previous tasks. To maintain the model's performance while new tasks, Adapter modules [Houlsby et al., 2019] are feed-forward modules added to the original network that help the model to learn the new task by training a small set of parameters compared to the entire model. These modules can always be bypassed to perform the original tasks.

**Inductive Transfer Learning** to train models that better represent our data, we can also train a model that learns to perform more than one task. Unlike the transductive case, tasks are jointly trained, and all may benefit from the process. During training, the model needs to learn an internal representation that provides valuable information to all the tasks supported by the model, improving the performance for all tasks. This approach has been previously applied in NLP to tasks such as Part of Speech tagging (POS) and Name Entity Recognition [Niehues and Cho, 2017, Zaremoodi and Haffari, 2018], showing positive results. We can consider multilingual machine translation as an application of these techniques as the different languages benefit from the knowledge leaked from the other languages in the system.

## 2.2 Natural Language Processing

The discussed techniques are applicable to a variety of data sources and tasks. In this section we will focus on it applications to Natural Language. We can define Natural Language as the languages that are used by a community of speakers and which have evolved over generations naturally by the situation of such community. They are different from structured languages, such as programming language, that follow a set of predefined rules and which are designed for specific purpose. The evolution process has lead to languages that are easily understandable by humans but challenging to process by computers. Ellipsis, co-reference, irony are examples of resources that commonly used by human speakers but are not easily automatically processed.

Natural language Processing (NLP) includes a great variety of tasks. In this work, we are focusing on: Machine Translation, where two (section 2.2.1) or several (section 2.2.2) languages are involved; Spoken Language Translation (SLT) with the source language consisting of the audio speech utterances and the target language being text; and Natural Language Inference (NLI) (section 2.2.4) that looks for the sentence's semantics.

### 2.2.1 Machine Translation

The main task we will discuss during this work is Machine Translation. Given a sentence $x$ in the source language, this task aims to generate a sentence $y$ in the target language with the same semantic information. Natural languages have naturally evolved by the need of their communities of speakers, meaning that different languages have developed their own grammatical rules. Figure 2.4 shows some of the challenges that may be found when translating between distance languages such as Japanese and English. One of this differences is the script, while English uses the Latin script, Japanese uses a combination of kanji and katakana scripts, which do not show any explicit segmentation between words (e.g English uses the space). In addition, as observed in the example of Figure 2.4, there can be a difference in the matching between the words in both languages. Words with the same meaning can appear at different positions and languages can use different amount of words (or tokens) to produce the same meaning. Some words can be translated into several words, as *Eki* being translated into *train station* or some target words that do not appear on the source sentence may be required on the target language to produce a grammatically correct sentence as it is the case of *the* in the example.

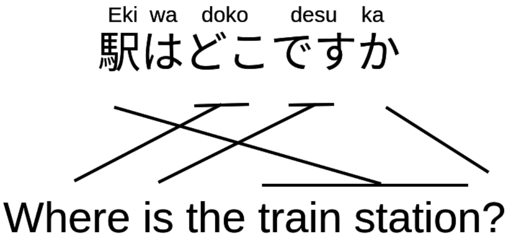

**Figure 2.4:** Word alignment example between Japanese and English

These phenomenons require models to find a match between tokens and learn to align and produce proper grammatical sentences. The first approaches were **Dictionary-based** systems that mapped words between languages, which showed poor results. **Rule-based Machine Translation (RBMT)** improved previous systems by adding syntactic and morphological rules designed by human experts. While requiring time and human effort, these systems leverage human knowledge to produce grammatical translations. **Statistical Machine Translation (SMT)**

[Weaver et al., 1955, Koehn et al., 2003] removes the need of human experts by learning the following probability $p(y|x)$ and applying the Bayes' rule:

$$p(y|x) = p(x|y)p(y) \tag{2.4}$$

Where $p(y)$ is a language model that learns the probability of words appearing together in the target language. Even though this approach learns the correct mappings from the data, it misses the alignment information. This word alignment can be computed with specific modules such as GIZA++ [Och and Ney, 2003]. Given an alignment $a$, it can be added to the translation model as:

$$p(y|x) = p(x|y, a)p(y) \tag{2.5}$$

SMT systems learn from human translations and they are complex systems that require tuning different components. Differently from SMT, **Neural Machine Translation** [Bahdanau et al., 2015] uses Sequence-to-Sequence [Sutskever et al., 2014] and attention-based mechanisms [Bahdanau et al., 2015] that allows to jointly learning translation and alignment. These models compute the conditional probability of the target sentence given the previously decoded tokens and source sentence as follows:

$$p(y_n|y_1, y_2, ..., y_{n-1}, x) \tag{2.6}$$

These models are trained in an end-to-end fashion using as objective the cross-entropy between the predicted target tokens for each pair of source and target sentences $(x, y)$ on the training dataset $D$ as follows:

$$L(x, y) = -\sum_{(x,y)\in D} logp(y|x) \tag{2.7}$$

All the approaches mentioned rely on parallel data between source and target languages to learn this conditional probability, but in some scenarios, monolingual data may be more abundant, as no labeling or manual translations are required to produce it. Several methods have been proposed to train fully unsupervised systems both using statistical machine translation [Artetxe et al., 2018] and neural machine translation [Artetxe et al., 2019, Lample et al., 2018]. On supervised models, monolingual data has also been used to improve train models [Bojar and Tamchyna, 2011] and improve the performance for low-resource languages [Currey et al., 2017] and combining both data sources [He et al., 2016]. The most popular approach is **back-translation** [Sennrich et al., 2016a] where the monolingual data is translated using a model in the opposite direction (target to source) to create a synthetic corpus and continue training a supervised model. This process can be done iteratively [Hoang et al., 2018] producing new synthetic data as the model is trained, improving its quality until convergence.

Text is essentially a sequence of discrete tokens. On NMT systems, it is represented as embeddings [Mikolov et al., 2013], non-contextual representations of each token on a contiguous high dimensional space, representing the tokens as well as their similarity. This can be learnt in an end-to-end fashion or pretrained from monolingual data [Mikolov et al., 2013, Pennington et al., 2014] . To generate text, we need to decide the vocabulary of possible tokens of our model. This vocabulary usually consists of a dictionary of the $V$ most frequent tokens on the available training data. All mentioned methods are subject to the vocabulary size available. Larger vocabularies can help improving the text generalization at the cost of requiring more computational resources and models with higher capacity. On the other hand, limiting the vocabulary size may lead to out-of-vocabulary (OOV) errors when the appropriate token has been excluded. OOV can lead to loss of contextual information on the source sentence and generation errors on the target. To tackle this problem, several alternatives have been proposed to represent text at different granularity levels, characters [Costa-jussà and Fonollosa, 2016, Chung et al., 2016], and bytes [Costa-jussà et al., 2017] reduce vocabulary size by an order $f$ magnitude at the cost of reducing the amount of semantic information provided by the individual tokens. Subwords , such as BPE [Sennrich et al., 2016b], and Sentencepiece [Kudo and Richardson, 2018] compute subwords representations based on the frequency they appear on the training data. These methods are the most popular as they represent a trade-off between words and characters approaches, allowing out-of-vocabulary words to split as subwords or even characters.

Another important aspect that has a significant impact on translation performance is decoding. The most straightforward method, **Greedy Decoding**, consists in selecting at each step the most probable token from the vocabulary given the normalized output from the network. The main disadvantage of this approach is missing error-control capabilities. Once a token is selected, it is used as input for the decoder, propagating the error to the following sentence tokens. **Beam Search** [Graves, 2012] alleviates this problem by keeping the $n$ options better scored according to their model's probability normalized by number of tokens. The $n$ most probable tokens are selected at any time, keeping only the $n$ best-scored hypotheses. This allows the decoding process to discard hypotheses that initially had high probabilities but divert as new tokens are generated. Selecting an appropriate beam size has a significant impact on the model performance [Britz et al., 2017]: small sizes do not considering enough possibilities while large sizes assign higher scores to low probability tokens.

To develop any system, it is necessary to evaluate the quality of the obtained results. The most accurate evaluation of MT would be **human evaluation** where an expert in both languages evaluates both the syntax and semantics of the translated sentence. This approach can be slow and expensive as several evaluators usually score the same sentence to measure their level of agreement. It has proven inefficient when constant evaluation is required or fluent speakers of both languages are scarce. **Automatic evaluation** is an alternative, where a metric is computed between the generated sentences and a set of human-translated references. The most common metric is BLEU (BiLingual Evaluation Understudy)[Papineni et al., 2002] which computes the precision of generated groups of contiguous words, n-grams, up to size $N$. This precision is computed as the sum of the clipped count of all n-grams in the candidate sentences $c$ divided by the total number of n-grams.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} count_{clip}(n - gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} count(n - gram')} \tag{2.8}$$

While equation 2.8 measures generated tokes that appear on the reference sentences, it does not measure its coverage. Partial translations that correctly predict n-grams but miss the complete meaning of the source sentence can be highly scored. To reduce this behavior, the following brevity penalty is added to the computation, penalizing candidates shorter than a reference sentence $r$:

$$BP = \begin{cases} 1 & if \quad c < r \\ \\ e^{1-\frac{r}{c}} & if \quad c \leq r \end{cases} \tag{2.9}$$

The final score for all n-gram precisions is obtained as shown in 2.10 usually with $N = 4$ and uniform weights $w_n = \frac{1}{N}$

$$BLEU = BP \cdot exp(\sum_{n-1}^{N} w_n log p_n) \tag{2.10}$$

There are some alternatives to BLEU, which mainly focus on achieving a higher correlation of scores to human evaluation. ROUGE [Lin, 2004] is similar to BLEU but it is based on recall, commonly used on text summarization tasks. Methods such as METEOR [Lavie and Agarwal, 2007], and BEER [Stanojević and Sima'an, 2014] try to better capture different phrasings or synonyms by introducing language-specific resources, which may limit the number of languages they can be applied to.

### 2.2.2 Multilingual Machine Translation

Training bilingual systems can be problematic when our objective is translating between several pairs of languages. Following the previous method, to translate all possible directions between $n$ languages, a total of $n(n-1)$ systems need to be trained. All the possible pairs only excluding the autoencoding directions. This quadratic growth of the number of systems can quickly become a time and computational resources limitation. **Multilingual Machine Translation** seeks developing systems that can translate between several languages by training a single system that can scale in a more efficient way with the number of languages. Several approaches have been proposed for **multi-source translation** [Och and Ney, 2001, Zoph and Knight, 2016], using several source languages as input to benefit from the information from the different views. As a limitation, these methods require multi-parallel data in both training and inference, limiting the applicability of such methods.

In this section, we will focus on **Multi-way Neural Machine Translation** methods that support several languages, but each sentence is translated from and to a single language. In this case, parallel data is only required between pairs of languages. Three types of systems can be described according to the available languages:

- **One-to-many translation:** A system is trained to translate between a single source language into several target languages.
- **Many-to-one translation:** A system is trained to translate between several source languages into a single target language.
- **Many-to-many translation:** A system that is trained to translate from several languages into several languages.

One-to-many and many-to-one translation benefit from transfer learning, by inductive bias, as the encoder and decoder respectively are common for all translation directions. Many-to-many translation is not as direct, and several architectures have been proposed to enforce this behavior.

The first proposed systems were based on **minimal parameter sharing** between languages [Luong et al., 2016, Dong et al., 2015]. These systems used language-specific encoders and decoders, reducing the system's growth from quadratic to linear with the number of languages. [Firat et al., 2016a] proposed the use of a common attention mechanism between languages, enforcing a common representation space. Improving this common representation also shows positive transfer learning from high to low resource languages, showing performance improvements compared to systems trained on bilingual data.

As previous works showed that multilingual systems benefited from sharing attention mechanism, the following proposed paradigm and the most popular to the present day is **fully shared parameters** between languages. These systems proposed a universal encoder and decoder for all languages. [Ha et al., 2016] proposed a system with language-specific embeddings and projection layer, keeping the token predictions to a single language. On the other hand, [Johnson et al., 2017] proposed a fully shared encoder-decoder including sharing embeddings between source and target languages, respectively. Language tokens are added to the source sentence to specify the desired target language. This architecture showed that, by sharing all parameters, it was able to map several languages into a common representation space [Kudugunta et al., 2019]. As in the minimally shared architectures, positive transfer learning is observed from high resource to low resource languages, but also negative transfer for high resource language, underperforming when compared to bilingual models trained on the same data [Arivazhagan et al., 2019b]. Sampling strategies [Wang and Neubig, 2019] during training have shown that can further transfer learning benefits at the cost of higher convergence times.

Fully shared architectures were one of the earliest to exhibit **zero-shot translation**, which allows translation in inference between pairs of languages that have not explicitly been trained together. Given a translation model exclusively trained on the translation directions $L_1 \to L_2$ and $L_3 \to L_1$, the model can translate $L_3 \to L_2$ without data for that language pair or explicitly training for that task. While this phenomenon has allowed for end-to-end translation among language pairs where no parallel data is available, its performance is still usually lower than pairs trained in a supervised fashion. Studies have shown [Gu et al., 2019] that spurious correlations between source and target languages from trained directions have a negative impact on zero-shot translation directions. Creating an information bottleneck between encoder and decoder [Pham et al., 2019] can help to reduce such correlations, improving zero-shot performance.

While fully shared architectures are robust and scale to massive amounts of languages and data [Aharoni et al., 2019], the models are limited by the representation capacity of the models, requiring larger models with more parameters as the numbers grow [Arivazhagan et al., 2019b, Fan et al., 2020]. With those limitations in mind, **partially shared parameters** are gaining popularity in recent times. These approaches combine the advantages of minimally and fully shared approaches, keeping zero-shot and transfer learning between languages while maintaining language-specific distributions. [Lu et al., 2018, Vázquez et al., 2019] proposed the use of language-specific encoders and decoders with a shared attention module that enforced a common language representation. Visualization of encodings show that languages are still clearly separated in the representation space. To improve the learning of such common representations, other works have proposed, in parallel with the work in this thesis, the use of auxiliary tasks in addition to the translation objectives. [Zhu et al., 2020] proposed the use of cosine distance between encoding representations to improve their similarity, while [Wang et al., 2020b] proposed the use of denoising autoencoding. All these approaches have a clear separation between the shared and language-specific components. [Zhang et al., 2021] proposed an architecture based on gated layers learned during training that automatically select which elements should be shared based on the input source sentence.

In this section, we have explained several methods to jointly train several translation directions. However, in recent years **language addition** to trained models has gained popularity, and several approaches have been proposed. Most of these approaches are based on the fully-shared architecture, being the most popular, even though it presents some limitations on this scenario. Fully sharing all network parameters implies that to add a new language, we have to modify all parameters or fine-tune the parameters of one of the languages already in the system with new data. Due to catastrophic forgetting, fine-tuning just with the desired data can lead to lower performance on the other languages. The initial models [Lakew et al., 2018] proposed a method based on vocabulary expansion for the new progressively added languages. Other works [Tang et al., 2020] tackle the performance loss on previous languages by fine-tuning the network with a combination of the new and original data, at the cost of increasing convergence time. [Lin et al., 2021] proposes computing language-specific masks that select which network parameters are more relevant for each language. While reducing the impact on the other languages but not eliminating it, this training requires a two-step training: first, training the entire network and second, fine-tuning the language-specific subnetworks. Another approach to add a new language is incorporating new modules to the system that limit the parts of the network affected by the new language. [Bapna and Firat, 2019] propose the use of feed-forward adapter networks between the shared encoder and decoder as language or domain-specific modules. Only the adapter's parameters are trained during training, leaving the original model untouched, preventing any catastrophic forgetting of previous tasks.

### 2.2.3 Spoken Language Translation

All the methods mentioned focus on translating between written languages, but text is not the only way humans use natural language to communicate; most communication happens through speech. **Spoken language translation** is the task of translating audio utterances of speech in a source language into a textual target language. The combination of both data modalities introduces new challenges to the ones described for MT. Audio is recorded as samplings given a fixed size sampling rate, usually 8 or 16 kHz, meaning that each second of the recording contains thousands of elements, leading to sequences usually orders of magnitude longer than their transcriptions and for each sample to contain much less semantic information than words or subwords used for text. In addition, unlike text, speech data usually does not include explicit word boundaries, letting the model learn them. Taking into account these characteristics, we can represent the input features of our model as two main families:

- **Contiguous Representations:** Such as raw recorded samples as shown in figure 2.5 right. Audio is expressed as sampled amplitudes over the temporal dimension. While this representation keeps the original sequence size, elements are dependent only on the temporal dimension, as in text.

- **Discrete Representations:** Computed by applying transformations over the utterances over fixed-sized windows of the waveform. Popular methods such as Mel Frequency Ceptral Coefficients (MFCC) [Davis and Mermelstein, 1980], and Perceptual Linear Prediction (PLP) [Hermansky, 1990] combine transformations to represent each window as a range of frequencies with representations based on the human auditory spectrum, e.g., Mel scale for MFCC. While these processes significantly reduce the temporal dimension, they may create 2D dependencies between the frequency features and time. . Figure 2.5 shows a plot example.

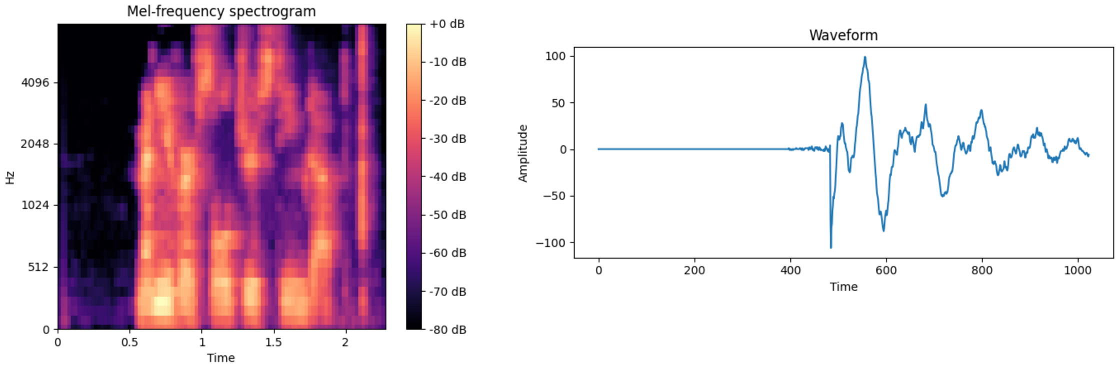

**Figure 2.5:** Speech input formats. Left: Mel-spectrogram. Right: Waveform.

The first models relied on dividing the problem into two subtasks

[Stentiford and Steer, 1990, Waibel et al., 1991], each one operating over one data modality. **Automatic Speech Recognition** (ASR) is the task that given a speech utterance $x$ computes a textual transcription $y$, both on the same language. While similar to MT, this task presents some differences. Transcription seeks the generation of the exact words from the spoken input without rephrasing or interpreting semantics. Consequently, this task's alignment is monotonic, as words are always produced in the original order. Once a transcription was generated, it could be translated to the desired target language using the MT models discussed. The advantage of this approach is benefiting from the more abundant ASR and MT data compared to ST. On the other hand, cascade methods suffer from propagating errors between models and using synthetic data instead of the actual language distribution. Several approaches have been proposed to mitigate this effect, such as the use of n-best transcriptions [Woszczyna et al., 1993] or lattices [Schultz et al., 2004], that allowed the exploration of different transcription options.

With the introduction of the Sequence-to-Sequence architecture and the publication of new datasets, end-to-end SLT systems have gained popularity. Several methods proposed architectures that rely exclusively on SLT data without using intermediate transcriptions. [Duong et al., 2016] proposed a model based on an RNN Sequence-to-Sequence with attention and PLP input features, while [Salesky et al., 2019] proposed the use of pre-computed phoneme boundaries to reduce speech sequence length. [Gangi et al., 2019b] adapted the Transformer Architecture to SLT using Mel Spectrograms as input and adding 2D attention to benefit from 2D dependencies on the data. Other models proposed multitask [Weiss et al., 2017] systems that combined the tasks of SLT, ASR, and MT in an end-to-end fashion. These methods allowed to leverage all the available data into a single model. [Liu et al., 2020b] used a shared encoder for SLT and NMT and boundaries computed by an auxiliary ASR task with CTC loss.

Similarly to the MT case, in recent years the interest for **Multilingual Spoken Language Translation** has increased with the release of several new datasets for the task [Wang et al., 2020a, Di Gangi et al., 2019, Iranzo-Sánchez et al., 2020]. [Inaguma et al., 2019] proposed a one-to-many approach based on RNN architecture. [Gangi et al., 2019c] applied Transformers to the same scenario. In parallel to the work in this thesis, [Li et al., 2021] combined a Wav2Vec 2.0 [Baevski et al., 2020] pre-trained speech encoder and an MBART [Liu et al., 2020a] pre-trained decoder on textual data, leveraging unlabelled data from both modalities.

### 2.2.4 Natural Language Inference

Assessing the generalization capabilities of NLP systems is not an easy task. Models can learn to generalize between languages without supervision [Pires et al., 2019] or rely on spurious correlations from the data [Gu et al., 2019]. Contextual encodings' visualization is a popular option [Johnson et al., 2017], but it is subject to dimensionality reduction techniques that could alter the distances between clusters on the final plots and to the set of sampled sentences that. These factors may difficult to extract meaningful conclusions when systems show similar performance. An alternative to analyze and compare the performance of our systems is the use of **probing tasks.** These tasks aim to test the generalization capabilities of systems on other tasks different from the ones used for training. These became more popular with the introduction of pre-trained large language models, [Peters et al., 2018b, Devlin et al., 2019] that showed state-of-the-art performance on a diverse set of tasks, becoming a popular tool to compare the performance of these systems with public benchmarks such as GLUE [Wang et al., 2018], and SUPERGLUE [Wang et al., 2019].

From these tasks, the most popular one to assess the cross-lingual capabilities of multilingual systems is **natural language inference** (NLI) [Bowman et al., 2015, Williams et al., 2018]. The task consists of given a reference sentence and a hypothesis sentence to inference their relation between three possible classes. The possible options are: *entailment*, if the reference sentence contains the hypothesis, *contradiction* if the hypothesis facts are not true for the reference sentence or *neutral* if the two are not related. Figure 2.6 shows an example of all three classes applied to a single reference sentence.

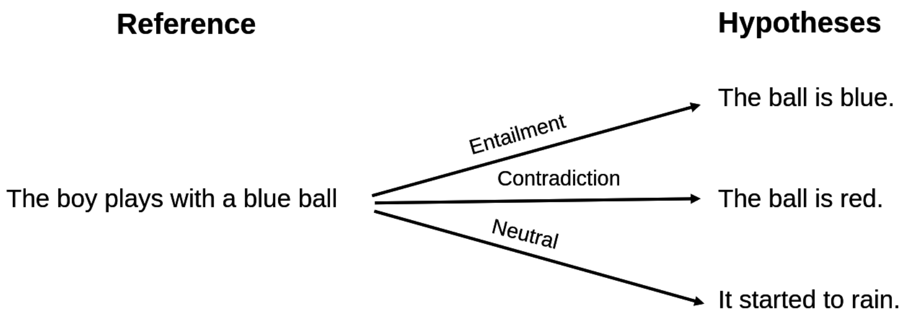

**Figure 2.6:** NLI example.

On the multilingual setting, a classifier is trained using English data and evaluated on a multilingual test set to measure the accuracy difference between the supervised direction and the rest of languages [Conneau et al., 2018]. Significant performance gaps may indicate a dissimilarity between the model's contextual representations for different languages. This approach has been applied to several pre-trained models [Conneau and Lample, 2019] and even to compare the multilingual representation learned by an MT model compared to a multilingual BERT large language model [Siddhant et al., 2020] showing that both models can exhibit comparable performance when trained on massive amounts of data.

# 3 Multilinguality by Incremental Training

Traditional Multilingual NMT systems rely on jointly training $N$ languages into a single system, with varying degrees of parameter sharing between languages (see Section 2.2.2). These architectures allow performing a maximum of $N^2 - 1$ translation directions by training at most $N$ language-specific encoder-encoders systems [Escolano et al., 2019a] or just one for universal encoder-encoder approaches [Johnson et al., 2017, Ha et al., 2016] that can scale to massive amounts of languages [Arivazhagan et al., 2019b].

The universal encoder-decoder presents several advantages, such as positive transfer learning from high resource to low resource languages, leading to significant performance improvements and zero-shot translation between never trained language pairs. On the other hand, the representation capacity of the shared parameters can become a bottleneck for the model's scalability to larger sets of languages [Arivazhagan et al., 2019b]. Parameter sharing creates a strong dependency between all languages, as all tasks depend on such parameters. Any modification for a subset of the task may harm the performance of other tasks or even lead to catastrophic forgetting, affecting the addition of new languages to the system or even fine-tuning a subset of the supported languages. Popular techniques to reduce this effect include retraining the model for the previous tasks [Chu et al., 2017] joint with the new ones or using Adapter modules [Bapna and Firat, 2019].

Work adapted from: **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2019a). From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Compu-tational Linguistics: Student Research Workshop* and **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2021). From bilingual to multilingual neural machine translation by incremental training. In *Journal of the Association for Information Science and Technology* p. 190-203

Positive transfer learning from high resource tasks has also been explored on bilingual NMT systems. Several systems have been proposed that benefit from previously trained models as initialization for new languages, using vocabulary expansion [Lakew et al., 2018] and knowledge distillation [Kim et al., 2019] to transfer knowledge between languages. These models still required retraining parameters from the previous task, either affecting its performance or completely forgetting it.

This chapter proposes a training strategy for bilingual NMT systems and efficiently extends them to a multilingual setting by incrementally training new languages. Our proposal aims to achieve the positive aspects of the previously mentioned architectures, zero-shot translation, and positive transfer learning while eliminating the limitations of such architectures due to parameter sharing.

In section 3.1 we propose a training strategy for bilingual training that combines translation, reconstruction, and correlation between parallel sentences to obtain common representation between both languages.

In section 3.2 we propose the incremental training of new languages without retraining of previous parameters, preserving the performance original translation directions. This approach is shown to offer a scalable strategy to new languages without retraining any of the previous languages in the system and enabling zero-shot translation.

In sections 3.3 and 3.4 we describe the experimental framework designed to test our approach on different amounts of data and languages, including zero-shot translation.

## Notation

Before explaining our proposed model we introduce the annotation that will be used. Languages will be referred as capital letters $L_0, L_1, L_2$ while sentences will be referred in lower case $l_0, l_1, l_2$ given that $l_0 \in L_0$, $l_1 \in L_1$ and $l_2 \in L_2$.

We consider as an encoder $(e_0, e_1, e_2)$ the layers of the network that given an input sentence produce a sentence representation $(h(l_0), h(l_1), h(l_2))$ in a space. Analogously, a decoder $(d_1, d_2, d_3)$ is the subset of layers of the network that, given the source sentence representation, can produce the tokens of the target sentence. Encoders and decoders will always be considered independent modules that can be arranged and combined individually as no parameter is shared. Each language and module has its weights independent from all the others present in the system.

## 3.1 Bilingual Joint Training

Due to the nature of neural networks, different training runs of the same architecture with the same data may lead to different results. Parameter initialization, random shuffling of the data batches are some factors that may lead that each network's parameters converge to a different local minimum, with its own hidden representation space. This section proposes a training schedule that enforces a shared representation between the two languages, serving as a platform for new languages to the system.

Given two languages, $L_0$ and $L_1$, our objective is to train independent encoders and decoders for each language, $e_0, d_0$ and $e_1, d_1$ that produce contextual representations $h(l_0), h(l_1)$ in a shared space. For instance, given a sentence $l_0$ in language $L_0$, we can obtain a representation $h(l_0)$ from the encoder $e_0$, than can be used to either generate a sentence reconstruction using decoder $d_0$ or a translation using decoder $d_1$.

The motivation to choose this architecture is the flexibility to add new languages to the system without modification of shared components and the possibility to add new modalities (i.e., speech) in the future as the only requirement of the architecture is that encodings are projected into the shared space. Our objective is to explore the viability of this objective purely enforced by the training schedule without parameter sharing.

With this objective in mind, we propose a training schedule that combines two tasks (auto-encoding and translation) and the two translation directions simultaneously by optimizing the following loss:

$$L = L_{L_0-L_0} + L_{L_1-L_1} + L_{L_0-L_1} + L_{L_1-L_0} + d \tag{3.1}$$

Where $L_{L_0-L_0}$ and $L_{L_1-L_1}$ correspond to the reconstruction losses of both language $L_0$ and $L_1$; $L_{L_0-L_1}$ and $L_{L_1-L_0}$ correspond to the translation terms of the loss measuring the token generation of each decoder given a sentence representation generated by the other language encoder (using the cross-entropy between the generated tokens and the translation reference), and $d$ corresponds to the distance metric between the representation computed by the encoders. This last term forces the representations to be similar without sharing parameters while measuring similarity between the generated spaces. We tested different distance metrics, such as different options of Minkowski distances or the use of a discriminator network that tried to distinguish the source language from their representations. We experienced a space collapse for all these alternatives in which all sentences tend to be located in the same spatial region, making them non-informative for decoding. Consequently, the decoder performs as a language model, producing an output only based on the information provided by the previously decoded tokens. Weighting the distance loss term in the loss did not improve the performance because for the small values required to prevent the collapse, the architecture did not learn an informative enough representation of both languages to work with both decoders. To prevent this collapse, we propose a less restrictive measure based on correlation distance [Chandar et al., 2016] computed as in equations 3.2 and 3.3. The rationale behind this loss is maximizing the correlation between the representations instead of enforcing the distance over individual values.

$$d = 1 - c(h(L_0), h(L_1)) \tag{3.2}$$

$$c(h(L_0), h(L_1)) = \frac{\sum_{i=1}^{n}(h(l_{0,i} - \overline{h(L_0)}))(h(L_{1,i} - \overline{h(L_1)}))}{\sqrt{\sum_{i}^{n}(h(l_{0,i}) - \overline{h(L_0)})^2 \sum_{i}^{n}(h(l_{1,i}) - \overline{h(L_1)})^2}} \tag{3.3}$$

where $X$ and $Y$ correspond to the data sources we are trying to represent; $h(x_i)$ and $h(y_i)$ correspond to the intermediate representations learned by the network for a given observation; and $\overline{h(X)}$ and $\overline{h(Y)}$ are, for a given batch, the intermediate representation mean of $X$ and $Y$, respectively. Figure 3.1 shows the interactions between network components, in red each term of the proposed loss function.

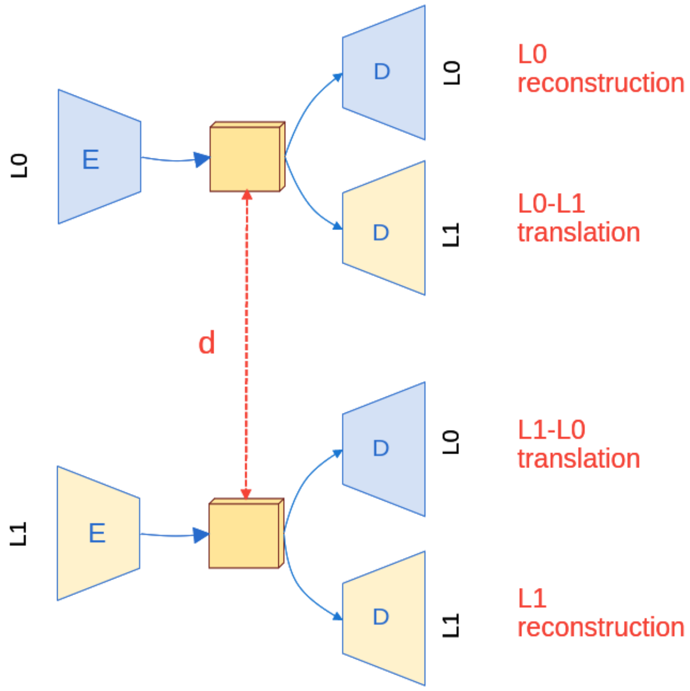

**Figure 3.1:** Joint training schedule.

## 3.2 Incremental training

Given the jointly trained model between languages $L_0$ and $L_2$, the following step is to add new languages to use our architecture as a multilingual system. Since parameters are independent between encoders and decoders, our architecture enables the addition of new languages without retraining the current languages in the system.

Let us say we want to add language $L_2$. To do so, we must have parallel data between $L_2$ and any language in the system. So, assuming we have trained $L_0$ and $L_1$, we need parallel data with either $L_2 - L_0$ or $L_2 - L_1$. For illustration, let us choose to have $L_2 - L_0$ parallel data. Then, we can set up a new bilingual system with language $L_2$ as source and language $L_1$ as target. To ensure that the representation produced by this new pair is compatible with the previously jointly trained system, we use the previous $L_0$ decoder ($d_0$) as the decoder of the new $L_2L_0$ system and we freeze it. During training, we optimize the cross-entropy between the generated tokens and the language $L_0$ reference data, updating only the language

$L_2$ encoder ($e_2$). Doing this, we train $e_2$ to produce good quality translations on the shared space learned during the joint training step without requiring an additional distance. The language $L_2$ sentence representation $h(L_2)$ is only enforced by the loss of the translation to work with the already trained module as it would be trained in a bilingual NMT system.

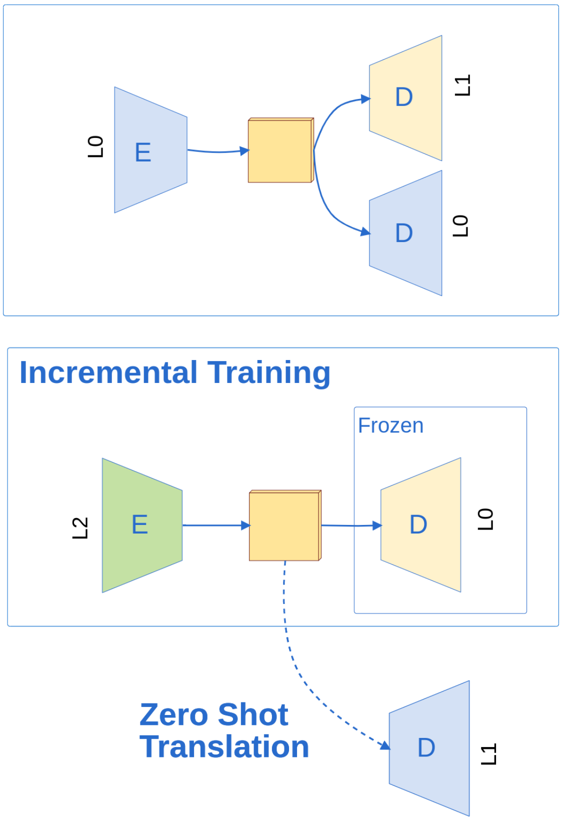

**Figure 3.2:** Language addition and zero shoot training scheme

Mapping the new language into this shared space means that the newly trained encoder $e_2$ can be used as input of the decoder $d_1$ from the jointly trained system to produce zero-shot $L_2$ to $L_1$ translations. See Figure 3.2 for illustration. This property proves that the devised training strategy can converge to the previous space while representing the most salient features for the task and that those can be preserved and shared to new languages just by enforcing the new modules to train with the previous one, without architecture modifications.

An additional property of our approach is that training new modules using only frozen modules is that, by definition, negative transfer learning to the jointly trained languages is not possible as weights are never updated during the incremental training step. Negative transfer learning is a common problem of previous architectures based on parameter sharing that are subject to their robustness to fine-tuning or limited to the use of Adapter modules for the new tasks.

## 3.3 Experimental Framework

Training data availability is one of the most critical factors during the training of any neural system. In addition, when working on machine translation, the choice of language can also have a significant impact as the relatedness of the languages can also significantly impact the final results. Having these two factors in mind, we propose the following three experimental scenarios for our approach:

- **Large dataset:** Joint training between English and Spanish with more than 15 million parallel sentences and incrementally adding French and German, both with more than 5 million parallel sentences with English available. All four are European languages, Spanish and French romance languages, and German and English Germanic languages, even though romance languages heavily influence the latter.

- **Intermediate dataset:** Joint training between Russian and English, with 6 million sentences available and incrementally adding Kazakh to the system with 4 million sentences available with Russian. All three languages are from different language families, with Russian and Kazakh sharing Cyrillic script.

- **Small dataset**: Joint training between Turkish and English with 200 thousand sentences available and incrementally adding Kazakh with 100 thousand parallel sentences with English available. Both Turkish and Kazakh are Turkic languages, even though Turkish is written using Latin script and Kazakh uses the Cyrillic script.

### 3.3.1 Dataset

Table 3.1 show de details of the datasets employed for all three proposed configurations. For all languages, preprocessing consisted of a pipeline of punctuation normalization, tokenization, corpus filtering of sentences longer than 80 words, and true-casing. These steps were performed using the scripts available from Moses [Koehn et al., 2007]. Preprocessed data is later tokenized into BPE subwords [Sennrich et al., 2016b]. We ensure that the vocabularies are independent and reusable when new languages are added by creating vocabularies monolingually, i.e., without having access to other languages during the code generation.

| Language Pair | Corpus | Language | Set | Segments | Words | Vocab (BPE) |
|---|---|---|---|---|---|---|
| ES-EN | EPPS and UN subset | ES | Training | $16.5 \cdot 10^6$ | $837.8 \cdot 10^6$ | |
| | | EN | | | $449.8 \cdot 10^6$ | |
| | newstest 2012 | ES | Validation | $3.003 \cdot 10^3$ | $137.7 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $80.8 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | ES | Test | $3 \cdot 10^3$ | $123.2 \cdot 10^3$ | |
| | | EN | | | $71.5 \cdot 10^3$ | |
| FR-EN | EPPS and UN subset | FR | Training | $18.9 \cdot 10^6$ | $586.7 \cdot 10^6$ | |
| | | EN | | | $510.9 \cdot 10^6$ | |
| | newstest 2012 | FR | Validation | $3.003 \cdot 10^3$ | $91.4 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $80.8 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | FR | Test | $3 \cdot 10^3$ | $82.1 \cdot 10^3$ | |
| | | EN | | | $71.5 \cdot 10^3$ | |
| DE-EN | EPPS and UN subset | DE | Training | $5.5 \cdot 10^6$ | $151.6 \cdot 10^6$ | |
| | | EN | | | $146.4 \cdot 10^6$ | |
| | newstest 2012 | DE | Validation | $3.003 \cdot 10^3$ | $95.3 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $84.8 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | DE | Test | $3 \cdot 10^3$ | $81.5 \cdot 10^3$ | |
| | | EN | | | $75.4 \cdot 10^3$ | |
| RU-EN | Yandex Corpus + ParaCrawl | RU | Training | $6.066 \cdot 10^3$ | $180 \cdot 10^3$ | |
| | | EN | | | $311 \cdot 10^6$ | |
| | Newsdev 2019 | RU | Validation | $4, \cdot 10^3$ | $117.3 \cdot 10^3$ | $31.8 \cdot 10^3$ |
| | | EN | | | $204.1 \cdot 10^3$ | $32 \cdot 10^3$ |
| | Newstest 2019 | RU | Test | $1 \cdot 10^3$ | $29.8 \cdot 10^3$ | |
| | | EN | | | $51.2 \cdot 10^3$ | |
| KK-RU | WMT19 Crawled Corpus | KK | Training | $4.18 \cdot 10^6$ | $284.9 \cdot 10^6$ | |
| | | RU | | | $109.1 \cdot 10^6$ | |
| | Ext. WMT19 Crawled Corpus | KK | Validation | $4 \cdot 10^3$ | $266.6 \cdot 10^3$ | $24.2 \cdot 10^3$ |
| | | RU | | | $102.2 \cdot 10^3$ | $34.6 \cdot 10^3$ |
| | Ext. WMT19 Crawled Corpus | KK | Test | $1 \cdot 10^3$ | $66.9 \cdot 10^3$ | |
| | | RU | | | $25.4 \cdot 10^3$ | |
| TR-EN | SETIMES | TR | Training | $200 \cdot 10^3$ | $5513 \cdot 10^3$ | |
| | | EN | | | $5467 \cdot 10^3$ | |
| | newsdev 2017 | TR | Validation | $1.001 \cdot 10^3$ | $26.7 \cdot 10^3$ | $16 \cdot 10^3$ |
| | | EN | | | $28.4 \cdot 10^3$ | $16 \cdot 10^3$ |
| | newstest 2017 | TR | Test | $3 \cdot 10^3$ | $84.6 \cdot 10^3$ | |
| | | EN | | | $86.6 \cdot 10^3$ | |
| KK-EN | News Commentary v14 | KK | Training | $100 \cdot 10^3$ | $2219 \cdot 10^3$ | |
| | | EN | | | $2216 \cdot 10^3$ | |
| | newsdev 2019 | KK | Validation | $1.033 \cdot 10^3$ | $35.5 \cdot 10^3$ | |
| | | EN | | | $36.7 \cdot 10^3$ | |
| | newstest 2019 | KK | Test | $1.033 \cdot 10^3$ | $36.7 \cdot 10^3$ | $7.36 \cdot 10^3$ |
| | | EN | | | $38 \cdot 10^3$ | $8.99 \cdot 10^3$ |
| KK-TR | OpenSubtitles | KK | Test | $2.594 \cdot 10^3$ | $22.5 \cdot 10^3$ | |
| | | TR | | | $26.3 \cdot 10^3$ | |
| KK-TR-EN | Tatoeba | KK | Test | 381 | $1.435 \cdot 10^3$ | |
| | | TR | | | $1.369 \cdot 10^3$ | |
| | | EN | | | $1.827 \cdot 10^3$ | |

**Table 3.1:** Size of the parallel corpora

### Large Dataset

Experiments are conducted using data extracted from the UN [Ziemski et al., 2016], and EPPS v7 datasets [Koehn, 2005] that provide 15 million parallel sentences between English and Spanish, German and French. *newstest2012* and *newstest2013* were used as validation and test sets, respectively. These sets provide parallel data between the four languages that allow for zero-shot evaluation. All four languages are tokenized into BPE subwords with 32 thousand merge operations.

### Intermediate Dataset

For experiments with larger datasets, we used the data shared between the Russian-English case used in WMT 2019[1] and that between the Russian-Kazakh case. The validation and test sets from the Russian-English case were extracted from the *Yandex* corpus. The validation set for the Russian-Kazakh case was extracted from *news-commentary-v14*. Finally, and only for visualization and analysis purposes, we extracted 381 multi-way parallel sentences in Turkish-Kazakh-English. These sentences were also extracted from the OPUS database[2]. For the latter, we downloaded the Turkish-English and the Kazakh-English datasets and matched the English sentences that were identical. Detailed statistics of the corpus are shown in Table 3.1. All three languages are tokenized into BPE subwords with 32 thousand merge operations.

### Small Dataset

For the experiments, we used the Turkish-English parallel data from *setimes2* [Tiedemann, 2009] which is used in WMT 2017[3] and the Kazakh-English parallel data from the news domain which is used in WMT 2019[4]. The training set for the Turkish-English data included approximately 200,000 parallel sentences, and the Kazakh-English data included approximately 100,000 parallel sentences. As de-

[1] http://www.statmt.org/WMT19/

[2] The datasets that we prepared for Kazakh-Turkish and Turkish-Kazakh-English, which are the only ones that do not belong to a benchmark, are freely available under request.

[3] http://www.statmachine translation.org/WMT17/

[4] http://www.statmt.org/WMT19/

velopment and test sets, we used *newsdev2016* and *newstest2016*, respectively, for the Turkish-English data, and *newsdev2019* was split into development and test sets for the Kazakh-English experiments. Additionally, we extracted the Kazakh-Turkish test set from the OPUS database [Tiedemann, 2012] to evaluate the zero-shot translation. Due to the small dataset size less merge operations were used for BPE subword tokenization, using 16 thousand operations for Turkish and English and 10 thousand for Kazakh.

### 3.3.2 Model Details and Training Setup

For both baseline and proposed systems, we used the transformer implementation provided by Fairseq[5]. We used 6 attention blocks with 4 heads, embedding/hidden dimensionality of 512, 1024 hidden feed-forward size, and 0.1 dropout. Learning rate was set to 0.0001 with inverse square root schedule and 4000 warmup updates. For all cases, we used Adam [Kingma and Ba, 2015] as the optimizer with 0.9 and 0.98 betas. The joint training was performed on two NVIDIA Titan X GPUs with 12 GB of RAM, while for the addition of languages, Titan X GPU was used. As early-stop criterion, the systems were trained until no improvement was observed on the validation set. All models have been trained using gradient accumulation with an effective batch size of 24000 tokens.

## 3.4 Results

This section will analyze the performance of our proposed approach on the three mentioned data configurations. Results would be presented in three main blocks, joint training of the initial bilingual system (Section 3.4.1), incremental training of new language (Section 3.4.2), and visualization of the encoder representations (Section 3.4.3).

[5] Release v0.6.0 available at https://github.com/pytorch/fairseq

### 3.4.1 Bilingual Joint Training Results

Our first experiment consists in comparing the performance of the jointly trained system to the standard Transformer. As explained in previous sections, this joint model is trained to perform two different tasks, auto-encoding and translation, in both directions. For all three tested configurations, the autoencoding task shows results above 96 BLEU, proving that the models effectively learn the task with minimal errors independently of the amount of data or the relatedness of the chosen languages.

More differences between configurations are observed when analyzing the machine translation results. Table 3.2 shows the obtained results on the large dataset of European languages. Results show that for both English to Spanish (EN-ES) and Spanish to English (ES-EN), the proposed architectures underperforms by approximately 2 BLEU points compared to the baseline bilingual systems. Similar results are observed on the intermediate configuration, Table 3.3, where both Russian to English and English to Russian directions underperform by 1 BLEU point when compared to their respective bilingual baselines. These show that similarity between the jointly trained languages does not play a significant role in these results, even between languages with significant alignment and grammatical differences such as English and Russian. The performance gap may be more likely explained by the differences between the combined tasks during training, such as monotonic and non-monotonic alignment between source and target, with might lead to negative transfer between tasks.

| **System** | **ES-EN** | **EN-ES** |
|---|---|---|
| Baseline | **32.60** | **32.90** |
| Joint | 29.70 | 30.74 |

**Table 3.2:** Joint training results measured in BLEU score for the large EN-ES dataset. Best results in bold.

| **System** | **EN-RU** | **RU-EN** |
|---|---|---|
| Baseline | **32.33** | **35.09** |
| Joint | 29.48 | 34.64 |

**Table 3.3:** Joint training results measured in for the intermediate RU-EN dataset. Best results in bold.

On the other hand, Table 3.4 shows that when applied to the small dataset configuration, the proposed approach outperforms the bilingual baseline by 0.71 BLEU on the English to Turkish (EN-TR) direction and by 0.51 on the Turkish to English direction (TR-EN). While the negative bias between tasks may still hold for this experiment, we observe that the proposed joint training took 6792 updates to converge on this configuration. In contrast, the bilingual EN-TR baseline took 4860 and the TR-EN baseline 4370, using the same regularization, indicating that when applied on limited data, the proposed approach may reduce overfitting to the training data, leading to better performance.

| System | EN-TR | TR-EN |
|---|---|---|
| Baseline | 11.85 | 14.31 |
| JointTraining | **12.56** | **14.82** |

**Table 3.4:** Joint training results measured in for the small TR-EN dataset. Best results in bold.

### 3.4.2 Incremental Training Results

Our second experiment consists of incrementally adding different languages to the system. Note that, since we freeze the weights while adding the new language, the order in which we add new languages does not impact performance. Table 3.5 shows the translation results for the high resource configuration. French-English direction performs 0.9 BLEU points below the baseline, and German-English performs 1.33 points below the baseline. French-English is closer to the baseline performance, which may be due to its similarity to Spanish, one of the initial system languages. It is also worth noticing that the added languages have better performance than the jointly trained languages (Spanish-English from the previous section). This reinforces the hypothesis that the auto-encoding task may have a negative impact on the translation task. While still behind the bilingual systems, results show that by training only the new language encoders, the approach can obtain comparable results on the high resource scenario.

| System | FR-EN | DE-EN | FR-ES | DE-ES |
|---|---|---|---|---|
| Baseline | **28.96** | **31.81** | - | - |
| Joint Training | 27.63 | 30.93 | - | - |
| Pivot | - | - | **29.09** | **21.74** |
| Zero-shot | - | - | 19.10 | 10.92 |

**Table 3.5:** Incremental training results measured in BLEU score for the large dataset. All blank positions are not tested or not viable combinations with our data. Best results in bold.

As previously mentioned, our joint strategy's objective is to serve as a platform that allows us to translate to the jointly trained languages even without explicit training. This section will evaluate the zero-shot performance of the approach between added languages and one of the jointly trained ones. On the large dataset scenario, German and French encoders are incrementally using the frozen English decoder. Table 3.5 shows the translation results compared to a cascade translation with English as a pivot language. Results show that while results are still far from pivot translation, similarly to other multilingual architectures, the proposed architecture can perform zero-shot translation for both studied languages. It is also observed that in both pivot and zero-shot translation, the selection of added language significantly impacts the final results, with better results between the romance languages, French and Spanish, than germanic languages even when the supervised French to English pair obtained lower scores.

| System | KK-EN |
|---|---|
| Baseline | 3.82 |
| Pivot Baseline | **16.62** |
| Pivot JointTrain | 13.69 |
| ZeroShot | 5.58 |

**Table 3.6:** Zero-shot translation (KK-EN) provided by our architecture compared to: a low-resource baseline (KK-EN), a pivotal system from KK-RU and from RU-EN, under both the baseline and our JointTrain architecture. Best results in bold.

Another advantage of shared architecture multilingual NMT systems [Johnson et al., 2017] is that they allow by parameter sharing the flow of information between high-resource and low-resource languages, leading to better results. As a result, the performance of the latter is improved. In this experiment, we want to confirm that our proposed architecture also presents this property by adding a new low resource language using incremental training. For both intermediate and small dataset scenarios, we will perform experiments adding an extremely low resource pair of Kazakh to English with only 100 thousand sentences and comparing with a bilingual baseline trained exclusively on this language pair. We added Kazakh to the already trained Russian decoder from the Russian-English trained system on the intermediate scenario. Table 3.6 shows how by incrementally training the low resource language with the frozen Russian decoder, we observe gains of 1.76 BLEU points with respect to the 3.82 points from the bilingual baseline system. On the small dataset scenario, we observe the same positive transfer when using the incremental training approach. Table 3.7 shows an improvement of 0.62 respect the bilingual baseline on the supervised Kazakh to English pair (KK-EN) and the Kazakh to Turkish zero-shot only a gap of 0.38 BLEU compared to the pivot results. These results show that this zero-shot approach outperforms both the baseline and direct Kazakh-to-English addition, proving that having access to more data provides better models in this architecture. This idea holds even for zero-shot translation directions.

| **System** | **KK-EN** | **System** | **KK-TR** |
|---|---:|---|---:|
| Baseline | 3.82 | Pivot Baseline | 4.74 |
| Incr. Train | **4.44** | Pivot Joint Train | **4.93** |
| | | ZeroShot | 4.36 |

**Table 3.7:** New supervised language (KK-EN) comparing: the baseline architecture to our added language (AddLang). Zero-shot translation (KK-TR) is provided by our architecture compared to a baseline which is a pivotal system from KK-EN and EN-TR. Best results in bold.

As an additional experiment, we also study the performance of jointly and incrementally trained modules on pivot translation. To do so, we translate the previous zero-shot translation by cascading using as pivot the one used as frozen decoder during the incremental training. Results show that, similarly to the joint training approach, only the model trained on the small dataset can outperform the bilingual baselines, which are correlated to the supervised directions used for the pivot system.

### 3.4.3 Common Representation Results

Our training schedule is based on training modules to produce compatible representations; in this section, we want to analyze this similarity at the last encoder's attention block. In order to graphically show the presentation, a UMAP [McInnes et al., 2018] model was trained to combine the representations of all languages. Figures 3.3 (A), (B) and (C) show 130 sentences extracted from the test set. These sentences have been selected to have a similar length to minimize the amount of padding required.

First, we will analyze the results of the high resource scenario. Figure 3.3 (A) shows the representations of all languages created by their encoders. Languages tend to be represented in clusters with no complete overlapping between languages as we would have liked. This mismatch in the intermediate representation is similar to what [Lu et al., 2018] reported in their multilingual approach, where authors argue that the language-dependent features of the sentences have a big impact on their representations.

However, since our encoder/decoders are compatible and produce competitive translations, we decided to explore the representations generated at the last attention block of the English decoder, and as shown in Figure 3.3 (B). We can observe much more similarity between English, French, and German (except for a small German cluster) and separated clusters for Spanish. These behaviors may be explained because French and German have directly been trained with the frozen English decoder and have been adjusted to produce representations for this decoder. The results also show that the proposed incremental training effectively forces similar representations between the frozen decoder and the new languages.

Finally, Figure 3.3 (C) shows the representations of the Spanish decoder. While around the center of the plot, several sentences show similar encodings for all languages, we can also observe clearly separated clusters between languages. Looking at the specific sentences that are plotted, we found that close representations do not correlate with better BLEU.

Similar conclusions can be extracted from the small configuration, even though it was the only one that outperformed the bilingual baselines in all tested directions. We used the 381 multiway parallel sentences extracted specifically for this analysis (see the statistics in Table 3.1). Figure 3.4 shows the sentence representations created by their encoders. The separated clusters show no overlapping between the language representations, as observed with the larger configuration. Related work [Arivazhagan et al., 2019a] shows similar results for the case of a multilingual system with shared encoder and decoder. While the system can produce representations that allow zero-shot translation, they form clear clusters for each language in the system. Plausible explanations for this difference may be the distance measure that we are using and/or the alignment of the source sentences. Some distance measures cause the representations to collapse in a small region of the space, making them non-informative for the decoder. Our distance measure, the correlation distance, enforces the representations to correlate, but it does not constrain the scale of the values in the contextual vectors. This measure enforces the sentence distribution within the same language to be similar between all languages. However, since we are not constraining the scale, each language can be represented in a different spatial region.

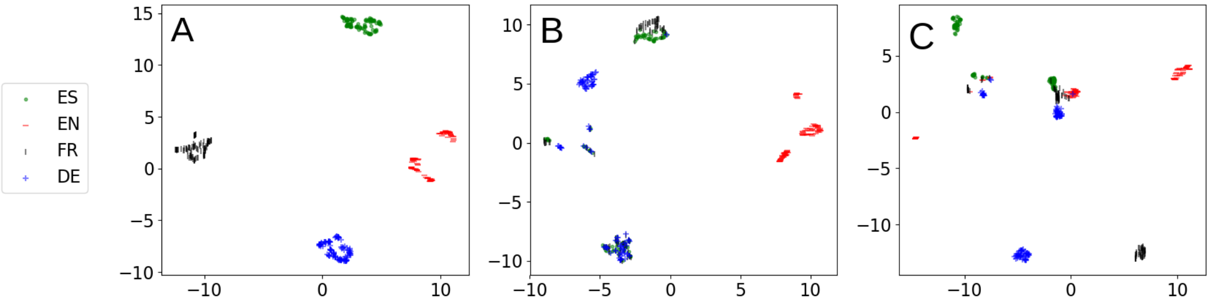

**Figure 3.3:** Plot A shows the source sentence representation of each of the encoder modules(ES,EN,DE,FR). Plots B and C show the representation of the target sentence generated by English (B) and Spanish (C) decoders given the sentence encodings of parallel sentences generated for all four language encoder modules.

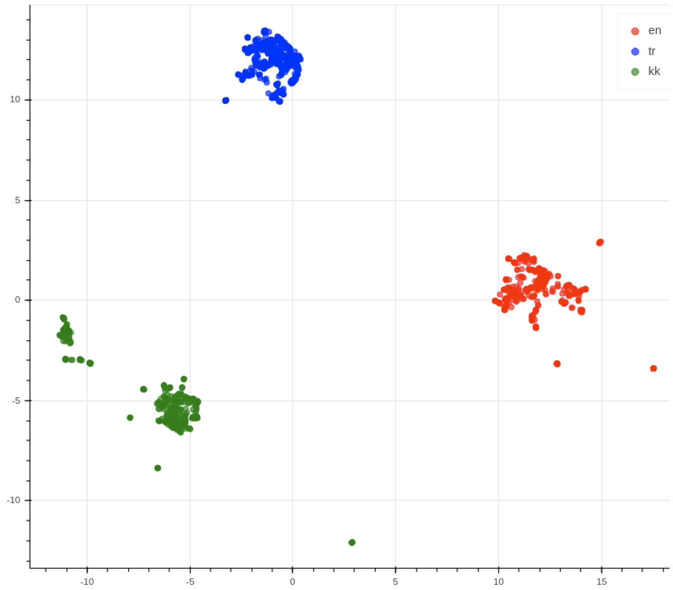

**Figure 3.4:** Plot show the encoder representations for jointly trained English (EN) and Turkish (TR) and incrementally trained Kazakh (KK)

## 3.5 Conclusions

This chapter reports a proof of concept of a bilingual system NMT which can be extended to a multilingual NMT system by incremental training. We have analyzed how the model performs for different languages. Experimental results show that by combining the tasks of autoencoding and machine translation, we can achieve representations that allow zero-shot translation between incrementally added languages. Experiments also show that training new language encoders to a frozen trained decoder can obtain positive transfer learning to low resource language pairs, as previously observed on multilingual systems, without the known negative transfer to high resource languages observed on architectures based on parameter sharing.

When applied to different data sizes and language families, we observe that the proposed approach can outperform bilingual baselines trained on the same data. Showing that the proposed training objective combining tasks may be beneficial to prevent overfitting on low resource scenarios, as convergence times indicate.

Finally, an analysis of the language representations learned by the systems shows that not a perfect alignment between parallel sentences is required to achieve zero-shot translation. By enforcing a less restrictive distance such as correlation distance, we can obtain this behavior without compromising the internal structure of the language representations.

# 4 Multilingual Joint Training

Multilingual NMT training usually requires parallel data between language pairs to translate. According to the proportion of available pairs, we can distinguish a variety of scenarios. Given $N$ languages, we denominate complete dataset, a dataset that includes parallel data between the $N^2$ - $N$ possible translation directions. On the other end, we denominate pivot dataset, a dataset including only parallel data between $N$ languages and a pivot language, usually English, to a total of $2N$ directions. These datasets benefit from the higher quantity of translated material to English while allowing systems to efficiently scale to a massive number of languages.

The use of pivot datasets has a significant impact on Multilingual NMT architectures and their development. On systems with language-specific encoders and decoders [Dong et al., 2015] we observe some differences according to desired tasks (see Section 2.2.2). It has been successfully applied on one-to-many and many-to-one tasks improving translation performance due to transfer learning. However, the many-to-many scenario failed to learn a cross-lingual mapping that allowed zero-shot translation between language pairs not observed during training. Several works have proposed the use of auto-encoding [Luong et al., 2016] or our bilingual joint training from Section 3.1, to enforce this common representation through an additional task. Other works [Firat et al., 2016a, Lu et al., 2018] have proposed sharing

Work adapted from: **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (2021). 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*, **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (2020b). Training multilingual machine translation by alternately freezing language-specific encoders-decoder. *Arxiv Preprint.*, **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2020a). The TALP-UPC system description for WMT20 news translation task:Multilingual adaptation for low resource MT. In *Proceedings of the Fifth Conference on Machine Translation*, and **Escolano, C.**, Costa-jussà, M. R., and Fonollosa, J. A. R.(2022). Multilingual machine translation: Deep analysis of language-specificencoder-decoders. *Accepted to the Journal of Artificial Intelligence Research.*

a common attention mechanism between that learned the cross-lingual mapping from language-specific representations. On the other hand, fully shared universal encoder-decoder systems [Johnson et al., 2017] do not exhibit these problems as sharing all parameters and token representations naturally enforce a common representation and transfer learning between languages at the cost of systems that are conditioned by the capacity of the model to support more languages [Arivazhagan et al., 2019b] and are less flexible once they are trained [Kudugunta et al., 2019].

As an alternative, recent works on using complete datasets [Freitag and Firat, 2020] show that its use on the universal encoder-encoder approach leads to better overall performance, especially on language pairs not involving the pivot language. Authors claim that these systems can outperform even cascade approaches through bilingual systems, which has traditionally been higher than multilingual translation performance.

This chapter extends the bilingual joint training proposed in Section 3.1 to a multilingual joint training between $N$ languages while preserving the incremental training capabilities from Section 3.2. Our proposal explores the use of complete training data on language-specific encoder-decoder Multilingual NMT and how this mitigates the gap between these approaches and universal encoder-decoders. We will follow the notation described in Section 3.

In section 4.1, we describe the differences between pivot and complete datasets and the impact it may have on our architecture.

In Section 4.2 we will propose a training schedule for multilingual joint training between $N$ languages without sharing parameters, focusing on the incremental training of new languages, both as source and target, to our system.

In Section 4.3 we propose an alternative schedule based on mimicking the incremental training from Section 3.2) by alternatively freezing encoders and decoders for a subset of the translation directions.

In Section 4.4 we propose a method to leverage monolingual data on an already trained language due to the modular nature of our approach.

In Section 4.5 we define the set of experiments to study the performance of our approaches on different languages and language-pair availability conditions as well as the learned cross-lingual mappings obtained on NLI (See section 2.2.4) probing task and visualization.

## 4.1 Complete Multilingual NMT

The training data available is a critical factor in any Machine Learning system, as it defines how the model converges and the overall performance of our system. For Multilingual NMT, data can become an even more important factor as the model supports several languages. Depending on our data availability, some translation directions may have to be learned indirectly as zero-shot translation. Figure 4.1 shows an example for four languages, where the arrows represent the translation directions with available parallel data. On the pivot datasets (right), the number of translation directions is much lower at the cost that most language pairs are only indirectly trained through the pivot language, resorting to cascade or zero-shot translation to perform those tasks. The lack of supervised training between languages may lead to attention mismatch between the non-pivot languages, requiring additional elements, such as parameter sharing, to enforce a common representation. On the other hand, the complete training approach (left) shows that each language in the systems is conditioned by all the other languages directly instead of relying exclusively on indirect relations. This prevents some common problems usually found on multilingual systems, such as the low performance for the non-English language pairs compared to a bilingual or pivot-based translation. Previous works [Freitag and Firat, 2020] have shown that even on architectures with full parameter sharing, complete training leads to better performance when compared to English-centric systems and bilingual baselines, especially on non-English translation directions.

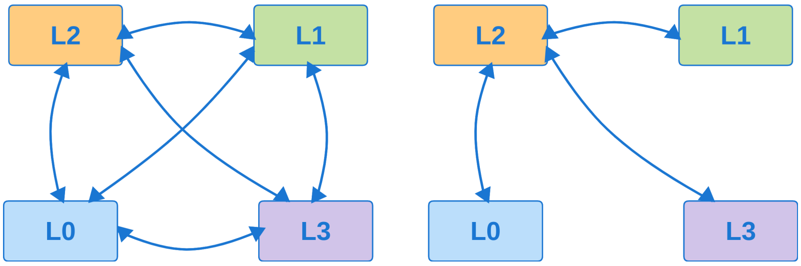

**Figure 4.1:** Complete training example (left) and pivot training example (right) with 4 languages.

Complete training also helps alleviate attention mismatch between the language representations learned by the system without requiring the addition of shared attention mechanisms to ensure a common representation space. These approaches would limit the language options for our systems in traditional systems, where all languages were jointly trained from the beginning. Following the approach from the previous chapter (see Section 3.2), we propose two approaches that focus on language-specific encoder-decoders and a two-step training. We schedule joint training between high-resource languages to establish a solid initial performance and incremental training for new languages, benefiting from positive transfer learning from previous languages.

We propose the use of a initial complete dataset to train language-specific encoders-decoders, eliminating the need for additional tasks or parameter sharing between languages. While this could have been a limitation on previous works focused on training all languages together, our incremental training step allows adding any new language to the trained language.

## 4.2 Language-specific Multilingual NMT

Our first proposed approach trains a separate encoder and decoder for each of the $N$ languages available, without parameter sharing across these modules. We denote the encoder and the decoder for the $i$th language in the system as $e_i$ and $d_i$, respectively. For language-specific scenarios, both the encoder and decoder are considered independent modules that can be freely interchanged to work in all translation directions. In what follows, we describe the proposed method in two steps: joint training and incremental training.

**Joint training** The straightforward approach is to train independent encoders and decoders for each language. The main difference from the standard pairwise training is that, in this case, there is only one encoder and one decoder for each language, which will be used for all translation directions involving that language. Following the common practice of sharing embedding parameters between source, target, and output projection, we propose the use of *tied* embeddings for each language. In this case, instead of sharing between source and target languages, parameters are shared between the encoder and decoder of the same language. Sharing the same token representation on directions that include any given language as source or target helps to learn better token representations. It enforces an additional link between languages to converge to a common representation space.

The training algorithm for this procedure is described in Algorithm 2. For each translation direction, $s_{i,j}$ in the training schedule $S$ with language $i$ as the source and language $j$ as the target, the system is trained using the language-specific encoder $e_i$ and decoder $d_j$.

**Algorithm 1** Multilingual training step

```
procedure MultilingualTrainingStep
    $N \leftarrow Numberoflanguagesinthesystem$
    $S = \{s_{0,0}, ..., s_{N,N}\} \leftarrow \{(e_i, d_j)\}$
    $E = \{e_0, ..., e_N\} \leftarrow Language - specificencs.$
    $D = \{d_0, ..., d_N\} \leftarrow Language - specificdecs.$
    for $i \leftarrow 0$ to $N$ do
        for $j \leftarrow 0$ to $N$ do
            if $s_{i,j} \in S$ then
                $l_i, l_j = get\_parallel\_batch(i, j)$
                $train(s_{i,j}(e_i, d_j), l_i, l_j)$
            end if
        end for
    end for
end procedure
```

Unlike the proposed approach from Section 3.1, it does not rely on jointly training tasks and correlation between language representations. Instead, using complete datasets, all encoders and decoders are combined to perform all translation directions involving their specific language, enforcing them to converge to a common representation space. This allows translation directions to be trained iteratively, reducing the computing power required to train several models in parallel.

Similarly, scalability to the number of languages differs from previous multilingual NMT approaches based on parameter sharing. For those architectures, the number of supported languages is bound by the capacity of the model [Arivazhagan et al., 2019b]. As a consequence, bigger models and more computational power are required for the task. While our proposed approach requires new encoders and decoders for each language, it does not use all of these parameters in parallel, only storing the parameters for the current language pair on GPU memory. Experimental results show that it can be trained with the same GPU computational requirements of a bilingual model, using additional storage that grows linearly with the number of idle encoders and decoders, which is considerably more affordable.

**Incremental training** Once we have our jointly trained model for $N$ languages, the next step is to add new languages. Since parameters are not shared between the independent encoders and decoders, language addition can be achieved by applying the proposed incremental approach from Section 3.2. We must note that in that section, the cross-lingual representation learned during the bilingual joint training (see Section 3.1) did not allow to incrementally train new languages as target, only new encoders. The proposed multilingual joint training approach allows the addition of both new source and target languages by incrementally training new encoders or decoders. Suppose we want to add language $N+1$. To do so, we must have parallel data between language $N+1$ and any language in the system. As an illustration, let us assume that we have $L_i - L_{N+1}$ parallel data. Then, we can set up a new bilingual system with language $L_i$ as the source and language $L_{N+1}$ as the target. To ensure that the representation produced by this new pair is compatible with the previously jointly trained system, we use the previous $L_i$ encoder ($e_{li}$) as the encoder of the new $L_iL_{N+1}$ system, and we freeze it. During training, we optimize the cross-entropy between the generated tokens and $L_{N+1}$ reference data but update only the parameters of to the $L_{N+1}$ decoder ($d_{l_{N+1}}$). By doing so, we train $d_{l_{N+1}}$ not only to produce good quality translations but also to produce similar representations to the already trained languages. Following the same principles, the $L_{N+1}$ encoder can also be trained as a bilingual system by freezing the $L_i$ decoder. See Figure 4.2 as a scheme for 4 languages ($L_0 ... L_3$) in the system and adding a fifth one ($L_4$) with parallel data to $L_0$. Once the new encoder or decoder is trained, zero-shot translation is possible between any language in the system, as the new module, as all encoders and decoders share the same representation space.

## 4.3 Frozen Training schedule

As discussed in the previous section, new languages are added into the system by learning a new encoder $e$ (or decoder $d$) with a frozen decoder $f(d)$ (or frozen encoder $f(e)$) already in the system. To simulate this setup in the initial training, we propose a modification of the joint training by alternately training encoders and decoders

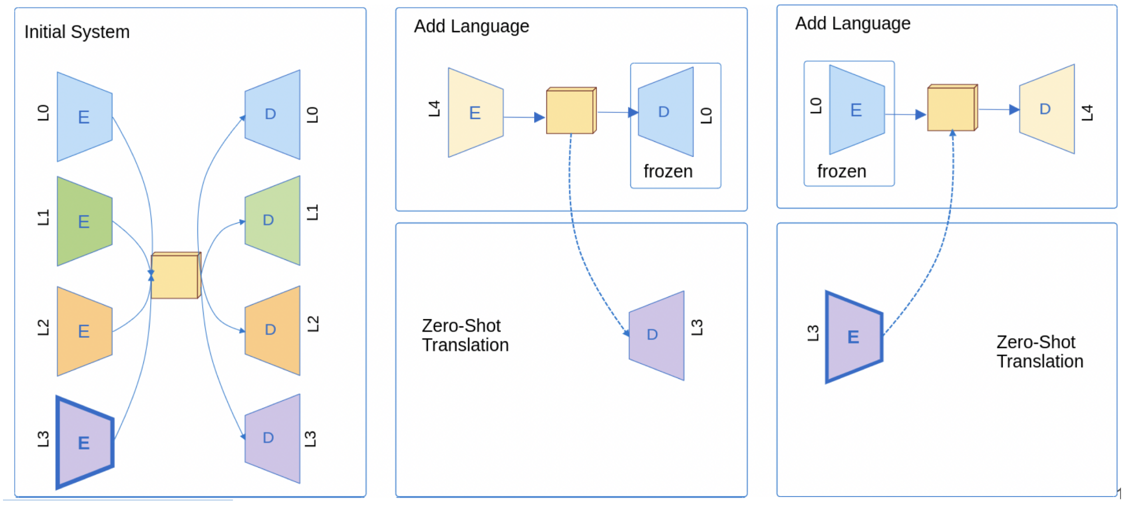

**Figure 4.2:** Block Scheme. (Left) Initial Joint Training. (Middle) Adding a new language in the source side with parallel data $L_0 - L_4$ and obtaining zero-shot translation from $L_4$ to $L_1, L_2, L_3$. (Right) Adding a new language on the target side.

while systematically freezing modules. For that purpose, we modify Algorithm 2 by adding new training schedules that define the frozen languages as follow:

**Algorithm 2** Multilingual frozen training step

```
procedure MultilingualTrainingStep
    $N \leftarrow Numberoflanguagesinthesystem$
    $S = \{s_{0,0}, ..., s_{N,N}\} \leftarrow \{(e_i, f(d_j)), (f(e_i), d_j), (e_i, d_j)\}$
    $E = \{e_0, ..., e_N\} \leftarrow Language - specificencs.$
    $D = \{d_0, ..., d_N\} \leftarrow Language - specificdecs.$
    for $i \leftarrow 0$ to $N$ do
        for $j \leftarrow 0$ to $N$ do
            if $s_{i,j} \in S$ then
                $l_i, l_j = get\_parallel\_batch(i, j)$
                $train(s_{i,j}(e_i, d_j), l_i, l_j)$
            end if
        end for
    end for
end procedure
```

We are still training all possible translation combinations among the $N$ languages (avoiding autoencoding and alternating batches in each direction).

We freeze the encoder or decoder for a subset of the combinations. When freezing $f(d_j)$, we effectively force the representation of $e_i$ to adjust to the rest of the encoders that $d_j$ learned from other translations, where it was not frozen. This principle holds because, if $e_i$ generated an incompatible representation, $f(d_j)$ would be unable to adapt to it given that it is frozen, which would increase the training loss for the direction. Similarly, freezing $f(e_i)$ allows $d_j$ to be more robust to a representation that has not been explicitly learned from it. See Figure 4.3 for an illustration of this with four languages for one training schedule.

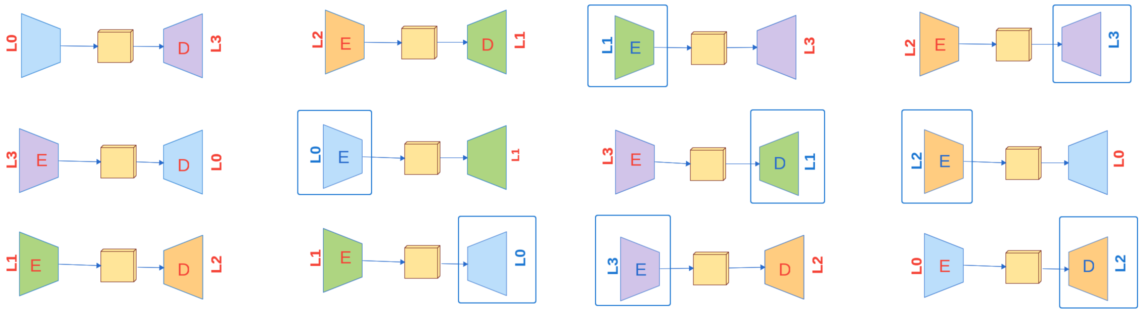

**Figure 4.3:** Alternate training of frozen encoders and decoders for 4 languages

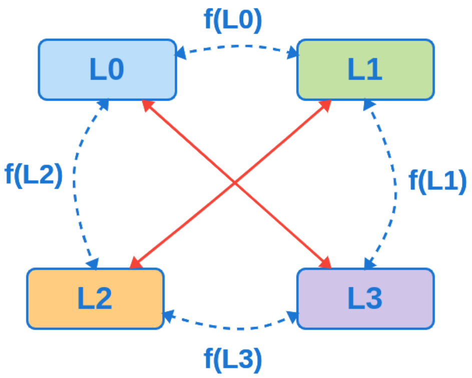

**Figure 4.4:** Frozen training schedule for 4 languages as directed graph.

Given a set of N languages, we can define a directed graph where each language is a node, and each edge is a training translation direction. Figure 4.4 shows the gradient flow between the different languages: $l$ means language; 0...3 are four different languages in the system; $f()$ means frozen and $n()$ non-frozen, dotted lines means frozen language pairs, continuous red lines mean non-frozen language pairs. Let's interpret the dotted arrow from box $l_0$ to $l_1$ with the scheme $f(l_0), n(l_1)$. When training the translation direction from $l_0$ to language $l_1$, we will freeze the $l_0$ encoder and only update the parameters of the $l_1$ decoder. For the opposite translation direction, we will freeze the $l_0$ decoder and only update the parameters of the $l_1$ encoder. This can be extended to translation pairs: ($l_1$,$l_3$); ($l_2$,$l_3$) and ($l_0$,$l_2$). For pairs ($l_1$,$l_2$) and ($l_0$,$l_3$) no language is frozen and therefore, there are continuous arrows in two directions because the gradient flows both ways. Note that the proposed schedule ensures that there are two different paths for any pair of languages from which information can flow during the training process by forming an Eulerian cycle between all these languages where one learns from another language and lets another learn from it.

Since different freezing schedules allow these properties, we study three different alternatives on this work, focused on freezing $\frac{n}{2}$ translation directions according to the following criteria:

- **Close schedule:** Chosen by linguistic similarity, the most similar languages are not frozen during training.
- **Far schedule:** Chosen by linguistic similarity, the least similar languages are not frozen during training.
- **Adapt schedule:** After each epoch, the languages with the highest validation loss are not frozen for the following epoch.

## 4.4 Monolingual Fine-tuning

The previous processes benefit from the additional corpus from the Multilingual NMT system, but as stated before, monolingual data is another common source of improvement for NMT systems. In this section, we will discuss how we added monolingual data to the previously described model. To employ monolingual data for language $L_i$, we define an autoencoder using the already trained $e_i$ encoder and $d_i$ decoder. Instead of training the autoencoder to reconstruct the input directly, we introduce an adaptor, between both modules, by stacking a decoder $d_j$ and an encoder $e_j$ from any other language on the system. This adaptor is responsible for processing the representation generated and mapping it into the common space learned by the jointly trained system. This is done by decoding $d_j$' to decode representation created by $e_i$ and encode it back with encoder $e_j$, to compute the reconstruction of the monolingual input using $d_i$. The motivation for this architecture is preventing the different objectives of the tasks of reconstruction and translation from harming performance, as observed in Section 3.4.

For our experiments, we greedy decoding as the autoregressive step was significantly time-consuming compared to the standard Transformer training. Figure 4.5 showcases how this process is applied. In this work, both encoder and adaptor were frozen, and only the final decoder was updated due to the non-differentiable decoding step.

In future iterations, decoder gradients could be propagated to the encoder, allowing the system to fully perform iterative back-translation [Hoang et al., 2018].

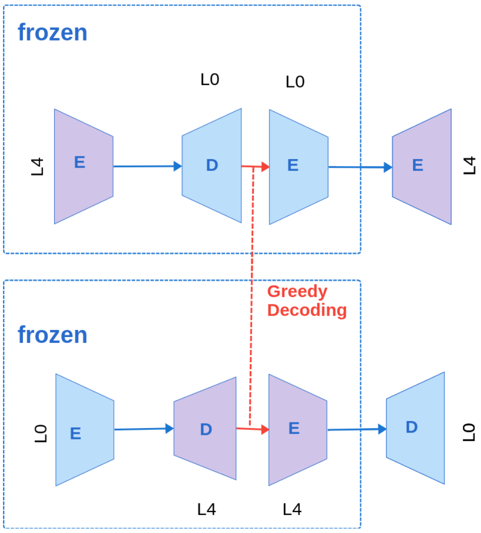

**Figure 4.5:** Training pipeline. Step 1 Supervised pretraining, Step 2 Unsupervised fine-tuning.

## 4.5 Experimental Framework

In this section, we are going to describe the experimental results of our proposed architectures. To do so, we are going to focus on two main tasks. machine translation to evaluate their performance on the supervised task, and natural language inference (see Section 2.2.4) to evaluate the shared representation space learned.

### Machine Translation

In this section, we aim to evaluate the translation performance of our multilingual system by two main objectives. Firstly, study the impact of the different proposed configurations on the translation performance, focusing on the use of tied embeddings and the different training schedules applied to the frozen approach. For experiments involving the frozen architecture, language pairs were selected according to their baseline validation loss as a measure of similarity:

- **Far schedule:** German-French and Spanish-English translation directions are never frozen during training. For all remaining directions, either the encoder or the decoder is frozen.

- **Close schedule:** German-English and Spanish-French translation directions are never frozen during training. For all remaining directions, either the encoder or the decoder is frozen.

- **Adapt schedule:** A new schedule is defined according to the current validation loss after each epoch starting from the Far schedule.

Secondly, analyze the proposed architectures based on language-specific encoders and decoders and determine their applicability compared to state-of-the-art fully shared (Shared) architecture. To do this, we designed a set of experiments to test several relevant scenarios for multilingual NMT systems:

- **Joint training**: The performance of the different approaches on the jointly trained systems.

- **Incremental training**: Measure the performance of our proposed approaches during incremental training compared to a Shared baseline and jointly trained languages, including the new language.

- **Low-resource and Monolingual Fine-tuning**: Measure the performance of our proposed LangSpec approach incrementally training a low-resource language and leveraging monolingual data.

- **Data Completeness:** As mentioned in Section 4.1, our proposed approaches rely on data completeness to converge to a common language representation space. With this experiment, we aim to analyze the robustness of our approach to the lack of some translation directions and how it compares to our fully shared baseline.

- **Fine-tuning robustness:** Performance loss due to fine-tuning is a known issue on shared systems [Kudugunta et al., 2019]. In this experiment, we aim to compare the performance of our approach without parameter sharing.

## Multilingual Representation

Another important aspect of our architectures is the cross-lingual mapping learned during training, necessary benefit from transfer learning towards incrementally trained languages and perform zero-shot translation. Following the same approach proposed in Section 3.4.3 we employ out visualization tool [Escolano et al., 2019b] to visualize the representations learned by the different models on two different points of the networks, the static embeddings, that represent a language both as source and target, and the contextual embeddings produced by our encoders averaged as sentence embeddings.

In addition to visualization of sentences representations, we introduce Natural Language Inference as a probing task to measure this mapping and evaluate how the different encoder layers produce this mapping. Following the procedure of [Conneau et al., 2018] a model is trained for the task using as input the encodings produced by our MT systems. As in the original work, the classifier consists of two fully connected layers with ReLU and Softmax activation. The classifier is fed with the following combination of the encoding of both reference and hypothesis:

$$h = [u, v, |u - v|, u * v] \tag{4.1}$$

Where $u$ is the reference encoding, $v$ is the hypothesis encoding, and $*$ is the element multiplication of both vector representations. Figure 4.6 shows an overview of this NLI architecture. In that work, encoders were explicitly trained on the task of natural language inference independently for each language, and representations were forced to share representation space through additional loss terms. In our experiment, the classifier is trained using only English data, without any additional loss term. This way, other languages' performance is only based on machine translation training. It can be considered a probing task of the quality of the cross-lingual mapping produced by our proposed translation architectures.

### 4.5.1 Datasets

For our task, we want to study the shared space already trained by the different configurations of multilingual machine translation systems. For each of them, a

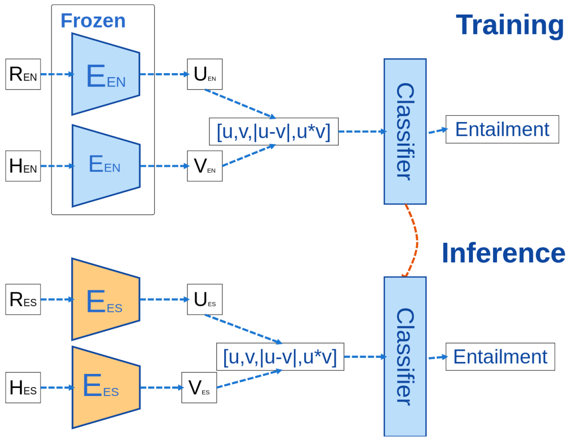

**Figure 4.6:** Experiment setup for NLI.

classifier is trained using its English encoder, which is frozen to help the classifier learn from the current shared space. To keep the encoding as described in equation 4.1 while using a Transformer encoder, the contextual embeddings are averaged to create a fixed-sized sentence representation. This approach was previously proposed by [Conneau et al., 2018], where pooling was employed to fix the representation size while not adding extra padding to the data. As a negative side, this creates an information bottleneck for the classification, as all sentence information has to be condensed into a single fixed-size vector, independently of the sentence's length.

Given that all language pairs in both language-specific architectures were trained to share sentence representations, we can evaluate the classifier's performance compared with all the other languages in the multilingual system without any extra adaptation.

| Language Pair | Corpus | Language | Set | Segments | Words | Vocab (BPE) |
|---|---|---|---|---|---|---|
| DE-EN | EPPS | DE | Training | $1.8 \cdot 10^6$ | $42.2 \cdot 10^6$ | |
| | | EN | | | $40,3 \cdot 10^6$ | |
| | newstest 2012 | DE | Validation | $3.003 \cdot 10^3$ | $92.3 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $72.9 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | DE | Test | $3 \cdot 10^3$ | $63.4 \cdot 10^3$ | |
| | | EN | | | $72.1 \cdot 10^3$ | |
| DE-ES | EPPS | DE | Training | $1.7 \cdot 10^6$ | $39.9 \cdot 10^6$ | |
| | | ES | | | $41.8 \cdot 10^6$ | |
| | newstest 2012 | DE | Validation | $3.003 \cdot 10^3$ | $92.3 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | ES | | | $89.9 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | DE | Test | $3 \cdot 10^3$ | $63.4 \cdot 10^3$ | |
| | | ES | | | $70.5 \cdot 10^3$ | |
| DE-FR | EPPS | DE | Training | $1.7 \cdot 10^6$ | $39.6 \cdot 10^6$ | |
| | | FR | | | $43.7 \cdot 10^6$ | |
| | newstest 2012 | DE | Validation | $3.003 \cdot 10^3$ | $92.4 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | FR | | | $93.1 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | DE | Test | $3 \cdot 10^3$ | $79.4 \cdot 10^3$ | |
| | | FR | | | $83.8 \cdot 10^3$ | |
| ES-EN | EPPS | ES | Training | $1.8 \cdot 10^6$ | $44.1 \cdot 10^6$ | |
| | | EN | | | $41.8 \cdot 10^6$ | |
| | newstest 2012 | ES | Validation | $3.003 \cdot 10^3$ | $89.8 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $81.5 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | ES | Test | $3 \cdot 10^3$ | $70.5 \cdot 10^3$ | |
| | | EN | | | $64.8 \cdot 10^3$ | |
| ES-FR | EPPS | ES | Training | $1.7 \cdot 10^6$ | $41.6 \cdot 10^6$ | |
| | | FR | | | $43.9 \cdot 10^6$ | |
| | newstest 2012 | ES | Validation | $3.003 \cdot 10^3$ | $89.9 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | FR | | | $93.0 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | ES | Test | $3 \cdot 10^3$ | $80.3 \cdot 10^3$ | |
| | | FR | | | $83.8 \cdot 10^3$ | |
| FR-EN | EPPS | FR | Training | $2.1 \cdot 10^6$ | $63.4 \cdot 10^6$ | |
| | | EN | | | $57.5 \cdot 10^6$ | |
| | newstest 2012 | FR | Validation | $3.003 \cdot 10^3$ | $93.1 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | | EN | | | $81.5 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | newstest 2013 | FR | Test | $3 \cdot 10^3$ | $83.8 \cdot 10^3$ | |
| | | EN | | | $72.1 \cdot 10^3$ | |
| RU-EN | Yandex Corpus | RU | Training | $9.27 \cdot 10^5$ | $18.5 \cdot 10^6$ | |
| | | EN | | | $24.7 \cdot 10^6$ | |
| | Newstest 2012 | RU | Validation | $3.003, \cdot 10^3$ | $81.6 \cdot 10^3$ | $31.8 \cdot 10^3$ |
| | | EN | | | $81.6 \cdot 10^3$ | $32 \cdot 10^3$ |
| | Newstest 2013 | RU | Test | $3 \cdot 10^3$ | $59.3 \cdot 10^3$ | |
| | | EN | | | $65.6 \cdot 10^3$ | |
| TA-EN | Yandex Corpus + ParaCrawl | TA | Training | $4.94 \cdot 10^5$ | $15.1 \cdot 10^6$ | |
| | | EN | | | $7.3 \cdot 10^6$ | |
| | Newsdev 2010 | TA | Validation | $1 \cdot 10^3$ | $117.3 \cdot 10^3$ | $16 \cdot 10^3$ |
| | | EN | | | $204.1 \cdot 10^3$ | $31.9 \cdot 10^3$ |
| | Newsdev 2020 | TA | Test | $1.275 \cdot 10^3$ | $66.6 \cdot 10^3$ | |
| | | EN | | | $29.7 \cdot 10^3$ | |
| TA | NewsCrawl | TA | Monolingual | $5.04 \cdot 10^5$ | $6.4 \cdot 10^6$ | $16 \cdot 10^3$ |
| EN | NewsCrawl | EN | Monolingual | $6.08 \cdot 10^5$ | $14.9 \cdot 10^6$ | $31.9 \cdot 10^3$ |

**Table 4.1:** Size of the parallel corpora

We used 2 million sentences from the *EuroParl* corpus [Koehn, 2005] in German, French, Spanish and English as training data, with parallel sentences among all combinations of these four languages (without being multi-parallel). For Russian-English, we used 1 million training sentences from the *Yandex* corpus[1]. As

[1] https://translate.yandex.ru/corpus?lang=en

validation and test set, we used *newstest2012* and *newstest2013* from WMT[2], which is multi-parallel across all the above languages. For English-Tamil *PMIndia* [Haddow and Kirefu, 2020], *Tanzil v1* [Tiedemann, 2012], *The UFAL EnTam corpus* [Ramasamy et al., 2012], The *NLPC UOM En-Ta corpus* [Fernando et al., 2020], *Wikimatrix* [Schwenk et al., 2021], and *Wikitiles* [Rozis and Skadins, 2017]. As monolingual Tamil data, we used *News Crawl* [Barrault et al., 2019], while for English, we used *News-commentary* [Tiedemann, 2012].

All non-Tamil data were preprocessed using standard Moses scripts [Koehn et al., 2007], applying punctuation normalization, tokenization, and truecasing. After these steps, all languages are tokenized at subword level using BPE [Sennrich et al., 2016b] with 32 thousand merge-operations. Tamil data has been tokenized at word-level using *Indic-NLP* [Kunchukuttan, 2020] and then tokenized with BPE with 16 thousand operations. Table 5.1 shows the details of all combinations after preprocessing. We evaluate our approach in 4 different settings: (i) the *initial* training, covering all combinations of German, French, Spanish and English; (ii) *adding* new languages, tested with Russian-English in both directions; and (iii) *zero-shot* translation, covering all combinations between Russian and the rest of the languages; (iv) Low-resource scenario and fine-tuning on monolingual data is performed on English-Tamil.

For the NLI task, we use the MultiNLI corpus [3] for training, which contains approximately 430k entries. We use the XNLI validation and test set [Conneau et al., 2018] for cross-lingual results, which contain 2.5k and 5k segments, respectively, for each language. All data is preprocessed following the same method described for English, Spanish, French, German, and Russian using the same BPE codes and vocabularies from the machine translation systems.

### 4.5.2 Model Details and Training Setup

All the experiments were done using the Transformer implementation provided by Fairseq[4]. We used 6 layers, each with 8 attention heads, an embedding size of 512 dimensions, 2048 hidden size feedforward layers. The dropout was 0.1 for the

[2] `http://www.statmt.org`

[3] https://cims.nyu.edu/ sbowman/multinli/

[4] Release v0.6.0 available at `https://github.com/pytorch/fairseq`

shared approach and 0.3 for language-specific encoders/decoders. All approaches were trained with an effective batch size of 32k tokens for approximately 200k updates with 0.0001 learning rate, using the validation loss for early stopping. As optimizer, we used Adam [Kingma and Ba, 2015] with 0.9 and 0.98 betas. All experiments were performed on an NVIDIA Titan X GPU with 12 GB of memory. For all systems (both shared and language-specific) we used *tied* embeddings.

When comparing shared and language-specific systems, we use the same number of parameters to perform each translation direction. Even though the language-specific systems have additional parameters for other languages, they are only used when their specific language is involved. Both in training and inference, all models use approximately 60.5 million parameters, slightly different due to each language's subword tokenization.

For the NLI experiments, we use the exact same encoders trained for the machine translation experiments, which are *not* further retrained or fine-tuned for this task. A classifier with 128 hidden units is exclusively trained on top of the English encoder, the only language we have training data available.

## 4.6 Results

In this section, we will evaluate the performance of our proposed language-specific approach (LangSpec) as well as our variant using frozen training schedules (Frozen). In contrast with our proposed approaches, the shared system requires retraining from scratch to add a new language. For that reason, we experiment with two variants of this system: one trained without Russian-English (Shared) and another one including this pair (Shared$^{RU}$). Note that, to make experiments comparable, we use the *Shared* version when comparing to our LangSpec and Frozen systems in Table 4.3, and the *Shared*$^{RU}$ version when adding new languages and performing zero-shot translation.

### 4.6.1 Joint Training Results

Our first experiments consists in comparing the translation performance of our proposed approaches during the joint training step. Before comparing their result, we will analyze the different training schedules for the frozen method. Table 4.2 shows the translation results for all translations directions between German, English, French, and Spanish. The *Far* configuration outperforms the *Close* schedule by 0.6 BLEU on average, and similar performance to the *Adapt* schedule, with a difference of only 0.05 BLEU. Looking at the individual language, we observe that the *Close* obtains small improvement on the non frozen pair German-English and English-French and Spanish-German, even though these improvements are not consistent in the reverse directions. These differences indicate that non-frozen pairs may obtain better performance individually, even when having lower overall performance. Finally, analyzing the frozen languages during the *Adapt* schedule training, we observed that it tended to converge to the *Far* schedule as the German-French directions showed the highest validation loss. This explains the similar overall results obtained by both configurations.

| | **Far** | | **Close** | | **Adapt** |
|---|---|---|---|---|---|
| de-en | 23.25 | n-f | 23.47 | n-n | **24.06** |
| de-es | **24.25** | n-f | 23.02 | f-n | 23.78 |
| de-fr | **25.08** | n-n | 23.46 | n-f | 24.00 |
| en-de | 19.55 | f-n | 20.88 | n-n | **21.30** |
| en-es | **28.60** | n-n | 27.42 | n-f | 27.97 |
| en-fr | 27.72 | f-n | **28.03** | f-n | 27.81 |
| es-de | 18.21 | f-n | **18.44** | n-f | 18.43 |
| es-en | **27.06** | n-n | 25.30 | f-n | 26.35 |
| es-fr | 29.34 | f-n | 28.93 | n-n | **29.92** |
| fr-de | **19.22** | n-n | 17.19 | f-n | 18.34 |
| fr-en | **25.11** | n-f | 24.31 | n-f | 24.91 |
| fr-es | **28.14** | f-n | 27.31 | n-n | 28.08 |
| *Avg.* | *24.63* | | *23.98* | | *24.57* |

**Table 4.2:** Initial training. In bold, best global results. In italic, average results for all translation directions.

With the previous results, we select the *Far* schedule to compare to our proposed language-specific (LangSpec) method and the fully shared baseline (Shared). Table 4.3 shows the results for all three configurations, with and without tied embeddings. Results show that, on average, the LangSpec configuration with tied embeddings outperforms both Frozen and Shared architectures with tied embeddings by 0.6

and 0.4 BLEU points, respectively, and obtaining the best results on 9 out of 12 translation directions. It is worth noticing that this was not the case for the non-tied configurations, with an improvement of 2.74 BLEU points on average by including tied embeddings, higher than the 0.65 BLEU obtained by the frozen approach.

| | **Shared** | | **LangSpec** | | **Frozen** | |
|---|---|---|---|---|---|---|
| | **¬Tied** | **Tied** | **¬Tied** | **Tied** | **¬Tied** | **Tied** |
| de-en | 24.40 | **25.04** | 22.04 | 24.54 | 23.25 | 23.76 |
| de-es | 24.04 | 25.01 | 22.38 | **25.02** | 24.25 | 24.66 |
| de-fr | 24.78 | 25.14 | 22.57 | 25.49 | 25.08 | **25,78** |
| en-de | 21.39 | 21.51 | 19.44 | **22.01** | 19.55 | 20.00 |
| en-es | 28.08 | 28.19 | 26.79 | 29.53 | 28.60 | **29.59** |
| en-fr | 28.43 | 28.67 | 26.94 | **29.74** | 27.72 | 28.56 |
| es-de | 19.51 | 20.21 | 17.70 | **20.31** | 18.21 | 18.84 |
| es-en | 26.66 | 26.93 | 24.9 | **27.75** | 27.06 | 27.64 |
| es-fr | 29.47 | 29.59 | 27.31 | **30.08** | 29.34 | 29.56 |
| fr-de | 19.22 | 19.81 | 16.88 | **19.97** | 19.22 | 19.80 |
| fr-en | 25.78 | 26.29 | 23.50 | **26.55** | 25.11 | 26.11 |
| fr-es | 28.15 | 29.03 | 26.78 | **29.07** | 28.14 | 28.98 |
| *Avg.* | *24.99* | *25.41* | *23.10* | ***25.84*** | *24.62* | *25.27* |

**Table 4.3:** Initial training. In bold, best global results. In italic, average results for all translation directions.

## 4.6.2 Incremental Training Results

Our next experiments focus on incrementally adding new languages and their zero-shot performance with jointly trained languages. Table 4.4 shows that when adding a new language into the system, language-specific encoder-decoders can outperform the shared architecture when adding a new language and a new domain by learning from the previous information in the frozen modules. LangSpec outperforms the shared architecture by 2.92 BLEU points for Russian-to-English and by 3.64 BLEU in the opposite direction. The frozen architecture also outperformed the Shared$^{RU}$ architecture by 1.43 BLEU on average while underperforming the Langspec architecture by 2 BLEU points on average. It is also worth mentioning that the Russian data is from a different domain than the frozen English modules used for training (*Yandex* corpus and *EuroParl*, respectively). These results are surprising as the English module (encoder or decoder) is always frozen during incremental training, preventing them from learning from the task. This indicates that the combination of language-specific parameters as well as embeddings could lead to better translation results on the incrementally trained supervised directions.

Additionally, retraining the shared encoder-decoder to add a new language took an entire week. In contrast, the incremental training with the language-specific encoder-decoders was performed in only one day.

On the other hand, when analyzing the zero-shot results, the shared encoder-decoder outperforms the LangSpec and Frozen approaches by 1.39 and 3.37 BLEU on average, respectively. This difference in performance suggests that, while limiting the amount of shared information during training can improve our model performance, it may also harm zero-shot translation. On the Shared architecture, representations are shared from the token embeddings to the contextual cross-lingual mapping learned by the attention layers. While this may limit representing new languages, it can also lead to better cross-lingual representations, improving the overall zero-shot performance.

| | **Shared**$^{RU}$ | | **LangSpec** | | **Frozen** | |
|---|---|---|---|---|---|---|
| | **¬Tied** | **Tied** | **¬Tied** | **Tied** | **¬Tied** | **Tied** |
| ru-en | 24.71 | 24.62 | 25.52 | **27.54** | 25.08 | 25.47 |
| en-ru | 19.91 | 20.03 | 21.44 | **23.94** | 21.33 | 22.01 |
| *Avg.* | *22.31* | *22.33* | *23.48* | ***25.74*** | *23.21* | *23.74* |
| ru-de | 15.36 | **16.52** | 12.73 | 13.77 | 11.85 | 13.11 |
| ru-es | 21.38 | **23.12** | 18.71 | 21.08 | 15.31 | 17.46 |
| ru-fr | 21.38 | **22.04** | 18.05 | 19.85 | 17.46 | 17.90 |
| de-ru | 16.23 | **17.27** | 14.39 | 16.99 | 14.99 | 15.38 |
| es-ru | 16.98 | **18.78** | 15.93 | 18.46 | 14.85 | 15.84 |
| fr-ru | 16.79 | **17.83** | 15.16 | 17.47 | 14.99 | 15.67 |
| *Avg.* | *18.02* | ***19.26*** | *15.99* | *17.94* | *14.91* | *15.89* |

**Table 4.4:** Adding a new language translation and Zero-shot.

## 4.6.3 Low-resource and Monolingual Fine-tuning Results

Our previous experiments have shown that our proposed approach effectively adds new languages and how the additional data can improve the overall performance after fine-tuning. This section's motivation is to explore the combination of both positive transfer and monolingual data in a low-resource task such as English-Tamil Translation and the impact of monolingual data on this scenario. To test our hypothesis, we trained a bilingual baseline with just the parallel data available for the task and compared its results to train a new language incrementally.

Tables 4.6 and 4.5 show that both directions benefit from adding Tamil into the LangSpec system with an improvement of 1.58 and 4.09 BLEU points, respectively, approximately a 40% improvement when compared to the bilingual baseline in both directions.

The monolingual fine-tuning results show that the English to Tamil translation direction benefits more (2.65 BLEU) from the technique than the Tamil to English direction (1.02 BLEU). This difference in the performance may be explained by the difference in the training of both decoders. While the Tamil decoder has been trained just with the parallel data for the task, the English decoder was trained with the multilingual NMT system with more data available, which may lead to a more robust representation.

| **System** | **BLEU** | **ΔBLEU** |
|---|---|---|
| Baseline | 3.42 | - |
| LangSpec | 5.00 | 1.58 |
| + Mono | 7.65 | 2.65 |

**Table 4.5:** Results measured in BLEU of the English to Tamil Translation direction.

| **System** | **BLEU** | **ΔBLEU** |
|---|---|---|
| Baseline | 6.51 | - |
| LangSpec | 10.6 | 4.09 |
| + Mono | 11.62 | 1.02 |

**Table 4.6:** Results measured in BLEU of the Tamil to English Translation direction.

### 4.6.4 Data Completeness Results

We have seen that our proposed LangSpec encoders-decoders does not suffer from attention mismatch as reported in previous research works [Firat et al., 2016a, Firat et al., 2016b, Lu et al., 2018] even if not sharing any parameter. We surmise that this is due to having parallel data in all language pairs from the initial system. Therefore, in this section, we exclude training data from certain language pairs to see how our system behaves. Beyond, learning the impact of attention mismatch, this experiment is motivated by the fact that there may be situations where complete parallel data among all the language pairs in the initial system is not available. We explore four situations (see Table 4.7):

1. *(EN)* including parallel data only with English (excluding parallel data from ES-FR, ES-DE, and DE-FR). This situation is equivalent to English-centered datasets.

2. *(EN+DE)* including parallel data with English and German (excluding parallel data from ES-FR).

3. *(EN+ES)* including parallel data with English and Spanish (excluding parallel data from DE-FR).

4. *(EN+FR)* including parallel data with English and French (excluding parallel data from DE-FR).

| | EN | | EN+DE | | EN+ES | | EN+FR | |
|---|---|---|---|---|---|---|---|---|
| | **Shared** | **LangSpec** | **Shared** | **LangSpec** | **Shared** | **LangSpec** | **Shared** | **LangSpec** |
| de-en | **24.40** | 24.35 | 23.92 | **24.63** | 22.22 | **24.07** | 23.03 | **23.96** |
| de-es | **24.04** | 0.32 | 23.98 | **25.16** | 22.72 | **24.74** | **22.35** | 22.21 |
| de-fr | **24.78** | 0.35 | 24.63 | **25.58** | **22.27** | 21.87 | 23.80 | **24.80** |
| en-de | 21.39 | **22.24** | 20.95 | **21.79** | 19.96 | **21.67** | 20.67 | **21.52** |
| en-es | 28.08 | **29.84** | 27.88 | **29.58** | 27.28 | **29.11** | 27.57 | **29.17** |
| en-fr | 28.43 | **29.99** | 28.11 | **29.72** | 27.83 | **29.29** | 28.25 | **29.17** |
| es-de | **19.51** | 0.11 | 19.62 | **19.73** | 17.90 | **19.84** | **18.53** | 16.50 |
| es-en | 26.66 | **27.15** | 26.52 | **27.53** | 24.78 | **27.20** | 26.09 | **26.89** |
| es-fr | **29.47** | 0.33 | **28.19** | 26.92 | 27.54 | **29.84** | 29.12 | **29.81** |
| fr-de | **19.22** | 0.16 | 18.76 | **19.34** | **17.37** | 16.06 | 18.14 | **19.08** |
| fr-en | 25.78 | **26.00** | 25.63 | **26.16** | 24.28 | **26.01** | 25.17 | **25.65** |
| fr-es | **28.15** | 0.21 | **27.39** | 26.65 | 27.13 | **28.86** | 27.76 | **28.56** |
| *Avg.* | ***24.99*** | *13.42* | *24.63* | ***25.23*** | *23.44* | ***24.88*** | *24.20* | ***24.86*** |

**Table 4.7:** Limiting training with parallel corpus from: pairs including English (*EN*), pairs including English and German (*EN+DE*), pairs including English and Spanish (*EN+ES*), pairs including English and French (*EN+FR*)

From the results in Table 4.7, we observe that for English-centered configuration, limiting training on language pairs to English, we see that our proposed methodology is not able to learn translation from the language pairs for which we do not have training data. For this particular case, we observe the attention mismatch mentioned in previous works. Consequently, the model cannot converge to an effective cross-lingual mapping between all languages. On the other hand, the shared architecture does not have this problem because, by nature, the shared parameters enforce a common representation between all input sentences. However, in situations where complete data is in this situation, we also observe that for the language pairs involving English, our proposed methodology can outperform the shared architecture by more than 2 BLEU points in all cases (except for fr-en, where we obtain 0.81 BLEU improvement).

On the other hand, for the remaining three situations where complete data for two languages are available, we observe that the performance of the language-specific encoder/decoders increases dramatically, and we do not observe the close to zero BLEU in zero-shot translation. Similar to situation 1 with the language pairs involving English, we see that the performance of our system in these cases is higher than that in the shared system (increasing up to 2.42 BLEU points in the case of ES-EN when lacking the DE-FR parallel corpus). The only exceptions are languages on the zero-shot translation directions, such as DE-ES and ES-DE, using complete data for all English and French language pairs.

### 4.6.5 Fine-tuning Robustness Results

| Effect | | **Shared**$^{RU}$ | | **LangSpec** | |
|---|---|---|---|---|---|
| | | | **ft** | | **ft** |
| Transfer | ru-en | 24.62 | 27.66 | 27.54 | **27.90** |
| | en-ru | 20.03 | 23.44 | 23.94 | **24.37** |
| | *Avg.* | *22.33* | *25.55* | *25.74* | ***26.14*** |
| Noise | de-en | **26.25** | 3.38 | 24.54 | **26.25** |
| | en-de | 22.11 | 1.99 | 22.01 | **22.72** |
| | es-en | 28.72 | 4.96 | 27.75 | **29.12** |
| | en-es | 29.78 | 1.83 | 29.53 | **30.53** |
| | fr-en | 27.98 | 5.33 | 26.55 | **28.24** |
| | en-fr | 29.63 | 1.72 | 29.74 | **30.33** |
| | *Avg.* | *27.41* | *3.20* | *26.68* | ***27.86*** |

**Table 4.8:** Fine-tuning results. Top table, the results after fine-tuning with Russian-English data. Bottom table, the results after fine-tuning with German-English data.

Previous work [Kudugunta et al., 2019] showed that overall translation performance might be affected when Shared architectures are fine-tuned on a subset of the supported languages. This section analyzes whether our proposed LangSpec architecture exhibits the same behavior when fine-tuned on an incrementally trained language and how robust the learned cross-lingual mappings are to new data. For our experiments adding the new Russian-English pair, we have new data from English that the English encoder/decoder already in the system has not seen. We want to know the impact on translation quality when fine-tuning the English encoder/decoder on these data. Basically, we simultaneously update the Russian encoder/decoder and English encoder/decoder for the language-specific case. It is important to note that the fine-tuned Russian modules are already trained using incremental training to enforce them to learn the system's cross-lingual representation. For the shared case, we update the shared encoder/decoder with the new data. As expected, Table 4.8 shows how this fine-tuning benefits the Russian-English performance and harms the other directions dramatically in the case of the shared encoder/decoder. However, for language-specific encoder/decoder, fine-tuning benefits all pairs of languages. Note that Table 4.8 (top) reports variations only on the results involving English modules, which are the ones modified by this fine-tuning. This fine-tuning has a double impact on the entire system. First, it is doing inductive transferring for the Russian-English and, therefore, improves its translation quality. Second, it adds noise/interference to the other language pairs in the system. By showing that we can fine-tune and not lose performance, we are proving that we are learning a robust intermediate space that is not forgotten by the perturbations on individual modules. These results show how the representation created by languages added to the system is more robust to catastrophic forgetting than the one obtained by the shared training.

### 4.6.6 Common Representation Results

To better understand our models' cross-lingual mapping, we will visualize both the static subword embeddings and the contextual encoding representations produced by our models. We use [Escolano et al., 2019b] tool[5], that allows us to visualize intermediate sentence representations. The tool performs a dimensionality reduction of these data using UMAP [McInnes et al., 2018]. Figure 4.7 shows static subword embeddings from the intermediate representation of 100 sentences in each language (German, English, Spanish, French, and Russian). Figure 4.7 shows that both LangSpec and Frozen approaches (center and right) create defined language-specific clusters without significant overlapping between languages. On the other hand, the Shared architecture (left) shows that all languages overlap in a single common space. This was expected as all languages share the same embedding table, with languages even sharing an important number of subwords.

[5] https://github.com/elorala/interlingua-visualization

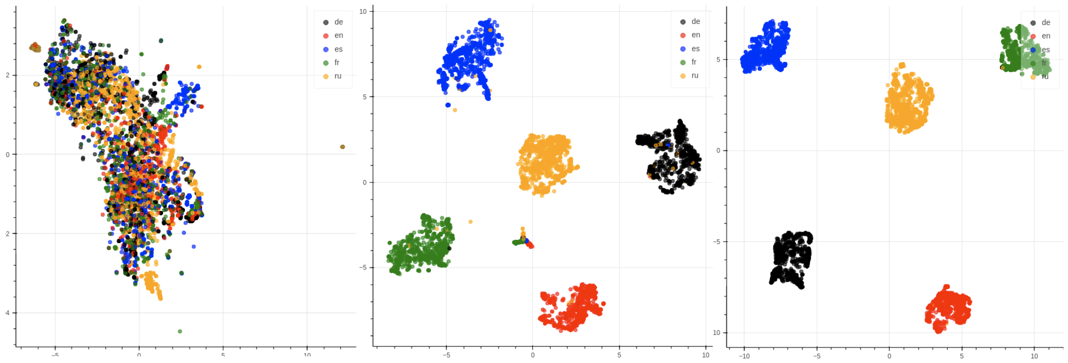

**Figure 4.7:** Subword embeddings for 100 test sentences, Shared$^{R}U$(left), LangSpec(center) and Frozen(right) systems.

To observe the contextual encoder representation, we compute sentence vectors as the average of all tokens contextual embeddings without padding. Figure 4.8 shows the visualization for all three architectures. The differences between systems results are much more subtle at this level, with clear overlapping between languages for all the architectures. These results show that our language-specific approaches create language-specific representations at subword level that are transformed into a shared space through the different encoder layers.

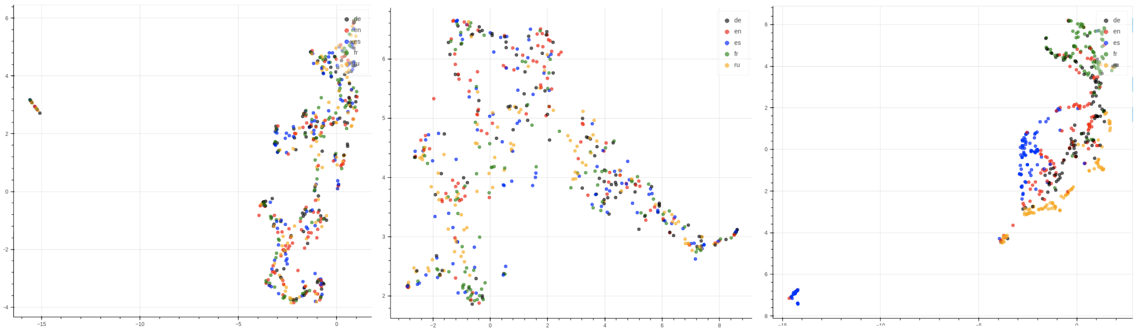

**Figure 4.8:** Sentence encoding visualization for 100 test sentences, Shared$^{R}U$(left), LangSpec(center) and Frozen(right) systems.

As all three visualization show similar overlapping results between languages, we performed additional analysis on NLI as a probing task to better compare the quality of these representations. In this case, we will Table 4.9 shows the results for the XNLI tasks for the output of different encoder layers for the language-specific encoder-decoders, using the LangsSpec approach. Note that our goal is not to improve the state-of-the-art in this task but rather to analyze the nature and quality of the cross-lingual representations arising in our proposed multilingual architecture to understand it better. Better performance is generally achieved at the highest layer (6), except for French and Russian. This may imply that better sentence representations may be achieved with more layers. These findings correlate with the observed visualizations, with language-specific input representations better aligned by the encoder's output.

| 2* | **Encoder layers** | | | | | |
|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** |
| en | 57.50 | 57.30 | 58.43 | 58.62 | 58.82 | **59.52** |
| de | 43.42 | 44.70 | 49.00 | 51.51 | 51.83 | **54.49** |
| es | 45.60 | 47.00 | 52.23 | 54.10 | 55.06 | **55.71** |
| fr | 44.90 | 44.20 | 52.11 | 55.71 | **57.36** | 54.81 |
| ru | 36.10 | 33.30 | 33.40 | 35.90 | **43.80** | 38.94 |

**Table 4.9:** XNLI (en, de, es, fr, ru) results according to the number of encoder layers

To better illustrate the model's performance, table 4.10 show the performance of the proposed methods compared to the shared system with and without Russian. Results show that the Frozen approach leads to better performance on the supervised English direction, by 0.2% over the Shared$^{R}U$ approach and 5.8% over LangSpec. On the Zero-shot languages, results show that the Shared approaches outperform the language-specific ones. We observe similar performances when comparing the two language-specific approaches, with the Frozen approach outperforming LangSpec by 2.4% on the incrementally trained Russian. These results indicate that the Frozen approach may positively impact the learned cross-lingual mapping, but it does not correlate with the translations results. We hypothesize that machine translation may benefit from some language-specific information or spurious correlations between supervised language pairs.

It is also noticeable that when comparing the performance of the shared model, it benefits from the additional data used for training, showing better results in all language pairs. The language-specific model in incremental training does not show this behavior, as the weights from the previous languages are frozen.

| | **Shared** | **Shared$^{RU}$** | **LangSpec** | **Frozen** |
|---|---|---|---|---|
| en | 58.32 | 59.96 | 54.49 | **60.20** |
| de | 59.94 | **62,15** | 59.52 | 57.80 |
| es | 58.40 | **60.59** | 55.71 | 55.60 |
| fr | 59.19 | **60.60** | 54.81 | 57.90 |
| ru | - | **55.98** | 38.94 | 41.40 |

**Table 4.10:** XNLI (en, de, es, fr, ru) accuracy comparison.

## 4.7 Conclusions

In this chapter, we presented two methods to train language-specific encoders-decoders multilingual systems, LangSpec and Frozen, that allow incremental additions of new languages in the system without retraining the entire system or adding any adapter. We believe that this approach can be particularly useful for situations in which a rapid extension of an existing machine translation system is critical. Our experiments show that training our systems on complete data, including all translation directions, effectively enforces a cross-lingual mapping.

On multilingual joint training, the language-specific encoder-decoders LangSpec outperform the shared architecture by 0.4 BLEU points on average. When adding a new language, the language-specific encoder-decoders outperform the shared ones by 3.4 BLEU points on average and by a fraction of the time required to retrain a complete architecture. Furthermore, without any variation in the quality of languages in the initial system when adding a new language. Our experiments also show encoders can be further trained using only monolingual with 1 BLEU points improvement on low-resource Tamil-English translation tasks.

Further analysis of our model's robustness to fine-tuning and different data conditions shows that the LangSpec approach is robust to fine-tuning, not only not showing catastrophic forgetting but also improving the performance of other translation directions involving the fine-tuned language. On the other hand, on the different data conditions tested, results show that when trained on pivot language-based data, our approach cannot perform zero-shot translation to other languages, while the Shared approach does. Moreover, we do not need parallel data among all language pairs in the initial system to learn translations from and to all languages; however, we at least need parallel data with more than one language. In this sense, language-specific encoders-decoders could take further benefit from incremental training with more than one language in the initial system.

We also examine the quality of the intermediate cross-lingual representation created with our proposed model in applying natural language inference and visualization. We observe that our approaches create language-specific subword embeddings that evolve into a shared cross-lingual mapping through the encoder layers. When compared to the shared system, we observe that parameter sharing provides better cross-lingual representations for the probing task, which correlates with the difference in the performance of the systems on zero-shot translation.

# 5 Multimodality by Incremental Training

End-to-end SLT systems are usually trained on a combination of ASR for pretraining, and SLT data [Gangi et al., 2019b]. This can become a limitation as annotated data for those tasks can be scarce or limited to high-resource languages. A common approach is using cascade systems [Sperber and Paulik, 2020] where the output of an ASR model is fed to an NMT system, leveraging both ASR and NMT data, which is more available in terms of the number of sentences and language availability. On the other hand, while cascade approaches incorporate NMT data, they require an intermediate decoding step, which is time-consuming and may propagate errors between models.

Data availability is even more important in multilingual systems, where annotated data is required between several languages. In recent years end-to-end Multilingual SLT has gained popularity with systems able to perform on the one-to-many [Gangi et al., 2019c], and many-to-many [Gangi et al., 2019c] scenarios, even allowing for zero-shot translation between SLT directions. With the release of large pretrained models on unlabelled data, recent works [Li et al., 2021] have proposed fine-tuning a combination of a Wav2Vec 2.0 [Baevski et al., 2020] speech encoder with an MBART [Liu et al., 2020a] text decoder on Multilingual SLT data. This leads to significant improvements compared to a randomly initialized system, as the model can benefit from large amounts of unlabelled data. A common trait of all these approaches is that they only provide zero-shot translation between languages

Work adapted from: **Escolano, C.**, Costa-jussà, M. R., Fonollosa, J. A. R., and Segura, C. (2021). Enabling Zero-shot Multilingual Spoken Language Translation with Language-Specific Encoders and Decoders. In *Proceedings of ASRU 2021* ©2021 IEEE, trained on the SLT task, which are still limited, without benefiting from NMT data available.

This chapter proposes the first end-to-end method that performs zero-shot SLT end-to-end between a speech encoder and a text decoder trained exclusively on NMT data, translating between language pairs where no SLT data is available. Our proposal consists in extending our LangSpec architecture from Section 4.2 by incremental training (see Section 3.2), focusing on the adaptations required to bridge both speech and text data modalities. Figure 5.1 depicts this behaviour.

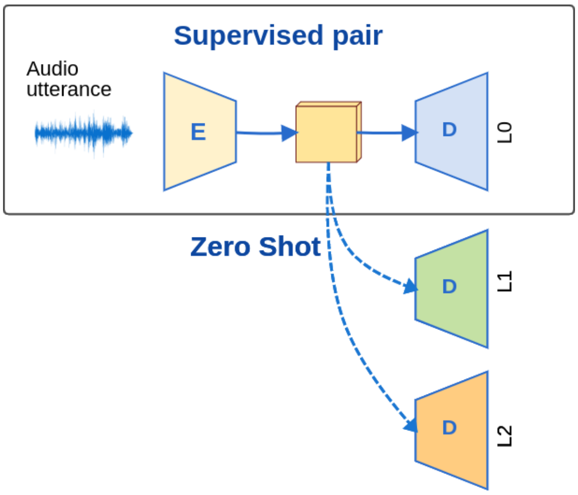

**Figure 5.1:** Diagram showing the incrementally trained speech encoder and zero-shot SLT directions.

In Section 5.1, we describe the speech encoder architecture used for our experiments and the adaptations required to use in combination with a previously trained NMT decoder.

In Section 5.2, we propose a method to incrementally train the LangSpec system with the speech encoder and how to perform zero-shot translation to all languages supported by the system.

In Section 5.3 we define the set of experiments performed, datasets, focusing on the translation performance, including zero-shot, and representations space learned by the different configurations.

In Section 5.4, we analyze the results obtained by the tested models. We analyze the learn encoder representation focusing on the observed differences between modalities and the impact of the Adapter modules.

## 5.1 Speech Encoder

Different data modalities have different properties that have to be addressed by our systems to perform correctly. As explained in Section 2.2.3, some of the most important differences we may find between text and speech are sequence length and the possibility of 2D dependencies, as is the case in our experiments. We use log mel spectrograms that deduce the temporal dimension of the input data, even though they are still an order of magnitude longer than its transcriptions, while introducing 2D dependencies between the frequency and temporal dimensions of our utterances.

Being the Transformer [Vaswani et al., 2017] an architecture designed for text, some modifications are required for speech processing. In this work, we use the S-Transformer architecture [Gangi et al., 2019b] as speech encoder for our approach. This architecture introduces two 2D-convolutions over the input representation to reduce by 4 the temporal dimension, making it closer to the sequence length of the correspondent transcription or translation target. To model 2D dependencies, they introduce a 2D self-attention that computes attention for the temporal and frequency dimensions in parallel by applying multi-head attention (Section 2.1.3.1) over the input and its transposed matrix, respectively. Queries $Q$, keys $V$, and values $V$ are computed by applying a 2D convolution with as many output channels as the number of heads. Dot product attention is also modified to induce the model to attend to short dependencies by introducing a length penalty $\pi(D)$ as shown in equation 5.1 where $D$ is a distance matrix where each $d_{i,j} \in D$ is the distance between positions $|i - j|$ and $\pi(d_{i,j})$ is defined as 0 if $d_{i,j}$ equals 0 and $log_e(d_{i,j})$ otherwise.

$$Attention(Q, K, V) = softmax(\frac{QK^t}{\sqrt{d} - \pi D})V \tag{5.1}$$

To adjust this speech encoder to our LangSpec approach (see Section 4.2), we use an extra 2D-convolutional layer in order to reduce the length of the input speech further, resulting in a total of 3 2D-convolutional layers, as in previous works [Hannun et al., 2019]. Each convolution, with kernel size 3 and stride 2, halves the input spectrogram's temporal dimension, resulting in an 8 times shorter sequence, obtaining sequences 2 times longer than the text target on average, measured on the validation set.

## 5.2 Incremental Training for Speech Encoders

Following the method described in Section 3.2 speech encoders are added by training them with a frozen decoder from the LangSpec system from Section 4.2. First, speech encoders are pretrained on the task of automatic speech recognition as proposed by [Gangi et al., 2019b]. For this step, both encoder and decoder are randomly initialized and trained from scratch. Secondly, we pair the pretrained speech encoder with a frozen decoder from one of the multilingual machine translation system languages. Even though the speech encoder is modified for the task, our decoder cannot be modified during training, which may harm the final model's performance in combination with the modality differences. To bridge both representation we add Adapter [Houlsby et al., 2019], randomly initialized, between speech encoder and text decoder. Previous works [Bapna and Firat, 2019, Artetxe et al., 2020] have used this technique to finetune a frozen model for specific languages or tasks. Results show that this method provides a lightweight mechanism to adapt representations to new tasks by only training a small portion of the model's total number of parameters. This module consists of layer normalization followed by a projection step $ff_1$ with a ReLU non-linear activation. This step projects the encoder's hidden representation into a different dimensionality space projected back to the original dimensionality by a second feedforward layer $ff_2$, being the projection dimensionality the only hyperparameter of the module to tune, allowing a fast tuning for each language. According to this projection size, two main tasks can be performed by the module. By down projecting the input representation to a smaller dimensionality, an information bottleneck is created, enforcing the model to focus on the most salient features for the task. By projecting to a higher dimensionality space, the module tries to capture new information in a richer feature space while maintaining that information when recovering the original dimensionality. The final encoder representation is the sum of the self-attention encoder and adapter outputs, as a residual connection, as shown in equation 5.2. Figure 5.2 illustrates this process and how the modules interact during training with the frozen text decoder and the new modules for speech. Once both encoder and adapter are trained, any decoder from the multilingual system, each specific for a different language, can be combined with the new speech modules, allowing zero-shot translation to these languages.

$$Adapter(h) = h + ff_2(ReLU(ff_1(norm(h)))) \tag{5.2}$$

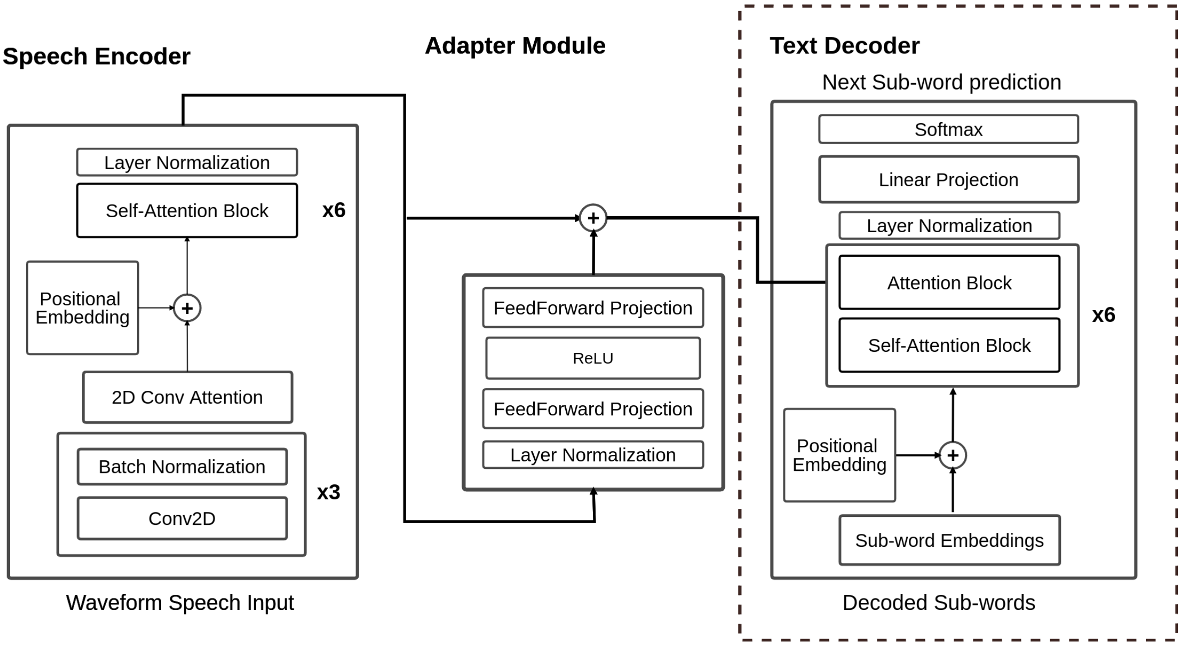

**Figure 5.2:** Close detail to our proposed architecture and its main components, speech encoder, Adapter and text decoder. Dotted square shows the frozen parts of the architecture during training.©2021 IEEE

## 5.3 Experimental Framework

We proposed several experiments to evaluate the performance of our models. We built a baseline system that consists of an end-to-end ASR and SLT architecture based on the S-Transformer [Gangi et al., 2019a]. Using the pre-trained ASR English encoder, we train the SLT systems from English-to-German, French, and Spanish (Baseline); we then alternatively add our proposed architecture presented in Section of the language-specific architecture (LangSpec) or the Adapter module (Baseline & Adapt). Finally, we add the combination of both proposed architectures (+LangSpec & Adapt). Further comparison, we present cascade results (Cascade) from the ASR model trained for each direction and the language-specific Multi-NMT. ASR results are provided in Word Error Rate (WER), while SLT results are provided in BLEU.

### 5.3.1 Data

| Language Pair | Corpus | Language | Set | Segments | Words | Vocab (BPE) |
|---|---|---|---|---|---|---|
| DE-EN | MUST-C | DE | Training | $22.90 \cdot 10^4$ | $54.40 \cdot 10^6$ | |
| | | EN | | | $25.20 \cdot 10^6$ | |
| | MUST-C validation | DE | Validation | $1.423 \cdot 10^3$ | $34.10 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | | EN | | | $35.50 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | MUST-C tst-COMMON | DE | Test | $3.00 \cdot 10^3$ | $60.80 \cdot 10^3$ | |
| | | EN | | | $60.90 \cdot 10^3$ | |
| ES-EN | MUST-C | ES | Training | $26.60 \cdot 10^4$ | $6.50 \cdot 10^6$ | |
| | | EN | | | $6.40 \cdot 10^6$ | |
| | MUST-C validation | ES | Validation | $1.316 \cdot 10^3$ | $33.90 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | | EN | | | $33.80 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | MUST-C tst-COMMON | ES | Test | $2.502 \cdot 10^3$ | $54.80 \cdot 10^3$ | |
| | | EN | | | $58.10 \cdot 10^3$ | |
| ES-EN | MUST-C | FR | Training | $27.50 \cdot 10^4$ | $6.90 \cdot 10^6$ | |
| | | EN | | | $6.40 \cdot 10^6$ | |
| | MUST-C validation | FR | Validation | $1.412 \cdot 10^3$ | $36.60 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | | EN | | | $34.20 \cdot 10^3$ | $31.90 \cdot 10^3$ |
| | MUST-C tst-COMMON | FR | Test | $2.632 \cdot 10^3$ | $58.20 \cdot 10^3$ | |
| | | EN | | | $63.50 \cdot 10^3$ | |

**Table 5.1:** Size of the parallel corpora

Our Multilingual NMT system is Langspec method described in Section 4.2, trained on Europarl dataset for German, English, Spanish, and French. The specific results for this model are available in Section 4.6. As validation and test set, we used *newstest2012/2013*, respectively, from WMT13 [Bojar et al., 2013], which is multi-parallel across all the above languages. All data were preprocessed by applying tokenization, punctuation normalization, and truecasing by using standard Moses scripts [Koehn et al., 2007]. Finally, data was tokenized into subwords by applying BPE [Sennrich et al., 2016b] with 32k operations using subword-nmt library. [1]

We use Must-C as speech dataset [Gangi et al., 2019a], which is a multilingual set extracted from TED talks in English with transcriptions and textual translations in 8 languages. Must-C is the largest corpus in the desired languages (English, German, French, Spanish). The amount of transcribed hours varies from 385 to 504, depending on the language pair. We are using the English transcribed speeches and the bilingual data on pairs English-German, English-French, English-Spanish. We use the training, validation, and test splits that Must-C provides. Multilingual validation and test sets have around 1.4K and 2.5K sentences (respectively) varying on the language pair. The provided speech preprocessing is used for all sets, 40-dimensional log Mel spectrograms computed with windows of 25ms and hop length of 10ms. The same preprocess used for the MultiMT data is applied to the textual, using the same vocabularies and BPE codes.

[1] https://github.com/rsennrich/subword-nmt

### 5.3.2 Model Details and Training Setup

As multilingual NMT architecture, we use the same configuration described on Section 4.5.2, based on a Transformer with 6 attention blocks for both encoder and decoder and 8 attention-heads each and 512 hidden dimensions, with the only modification of adding layer normalization as the last step of both encoder and decoder. Experimental results showed that adding this step to the original textual space helped model convergence during the SLT training. Our baseline SLT architecture follows the parameters in [Gangi et al., 2019a]. The basic parameters are identical to the MultiNMT system, with the only modifications to the original system of using 3 2D-convolutional layers (instead of 2) and 2048 hidden dimensions for the feedforward layers to match the MultiNMT model. The addition of a 3rd 2D-convolutional helped the model learn the mapping to the pre-trained space. On average, over the validation set, this layer reduced the input speech sequence from 4 times longer to just 2 times longer than text, which can help the trained decoder attend the input. All models have beeen trained with 0.1 dropout, 0.0001 learning rate, with inverse square root learning rate schedule, 4000 warmup updates and Adam optimizer with 0.9 and 0.98 betas. Experiments were performed on a single NVIDIA GTX 2080 GPU with batch size of 2000 elements and 64 batches of gradient accumulation.

We tested several alternatives to projection size for the Adapter module to see their effect on the task. Figure 5.3 shows the performance of the model for the tasks of English to Spanish, French, and German SLT compared to each respective bilingual baseline system. All three tasks projecting the encodings to a lower dimensionality space were harmful to the task, even when keeping the original 512 dimensions. When over-parametrizing the space, we observe improved performance in all cases, especially for 4096 dimensions. We tested our models until 9120 dimensions as it was the biggest size we could use without out of memory errors on a single GPU. Experiments show that the models obtain their best performance (close to or slightly better than the baseline system) at 4096 dimensions for German and French and 9120 dimensions for Spanish and English using ASR data. We also measured the impact on the number of parameters of the model by adding the Adapter modules. Baseline models have approximately 77 million parameters for all languages with slight differences due to each language vocabulary. Adapters with 4096 projection size accounted for additional 4 million parameters or 5,5% of the total number of parameters. 9192 projections size accounted for 8 million parameters or 11% of

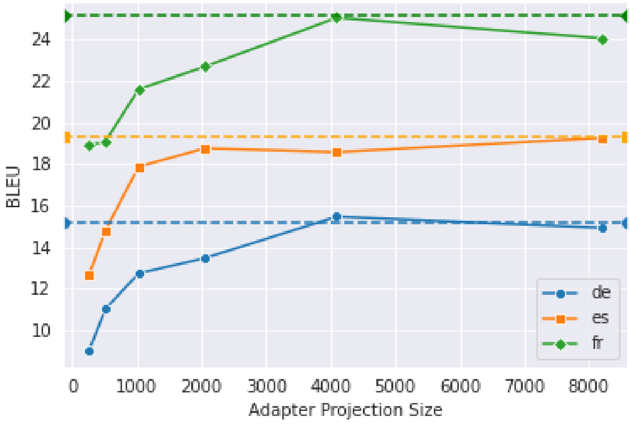

**Figure 5.3:** Model performance for several values of the Adapter projection size for Spanish, German and French. The baseline model in dotted lines.©2021 IEEE

the total number of parameters. These numbers show that Adapter modules are a lightweight and easy-to-tune option for this task.

## 5.4 Results

This section analyzes the performance of our system, both on performance and learned representations. Section 5.4.1 analyzes the SLT results of the results compared with the baseline system and the zero-shot SLT performance of the system. Section 5.4.2 studies the learned encoder representation focusing on the differences between data modalities and the use of Adapter modules.

### 5.4.1 Spoken Language Translation Results

This section aims to test the translation performance of our proposed incremental training for speech. Three different systems are trained on a single supervised translation direction each, from English speech to German (LangSpec$_{ende}$), to French (LangSpec$_{enfr}$), and Spanish (LangSpec$_{enes}$). Each uses the correspondent frozen decoder from our Language-Specific Multilingual NMT architecture (See Section 4.2). Each system is evaluated on supervised ASR and SLT and zero-shot SLT, focusing on the impact of the Adapter module in each scenario.

As our proposed approach can leverage ASR, SLT, and MultiNMT data, we propose two different baseline architectures to compare the impact of these data sources. These two systems are a cascade approach (Cascade) and an end-to-end SLT baseline (Baseline). The cascade approach uses MultiNMT to translate the output of an ASR model, where both models were trained individually using MultiNMT and ASR data, respectively. The end-to-end baseline approach (Baseline) uses the S-Transformer architecture, trained with SLT data.

Our first experiment tests our approach's performance on the two tasks employed during training, ASR pretraining and SLT. Table 5.2 shows that adding a new speech encoder to the LangSpec architecture is possible by training an S-transformer speech encoder with one of the languages in the system (English (en), German (de), French (fr), Spanish (es)). The Langspec architecture without Adapter shows BLEU results that vary from 10.80 in English-to-German up to 19.10 in English-to-French. Compared to the baseline systems, we observe a gap of +5 BLEU points due to the difference between modalities and using a frozen NMT decoder, never trained on the new task.

These results are improved in a large amount when adding the Adapter module (& Adapt), up to almost 6 points BLEU (English-to-French), reducing the gap to the end-to-end baseline system to ±0.2 BLEU for all tested SLT directions. In fact, the Adapter module consistently improves in all directions on SLT models, even when applied to the Baseline, trained from scratch, by +1 BLEU points. This finding shows that the Adapter is helpful to bridge the modalities' representation in all scenarios, not only when limiting the parts of the network trained and both encoder and decoder are jointly trained for the task of SLT.

Our second experiment focuses on the zero-shot capabilities of our Langspec approach. Our proposed incremental training for speech aims to add new speech encoders while maintaining the cross-lingual mapping learned by the Multilingual NMT system, allowing zero-shot translation from speech to other supported languages. We test this behavior by translating to all three languages (German, French, and Spanish rows) using the same models trained only on one of those directions or ASR. Table 5.3 shows the zero-shot with and without an Adapter module ( Adapt). Underlined results indicate the translation direction used for training each system. Otherwise, zero-shot SLT directions are shown. Results show that all systems can perform zero-shot SLT independently of the text decoder used for training from the

| **System** | **ASR$_{en}$** | **SLT$_{ende}$** | **SLT$_{enfr}$** | **SLT$_{enes}$** |
|---|---|---|---|---|
| Cascade | - | 17.30 | 27.15 | 21.29 |
| Baseline | 28.75 | 15.19 | 25.18 | 19.36 |
| Baseline & Adapt | 28.54 | 16.40 | 26.87 | 20.90 |
| LangSpec | 29.60 | 10.80 | 19.10 | 14.23 |
| LangSpec & Adapt | 29.37 | 15.48 | 25.04 | 19.26 |

**Table 5.2:** WER results for ASR$_{en}$ and BLEU results for SLT$_{ende}$, SLT$_{enfr}$, SLT$_{enes}$ models, each trained only its specific task. ©2021 IEEE

LangSpec architecture (en, de, fr, es). We can also observe some correlation between zero-shot results between similar languages, being Spanish and French results better overall for the LangSpec$_{enfr}$ and LangSpec$_{enes}$ systems, being both trained on romance languages.

Adding an Adapter module to our LangSpec systems consistently improves all translation directions on SLT models. Improvements comprehend from 0.08 BLEU for German Zero-shot translation using the Spanish LangSpec$_{enes}$ model up to 2.82 BLEU on French zero-shot translation using the German LangSpec$_{ende}$ model. We also observe higher zero-shot results when training on the French text decoder, which is consistent with the supervised results. This correlation indicates that the selection of the supervised language pair may impact the final performance of the zero-shot translation directions. Higher results on the supervised directions may be related to language similarity and better mappings between the two data modalities.

In addition to models trained with SLT data, we also performed the same zero-shot experiments with a model trained using only ASR data in English (LangSpec$_{en}$). Results show that zero-shot translation can also be achieved for all tested languages in this scenario. The quality of the results correlates with the supervised SLT performance, French being the best direction at 10.85 BLEU points. By contrast, unlike the models trained with SLT data, the addition of adapters seems to harm the zero-shot performance when added to the LangSpec$_{en}$, despite showing a slight improvement of 0.23 of WER on the supervised ASR task. The differences in the tasks could explain this discrepancy (e.g., monotonic vs. non-monotonic alignment) or the less semantic nature of the task of ASR compared to SLT.

| System | de | fr | es |
|---|---:|---:|---:|
| LangSpec$_{en}$ | 6.77 | 10.85 | 6.75 |
| LangSpec$_{en}$ & Adapt | 5.88 | 8.27 | 5.64 |
| LangSpec$_{ende}$ | *10.88* | 10.66 | 8.18 |
| LangSpec$_{ende}$ & Adapt | *15.48* | 13.48 | 10.61 |
| LangSpec$_{enfr}$ | 8.43 | *19.10* | 9.83 |
| LangSpec$_{enfr}$ & Adapt | 9.41 | *25.04* | 11.25 |
| LangSpec$_{enes}$ | 8.05 | 14.06 | *14.23* |
| LangSpec$_{enes}$ & Adapt | 8.13 | 14.46 | *19.26* |

**Table 5.3:** Zero-shot BLEU results from English to different targets (Tgt) (German (de), French (fr), Spanish (es)) using 4 supervised models (LangSpec$_{en}$, LangSpec$_{ende}$, LangSpec$_{enfr}$, LangSpec$_{enes}$). Supervised results are in italics and underlined. ©2021 IEEE

### 5.4.2 Common Representation Results

After studying the performance of our method, one question that arises is how similar the obtained speech representations are to the textual ones from our Multilingual NMT data. Zero-shot performance would benefit from mapping all languages and modalities in the same space. The newly learned representation would be more similar to the ones the system was trained with. This mapping of two data modalities into a shared space becomes more challenging than mapping different languages due to the different natures of the data. Speech utterances may have even an order of magnitude more elements than their textual transcription/translations and are split in arbitrarily given their sampling frequency.

#### Visualization of the intermediate representation.

We further analyze our model by providing a visualization of the intermediate representation and reporting the accuracy in cross-modal sentence retrieval. We use a tool [Escolano et al., 2019b] freely available[2] that allows us to visualize in the same space the intermediate sentences representations from different languages. The tool uses the encoder output fixed-representations as input data and makes a dimensionality reduction of these data using UMAP [McInnes et al., 2018] to visualize the

[2] https://github.com/elorala/interlingua-visualization

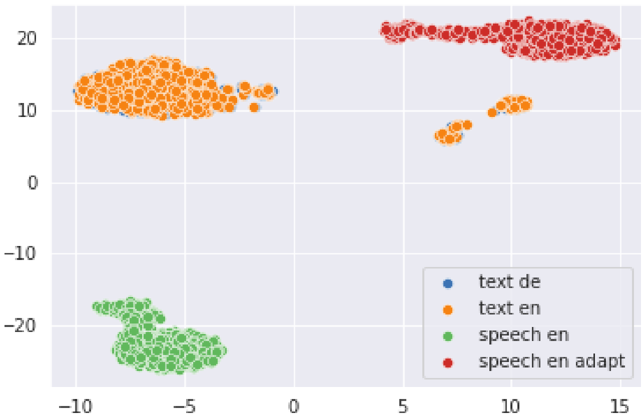

**Figure 5.4:** Visualization. Speech representation without Adapter (green, bottom left), with Adapter (red, top right) and text representation English and German, which are overlapped, (orange,blue respectively, top left) ©2021 IEEE

sentence representations into a 2D plot. We compare the intermediate representation for the speech and text segments for 820 sentences (randomly extracted from the Must-C test set) in Figure 5.4. We observe that the speech representations are far from the text representations, which are altogether in the same part of the space (English/German text overlap). Instead of projecting the speech representation into the region where text representation is found, the Adapter module seems to provide additional information. However, it does not create a mapping between modalities. The distance between models is only reduced by a small amount. The distribution of the tokens in the space is similar in both cases, indicating that the relative distance between sentences from the same set is preserved.

### Cross-modal sentence retrieval

If our hypothesis is correct, the relative position of the sentence in the space should be similar with or without the Adapter, showing a clear correspondence between them. We performed a top-1 sentence retrieval using the same set to compute the cosine distance between representations of speech utterances before and after the Adapter module. Results show that for 73.41% of the sentences, the Adapter's closest representation, from speech without Adapter set, was the same sentence, which proves the previous hypothesis that the Adapter module is not mapping the sentences into a completely new space but modifying the sentences created by the speech encoder. The performance improvement from the Adapter may come from disambiguation or additional information provided by the Adapter over the original representation.

The results from these experiments show two main conclusions about the proposed model. Adapter modules do not learn transformation into a new representation space. The end-to-end training used as baseline fails to capture some useful information for the task. Adding an Adapter between the encoder and decoder can be helpful to mitigate it.

Another question about the results is the impact of the trained Multilingual NMT decoder on the task. The proposed incremental training shows a performance improvement due to transfer learning compared to a randomly initialized decoder when applied to textual data (see Sections 3.4.2 and 4.6). Results on SLT do not show this improvement, and they are similar to the end-to-end baseline without an Adapter. The differences between modalities and obtained representations may explain this behavior. As shown by the analysis on the intermediate representation, adding the Adapter does not show significant improvements.

## 5.5 Conclusions

In this chapter, we present two main contributions. First, extending a MultiNMT system to perform SLT and zero-shot MultiSLT is possible by coupling language-specific encoder-decoders, even from monolingual ASR data only. Our method eliminates dependencies to MultiSLT data, allowing end-to-end systems and cascade systems to be trained on the same data. Experimental results show that our method provides results on pair with an end-to-end baseline architecture, 0.2 difference for all tested languages while providing zero-shot SLT even from models trained only on ASR. Second, the Adapter module is a lightweight and effective method to bridge different modalities in an end-to-end model. On SLT, it can bridge the speech and text representations leading to consistent improvements in all tested translation directions. These improvements are up to +6 BLEU points on the English-to-French SLT direction and up to +1 BLEU points on state-of-the-art end-to-end baseline systems. These improvements reduce the gap between cascade and end-to-end systems and are consistent on all zero-shot translation directions for systems trained in SLT tasks.

Analysis of the cross-lingual mappings shows that this technique improves the model's overall performance while maintaining the structure of the representation space, making it suitable for the end-to-end baseline systems and other similar tasks.

Further work includes jointly training speech and text language-specific encoder-decoders and new language addition methods to improve the knowledge transfer from the MultiNMT model to the SLT tasks. Taking advantage of the trained MultiNMT system in both source and target size may lead to better cross-modal mappings and better overall performance.

# 6 Conclusions

With the appearance of fully shared universal encoder-decoder approaches, language-specific encoder-decoder lost popularity in multilingual machine translation research. This thesis aims to reflect on the applicability of language-specific systems on the current state-of-the-art. This chapter focuses on the insights learned throughout this thesis, investigating the use of these approaches using current deep learning models and evaluating how these methods fulfill our initial research objective from Section 1.2.

Our first objective is **exploring the capabilities of language-specific encoder-decoder architectures**. The main challenge for these architectures is enforcing a cross-lingual mapping between the independent modules. During this work, we propose two main approaches to overcome these difficulties by adding auxiliary tasks to a bilingual model, Chapter 3, and linking modules through translation directions, Chapter 4. We observe that the language-specific architectures can perform zero-shot translation and positive transfer learning as previously shared systems in both cases. On the other hand, we observe that shared models consistently outperform the studied language-specific approaches on zero-shot translation, even when showing better average performance on supervised directions.

A plausible explanation of this performance difference relates to the second objective: investigating the **learning cross-lingual representations without parameter sharing**. A common trait between all proposals is learning language-specific embeddings representations that become more similar as they advance through the encoder layers. This is more evident with the approach from Chapter 4. This effect may be attributed to the differences between auto-encoding and translations tasks combined on Chapter 3.

Our previous objectives focused on jointly trained languages, where all languages are available simultaneously. The next main task of this thesis is the addition of new languages on an already trained system by **training new languages on a previous cross-lingual representation**. Our proposed incremental training approach shows that new languages can be efficiently added by just training a new encoder or decoder at a fraction of the resources required to retrain a fully shared model. Additionally, the performance of the previous translation is preserved, as previous languages are not modified during this process, preventing any negative transfer from new languages.

Our final objective is **training new modalities modules on a previous cross-lingual representation**. Chapter 5 shows that our incremental approach can bridge speech and text modalities with minor adaptations. Our results show that this approach can translate between encoders and decoders that have been trained on different tasks, allowing to translate between languages without existing parallel data.

As final remarks for this work, we would like to include our vision of language-specific encoder-decoder on the future of multilingual NMT research. With the recent improvements in neural architectures and hardware resources, we are witnessing an increased interest in low-resource and understudied languages. In this scenario, we could benefit from platforms and architectures that allowed the training of small independent modules into a known representation, opening new translation directions and tasks to those languages without additional resources.

On the same line of allowing efficient system combinations, with the surge of pre-trained models, we see more and more approaches that focus on learning new tasks on top of a general representation. The study of strategies to combine those systems could also lead to significant improvements to our current approaches.

# Bibliography

[pat, ] Multingual translator patent. `https://www.upc.edu/innovacio/ca/oficina-patents/technology-offers/MKT20210177_I_MartaRuiz.pdf`. Accessed: 2021-11-19.

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

[Aharoni et al., 2019] Aharoni, R., Johnson, M., and Firat, O. (2019). Massively multilingual neural machine translation. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics.

[Andrew et al., 2013] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR.

[Arivazhagan et al., 2019a] Arivazhagan, N., Bapna, A., Firat, O., Aharoni, R., Johnson, M., and Macherey, W. (2019a). The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

[Arivazhagan et al., 2019b] Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., Macherey, W., Chen, Z., and Wu, Y. (2019b). Massively multilingual neural machine translation in the wild: Findings and challenges.

[Armengol-Estapé et al., 2020] Armengol-Estapé, J., Costa-jussà, M. R., and Escolano, C. (2020). Enriching the transformer with linguistic factors for low-resource machine translation.

[Artetxe et al., 2017] Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

[Artetxe et al., 2018] Artetxe, M., Labaka, G., and Agirre, E. (2018). Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

[Artetxe et al., 2019] Artetxe, M., Labaka, G., and Agirre, E. (2019). An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

[Artetxe et al., 2020] Artetxe, M., Ruder, S., and Yogatama, D. (2020). On the cross-lingual transferability of monolingual representations. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. R., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

[Baevski et al., 2020] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[Bahdanau et al., 2015] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[Bapna and Firat, 2019] Bapna, A. and Firat, O. (2019). Simple, scalable adaptation for neural machine translation. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548. Association for Computational Linguistics.

[Barrault et al., 2019] Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. (2019). Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

[Bojar et al., 2013] Bojar, O., Buck, C., Callison-Burch, C., Haddow, B., Koehn, P., Monz, C., Post, M., Saint-Amand, H., Soricut, R., and Specia, L., editors (2013). *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria. Association for Computational Linguistics.

[Bojar and Tamchyna, 2011] Bojar, O. and Tamchyna, A. (2011). Improving translation model by monolingual data. In Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O., editors, *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 330–336. Association for Computational Linguistics.

[Bourlard and Kamp, 1988] Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294.

[Bowman et al., 2015] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

[Britz et al., 2017] Britz, D., Goldie, A., Luong, M.-T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, Copenhagen, Denmark. Association for Computational Linguistics.

[Casas et al., 2019] Casas, N., Fonollosa, J. A. R., Escolano, C., Basta, C., and Costa-jussà, M. R. (2019). The TALP-UPC machine translation systems for WMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162, Florence, Italy. Association for Computational Linguistics.

[Chandar et al., 2016] Chandar, S., Khapra, M. M., Larochelle, H., and Ravindran, B. (2016). Correlational neural networks. *Neural Comput.*, 28(2):257–285.

[Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

[Chu et al., 2017] Chu, C., Dabre, R., and Kurohashi, S. (2017). An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

[Chung et al., 2016] Chung, J., Cho, K., and Bengio, Y. (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.

[Conneau and Lample, 2019] Conneau, A. and Lample, G. (2019). Cross-lingual language model pretraining. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information*

*Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

[Conneau et al., 2018] Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

[Costa-jussà et al., 2017] Costa-jussà, M. R., Escolano, C., and Fonollosa, J. A. R. (2017). Byte-based neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 154–158, Copenhagen, Denmark. Association for Computational Linguistics.

[Costa-jussà and Fonollosa, 2016] Costa-jussà, M. R. and Fonollosa, J. A. R. (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.

[Costa-jussà et al., 2020] Costa-jussà, M. R., Escolano, C., Basta, C., Ferrando, J., Batlle, R., and Kharitonova, K. (2020). Gender bias in multilingual neural machine translation: The architecture matters.

[Currey et al., 2017] Currey, A., Barone, A. V. M., and Heafield, K. (2017). Copied monolingual data improves low-resource neural machine translation. In Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno-Yepes, A., Koehn, P., and Kreutzer, J., editors, *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 148–156. Association for Computational Linguistics.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

[Di Gangi et al., 2019] Di Gangi, M. A., Cattoni, R., Bentivogli, L., Negri, M., and Turchi, M. (2019). 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.

[Dong et al., 2015] Dong, D., Wu, H., He, W., Yu, D., and Wang, H. (2015). Multitask learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

[Duong et al., 2016] Duong, L., Anastasopoulos, A., Chiang, D., Bird, S., and Cohn, T. (2016). An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 949–959, San Diego, California. Association for Computational Linguistics.

[Escolano et al., 2019a] Escolano, C., Costa-jussà, M. R., and Fonollosa, J. A. R. (2019a). From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy. Association for Computational Linguistics.

[Escolano et al., 2020a] Escolano, C., Costa-jussà, M. R., and Fonollosa, J. A. R. (2020a). The TALP-UPC system description for WMT20 news translation task: Multilingual adaptation for low resource MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 134–138, Online. Association for Computational Linguistics.

[Escolano et al., 2021a] Escolano, C., Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (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.

[Escolano et al., 2019b] Escolano, C., Costa-jussà, M. R., Lacroux, E., and Vázquez, P.-P. (2019b). Multilingual, multi-scale and multi-layer visualization of intermediate representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 151–156, Hong Kong, China. Association for Computational Linguistics.

[Escolano et al., 2021b] Escolano, C., Costa-Jussà, M. R., and Fonollosa, J. A. R. (2021b). From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, 72(2):190–203.

[Escolano et al., 2022] Escolano, C., Costa-jussà, M. R., and Fonollosa, J. A. R. (2022). Multilingual machine translation: Deep analysis of language-specific encoder-decoders. *Accepted to the Journal of Artificial Intelligence Research.*

[Escolano et al., 2020b] Escolano, C., Costa-jussà, M. R., Fonollosa, J. A. R., and Artetxe, M. (2020b). Training multilingual machine translation by alternately freezing language-specific encoders-decoders.

[Escolano et al., 2020c] Escolano, C., Costa-jussà, M. R., Fonollosa, J. A. R., and Segura, C. (2020c). Enabling zero-shot multilingual spoken language translation with language-specific encoders and decoders.

[Fan et al., 2020] Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., and Joulin, A. (2020). Beyond english-centric multilingual machine translation.

[Fernando et al., 2020] Fernando, A., Ranathunga, S., and Dias, G. (2020). Data augmentation and terminology integration for domain-specific sinhala-english-tamil statistical machine translation. *CoRR*, abs/2011.02821.

[Firat et al., 2016a] Firat, O., Cho, K., and Bengio, Y. (2016a). Multi-way, multilingual neural machine translation with a shared attention mechanism. In Knight, K., Nenkova, A., and Rambow, O., editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875. The Association for Computational Linguistics.

[Firat et al., 2016b] Firat, O., Sankaran, B., Al-Onaizan, Y., Yarman-Vural, F. T., and Cho, K. (2016b). Zero-resource translation with multi-lingual neural machine translation. In Su, J., Carreras, X., and Duh, K., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 268–277. The Association for Computational Linguistics.

[Freitag and Firat, 2020] Freitag, M. and Firat, O. (2020). Complete multilingual neural machine translation. In Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-jussà, M. R., Federmann, C., Fishel, M., Fraser, A., Graham, Y., Guzman, P., Haddow, B., Huck, M., Jimeno-Yepes, A., Koehn, P., Martins, A., Morishita, M., Monz, C., Nagata, M., Nakazawa, T., and Negri, M., editors, *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 550–560. Association for Computational Linguistics.

[Gaido et al., 2020] Gaido, M., Di Gangi, M. A., Negri, M., and Turchi, M. (2020). 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.

[Gállego et al., 2021] Gállego, G. I., Tsiamas, I., Escolano, C., Fonollosa, J. A. R., and Costa-jussà, M. R. (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.

[Gangi et al., 2019a] Gangi, M. A. D., Negri, M., Cattoni, R., Dessì, R., and Turchi, M. (2019a). Enhancing transformer for end-to-end speech-to-text translation. In Forcada, M. L., Way, A., Haddow, B., and Sennrich, R., editors, *Proceedings of Machine Translation Summit XVII Volume 1: Research Track, MTSummit 2019, Dublin, Ireland, August 19-23, 2019*, pages 21–31. European Association for Machine Translation.

[Gangi et al., 2019b] Gangi, M. A. D., Negri, M., and Turchi, M. (2019b). Adapting transformer to end-to-end spoken language translation. In Kubin, G. and Kacic, Z., editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 1133–1137. ISCA.

[Gangi et al., 2019c] Gangi, M. A. D., Negri, M., and Turchi, M. (2019c). One-to-many multilingual end-to-end speech translation. *CoRR*, abs/1910.03320.

[Gehring et al., 2017] Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

[Geva et al., 2020] Geva, M., Schuster, R., Berant, J., and Levy, O. (2020). Transformer feed-forward layers are key-value memories. *CoRR*, abs/2012.14913.

[Graves, 2012] Graves, A. (2012). Sequence transduction with recurrent neural networks. volume abs/1211.3711.

[Gu et al., 2019] Gu, J., Wang, Y., Cho, K., and Li, V. O. (2019). Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

[Ha et al., 2016] Ha, T., Niehues, J., and Waibel, A. H. (2016). Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

[Haddow and Kirefu, 2020] Haddow, B. and Kirefu, F. (2020). Pmindia - A collection of parallel corpora of languages of india. *CoRR*, abs/2001.09907.

[Hannun et al., 2019] Hannun, A., Lee, A., Xu, Q., and Collobert, R. (2019). Sequence-to-sequence speech recognition with time-depth separable convolutions. In Kubin, G. and Kacic, Z., editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 3785–3789. ISCA.

[He et al., 2016] He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T., and Ma, W. (2016). Dual learning for machine translation. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 820–828.

[Hermansky, 1990] Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4):1738–1752.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

[Hinton, 1984] Hinton, G. E. (1984). Distributed representations.

[Hoang et al., 2018] Hoang, C. D. V., Koehn, P., Haffari, G., and Cohn, T. (2018). Iterative back-translation for neural machine translation. In Birch, A., Finch, A. M., Luong, M., Neubig, G., and Oda, Y., editors, *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 18–24. Association for Computational Linguistics.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[Hotelling, 1936] Hotelling, H. (1936). Relations between two sets of variates. In *Biometrika,28*, page 312–377.

[Houlsby et al., 2019] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

[Inaguma et al., 2019] Inaguma, H., Duh, K., Kawahara, T., and Watanabe, S. (2019). Multilingual end-to-end speech translation. *CoRR*, abs/1910.00254.

[Iranzo-Sánchez et al., 2020] Iranzo-Sánchez, J., Silvestre-Cerdà, J. A., Jorge, J., Roselló, N., Giménez, A., Sanchís, A., Civera, J., and Juan, A. (2020). Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8229–8233. IEEE.

[Jaques et al., 2017] Jaques, N., Taylor, S., Sano, A., and Picard, R. (2017). Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 202–208. IEEE.

[Johnson et al., 2017] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

[Kim et al., 2019] Kim, Y., Petrov, P., Petrushkov, P., Khadivi, S., and Ney, H. (2019). Pivot-based transfer learning for neural machine translation between non-english languages. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 866–876. Association for Computational Linguistics.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International*

*Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.*

[Koehn, 2005] Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers, MTSummit 2005, Phuket, Thailand, September 13-15, 2005*, pages 79–86.

[Koehn et al., 2007] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

[Koehn et al., 2003] Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

[Kudo and Richardson, 2018] Kudo, T. and Richardson, J. (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.

[Kudugunta et al., 2019] Kudugunta, S., Bapna, A., Caswell, I., and Firat, O. (2019). Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

[Kunchukuttan, 2020] Kunchukuttan, A. (2020). The IndicNLP Library. `https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf`.

[Lakew et al., 2018] Lakew, S. M., Erofeeva, A., Negri, M., Federico, M., and Turchi, M. (2018). Transfer learning in multilingual neural machine translation with dynamic vocabulary. *CoRR*, abs/1811.01137.

[Lample et al., 2018] Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics.

[Lavie and Agarwal, 2007] Lavie, A. and Agarwal, A. (2007). Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

[Lecun, 1987] Lecun, Y. (1987). *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models).* Universite P. et M. Curie (Paris 6).

[Lewis, 2010] Lewis, W. (2010). Haitian creole: How to build and ship an MT engine from scratch in 4 days, 17 hours, & 30 minutes. In Yvon, F. and Hansen, V., editors, *Proceedings of the 14th Annual conference of the European Association for Machine Translation, EAMT 2010, Saint Raphaël, France, May 27-28, 2010*. European Association for Machine Translation.

[Li et al., 2003] Li, D., Dimitrova, N., Li, M., and Sethi, I. K. (2003). Multimedia content processing through cross-modal association. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 604–611.

[Li et al., 2021] Li, X., Wang, C., Tang, Y., Tran, C., Tang, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. (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.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Lin et al., 2021] Lin, Z., Wu, L., Wang, M., and Li, L. (2021). Learning language specific sub-network for multilingual machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.

[Liu et al., 2020a] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020a). Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

[Liu et al., 2020b] Liu, Y., Zhu, J., Zhang, J., and Zong, C. (2020b). Bridging the modality gap for speech-to-text translation. *CoRR*, abs/2010.14920.

[Lu et al., 2018] Lu, Y., Keung, P., Ladhak, F., Bhardwaj, V., Zhang, S., and Sun, J. (2018). A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium. Association for Computational Linguistics.

[Luong et al., 2016] Luong, M., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2016). Multi-task sequence to sequence learning. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Grossberger, L. (2018). Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

[Ngiam et al., 2011] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *ICML*.

[Niehues and Cho, 2017] Niehues, J. and Cho, E. (2017). Exploiting linguistic resources for neural machine translation using multi-task learning. *arXiv preprint arXiv:1708.00993*.

[Och and Ney, 2001] Och, F. J. and Ney, H. (2001). Statistical multi-source translation. In *Proceedings of Machine Translation Summit VIII*, Santiago de Compostela, Spain.

[Och and Ney, 2003] Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

[Peters et al., 2018a] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018a). Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[Peters et al., 2018b] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018b). Deep contextualized word representations. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

[Pham et al., 2019] Pham, N.-Q., Niehues, J., Ha, T.-L., and Waibel, A. (2019). Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

[Pires et al., 2019] Pires, T., Schlinger, E., and Garrette, D. (2019). How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

[Ramasamy et al., 2012] Ramasamy, L., Bojar, O., and Žabokrtský, Z. (2012). Morphological processing for English-Tamil statistical machine translation. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pages 113–122, Mumbai, India. The COLING 2012 Organizing Committee.

[Rozis and Skadins, 2017] Rozis, R. and Skadins, R. (2017). Tilde MODEL - multilingual open data for EU languages. In Tiedemann, J. and Tahmasebi, N., editors, *Proceedings of the 21st Nordic Conference on Computational Linguistics, NODALIDA 2017, Gothenburg, Sweden, May 22-24, 2017*, volume 131 of *Linköping Electronic Conference Proceedings*, pages 263–265. Linköping University Electronic Press / Association for Computational Linguistics.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[Salesky et al., 2019] Salesky, E., Sperber, M., and Black, A. W. (2019). Exploring phoneme-level speech representations for end-to-end speech translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Florence, Italy. Association for Computational Linguistics.

[Schultz et al., 2004] Schultz, T., Jou, S.-C., Vogel, S., and Saleem, S. (2004). Using word latice information for a tighter coupling in speech translation systems. In *Eighth International Conference on Spoken Language Processing*.

[Schwenk et al., 2021] Schwenk, H., Chaudhary, V., Sun, S., Gong, H., and Guzmán, F. (2021). Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1351–1361. Association for Computational Linguistics.

[Sennrich et al., 2016a] Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

[Sennrich et al., 2016b] Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

[Siddhant et al., 2020] Siddhant, A., Johnson, M., Tsai, H., Ari, N., Riesa, J., Bapna, A., Firat, O., and Raman, K. (2020). Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8854–8861. AAAI Press.

[Silberer and Lapata, 2014] Silberer, C. and Lapata, M. (2014). Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

[Sperber and Paulik, 2020] Sperber, M. and Paulik, M. (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.

[Stanojević and Sima'an, 2014] Stanojević, M. and Sima'an, K. (2014). Beer: Better evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419.

[Stentiford and Steer, 1990] Stentiford, F. W. M. and Steer, M. G. (1990). *Machine Translation of Speech*, page 183–196. Chapman  Hall, Ltd., GBR.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Tang et al., 2020] Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning.

[Tiedemann, 2009] Tiedemann, J. (2009). News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

[Tiedemann, 2012] Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2214–2218. European Language Resources Association (ELRA).

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[Vauquois, 1968] Vauquois, B. (1968). A survey of formal grammars and algorithms for recognition and transformation in mechanical translation. In Morrel, A. J. H., editor, *Information Processing, Proceedings of IFIP Congress 1968, Edinburgh, UK, 5-10 August 1968, Volume 2 - Hardware, Applications*, pages 1114–1122.

[Vázquez et al., 2019] Vázquez, R., Raganato, A., Tiedemann, J., and Creutz, M. (2019). Multilingual NMT with a language-independent attention bridge. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 33–39, Florence, Italy. Association for Computational Linguistics.

[Vila et al., 2018] Vila, L. C., Escolano, C., Fonollosa, J. A., and Costa-Jussa, M. R. (2018). End-to-end speech translation with the transformer. In *IberSPEECH*, pages 60–63.

[Vincent et al., 2008] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. (2008). Extracting and composing robust features with denoising autoencoders. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1096–1103. ACM.

[Waibel et al., 1991] Waibel, A., Jain, A. N., McNair, A. E., Saito, H., Hauptmann, A. G., and Tebelskis, J. (1991). Janus: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 793–796. IEEE Computer Society.

[Wang et al., 2019] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Confer-*

*ence on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupala, G., and Alishahi, A., editors, *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

[Wang et al., 2020a] Wang, C., Pino, J. M., Wu, A., and Gu, J. (2020a). Covost: A diverse multilingual speech-to-text translation corpus. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4197–4203. European Language Resources Association.

[Wang and Neubig, 2019] Wang, X. and Neubig, G. (2019). Target conditioned sampling: Optimizing data selection for multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5823–5828, Florence, Italy. Association for Computational Linguistics.

[Wang et al., 2020b] Wang, Y., Zhai, C., and Hassan, H. (2020b). Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, Online. Association for Computational Linguistics.

[Weaver et al., 1955] Weaver, W. et al. (1955). Translation. *Machine translation of languages*, 14(15-23):10.

[Weiss et al., 2017] Weiss, R. J., Chorowski, J., Jaitly, N., Wu, Y., and Chen, Z. (2017). Sequence-to-sequence models can directly translate foreign speech. In Lacerda, F., editor, *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2625–2629. ISCA.

[Williams et al., 2018] Williams, A., Nangia, N., and Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

[Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280.

[Wold, 1982] Wold, H. (1982). Soft modeling: the basic design and some extensions. *Systems under indirect observation*, 2:343.

[Woszczyna et al., 1993] Woszczyna, M., Coccaro, N., Eisele, A., Lavie, A., McNair, A., Polzin, T., Rogina, I., Rosé, C., Sloboda, T., Tomita, M., et al. (1993). Recent advances in janus: a speech translation system.

[Zaremoodi and Haffari, 2018] Zaremoodi, P. and Haffari, G. (2018). Neural machine translation for bilingually scarce scenarios: a deep multi-task learning approach. 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 1356–1365, New Orleans, Louisiana. Association for Computational Linguistics.

[Zhang et al., 2021] Zhang, B., Bapna, A., Sennrich, R., and Firat, O. (2021). Share or not? learning to schedule language-specific capacity for multilingual translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

[Zhu et al., 2020] Zhu, C., Yu, H., Cheng, S., and Luo, W. (2020). Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

[Ziemski et al., 2016] Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B.

(2016). The united nations parallel corpus v1.0. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

[Zoph and Knight, 2016] Zoph, B. and Knight, K. (2016). Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.