Domain-Adapted Multilingual Neural Machine Translation

João Alves
j.miguel.alves@tecnico.ulisboa.pt
Instituto Superior Técnico,Universidade de Lisboa, Portugal
January 2021

Abstract

Europe is a continent of linguistic diversity: the European Union has 24 official languages. Globalization has increased the necessity of having information translated to many languages as possible, as fast as possible. Automatic translation, and in particular neural machine translation, can be a good solution to solve this problem. Neural machine translation provides an ideal setting for multilingual systems: it makes it possible to share components across multiple tasks. Multilingual systems make it possible to have a single model that translates from multiple source languages into multiple target languages.

While multilingual systems are appealing, the current reported performance is still behind that of dedicated bilingual models, for most language-pairs. To improve the performance of multilingual systems, we implement an existing approach in the literature, adapters. Adapters are tiny residual layers introduced in the middle of a pre-trained model that are used to adapt the model to a new language, improving its performance. We extend this method by conditioning adapters on one language only (as opposed to the language-pair setting initially proposed). By doing this, we are able to perform direct zero-shot translation and to improve the results in this scenario too.

Finally, we provide a thorough empirical analysis and comparison of different multilingual and pivot-based systems on 24×23 language-pairs. While English is the usual choice of pivot language, we also study the use of different pivot languages, French and German, to translate between Romance and Germanic languages, respectively.

1. Introduction

Globalization has increased the necessity to have information translated in as many languages as possible. Nowadays it is possible to identify three different possibilities to approach the translation problem: human translators only, machine translation only, or a combination of machine translation with human post-editing. Although it is not on pair yet with human translators [1, 2], machine translation has proven to be very effective and useful in many applications while significantly cheaper and more scalable than human translator. Even in applications where quality is important, machine translation can be effective if used with human post-editing.

Recently, a new generation of MT systems have emerged: neural machine translation (NMT), which ally the effectiveness and flexibility of neural networks with the increasing availability of data and computational power. The increasing computational capacity has allowed the emergence of multilingual systems [3]. Neural machine translation makes it appealing to develop multilingual translation systems since the neural architecture is language-agnostic and it is capable of capturing translation properties, such as long-distance re-ordering, even between highly dissimilar languages. It has already been shown that sharing a single translation model between multiple language pairs can achieve competitive results when compared with strong bilingual baseline [4, 5, 6], sometimes even with improvements. However, these improvements are not uniform: when translating to/from low-resource languages results may improve with transfer learning, but on the other hand high-resource languages are often penalized, due to the lack of capacity to accommodate all the language pairs [4].

Although research in multilingual models has shown encouraging results, there are still some challenges: - how to deal with a huge number of languages; - how to deal with different scripting systems (vocabulary); - how to deal with heavy imbalance data across languages and domains; - how to define the practical limit on model capacity; - how to deal with differing degrees of linguistic similarity.

Regarding the aerospace field, machine translation (and in particular multilingual neural machine translation) can have a tremendous impact. Although English is the language used in the aviation sector, it is not the native language of most of the world. Due to this, many maintenance errors have arisen [7, 8]. Maintenance errors can be costly for both aircraft and human life. Simple misunderstandings

can have a devastating impact. The aerospace field is facing a shortage of aviation maintenance technicians. While English is the language that is the most used in this field, the majority of the technicians speak English as a second language. Local languages can have an important role by helping to fill the shortage of technicians quickly. The new advances in the neural machine translation technologies have increased the speed and accuracy of translation and lowered the cost.

Furthermore, the European Space Agency (ESA) has 22 member states with different official languages. Multilingual machine translation can take an important role to spread the work developed across the different countries and to facilitate mutual assistance between member states.

The main contributions of this thesis are:

- We build multilingual NMT models handling the 24 European official languages, translating between any pair of languages among those 24 (while using only one model or a combination of two models). Our systems demonstrate good results, exhibit strong translation accuracy with much fewer parameters, improving the quality of low-resource languages (when compared with bilingual baselines), while keeping competitive results for high-resource languages;
- We inject tiny task-specific layers (adapters) into pre-trained models.[1] We provide in-depth analysis of various aspects of adapters that are crucial to achieve better quality in multilingual NMT. We demonstrate that is possible to close the gap to bilingual baselines with a small number of additional parameters;
- We provide a through empirical analysis and comparison among various strategies, including various choices of pivot languages.

This work is organized as follows: Section 2 covers a description of the state-of-art in multilingual neural machine translation and it presents our contributions; Section 3 covers the experiments performed and the results obtained are discussed; Section 4 sums up the main achievements and leaves suggestions for future work.

2. Multilingual Neural Machine Translation

Multilingual translation models are systems that share a single translation model between multiple language-pairs. According to the language-pairs and translation directions chosen, the multilingual neural machine translation (MNMT) systems can be used in different configurations:

- **many-to-one** - multiple source languages and one target language;
- **one-to-many** - one source language and multiple target languages;
- **many-to-many** - multiple source and target languages.

If we want to translate between N languages, if we follow a naive approach and use individually trained models, it would require $N \times (N-1)$ models. If N is too large, it is impractical to deploy and maintain this huge number of models. Multilingual approaches reduce the number of parameters required: depending on the approach, we can choose the number of parameters to be constant (**Universal Encoder-Decoder Models**) or to grow linearly with the number of languages, $O(N)$ (**Models with Language-Specific Encoders and Decoders**). In this subsection, we present these two different approaches.

2.1. Multilingual Models with Language-Specific Encoders and Decoders

Language-specific approaches require specific encoders or decoders for each language. Some additional features are added to produce shared representations. Although they lead better with the problem of accommodating more language-pairs, they do not take full advantage of the transfer learning feature. Furthermore, the training process tends to be slower and there is an increase in memory requirements due to the increase in the number of parameters.

The first multilingual neural attempt was proposed in [3]. The authors proposed a one-to-many model with a single encoder but separate decoders and attention mechanisms for each target language. The study has shown that it was possible to improve the results of low-resource languages by using a mix of low-resource languages and high-resource languages. This architecture was able to make full use of the source language data (English) across different language-pairs. [9] proposed a similar approach.

Firat *et al*. [10] proposed a many-to-many model (with up to 6 languages) with language-specific encoders and decoders with a single attention mechanism. This was the first work to introduce the idea of direct zero-shot translation. Once again, they showed improvements in low-resource settings. The authors argue that they may use, for each language, encoders and decoder with different architectures or different sizes.

2.2. Universal Encoder-Decoder Models

A universal encoder-decoder model uses only one encoder and one decoder for all language-pairs. It allows integrating any language in the source or target side of the encoder-decoder architecture with only one encoder and one decoder. Moreover it can achieve good results with a much smaller number of parameters (constant in the number of languages). However, the model capacity is a strong limitation to this kind of models. It has been shown that increasing the capacity is directly related with better results,

[1] Code available online at:https://github.com/JoaoMCAlves/Multilingual-Adapters

but scaling capacity leads to a significantly larger computational footprint. Moreover, if we want to add a new language or new data, the whole system needs to be retrained and the quality of translations drops when we add too many languages, especially for those with the most resources [4]. This problem is more evident when languages are not from the same language family.

Johnson *et al*. [6] and Kudugunta *et al*. [11] proposed a many-to-many model that shares all parameters across all language pairs. To do that, the authors used a shared vocabulary for all languages in the dataset. They only introduced a special token at the beginning of every source sentence indicating the target language. The authors expected to obtain good translation results mainly when the target languages are related. The authors were able to perform direct zero-shot translation without any special treatment.

In [12], the authors tested adding different tokens at the beginning of their source sentences: target-specific, source-specific, and pair-specific. The study had the best results when target-specific tokens were used. Moreover, they have tested three different attention mechanisms: target-specific attention, source-specific attention and paired attention which represents a specific language combination. The best results were achieved with a target-specific attention model.

2.3. Transfer Learning and Zero-Shot Translation

Transfer Learning is the mechanism that enables the knowledge from a learned task to improve the performance on a related task, which typically reduces the amount of data needed to achieve the same results. In Natural Language Processing, transfer learning has already been applied to different tasks such as speech recognition, document classification or sentiment analysis.

In multilingual NMT, low-resource languages take advantage of being trained together with high-resource ones. This mechanism is even stronger across similar languages. The most common technique consists of training together low-resource and high-resource languages. An extreme case of transfer learning is zero-shot translation. In this case there is no parallel data between the languages that we are considering.

2.3.1 Pivot-Based Zero-Shot Translation

The most common alternative to multilingual systems is pivot-based zero-shot translation using bilingual direct models. The text is firstly translated from the source language to the pivot language, and then from the pivot language to the target language.

Although this strategy usually achieves good results, it has a few disadvantages. The two-step translation strategy has the potential to propagate errors. As it is a two-step translation system, it requires doubling the latency and computational overhead which is a concern for large-scale NMT models. Moreover there is the possibility of losing important information when the source is translated to the pivot language.

In this work, we are going to combine a multilingual approach with pivot-based zero-shot translation. We are going to use two multilingual models to perform pivot-based zero-shot translation, using the many-to-one model followed by the one-to-many model.

We also explore the use of two different pivot languages: French (fr) and German (de) to translate between Romance languages and German languages, respectively. English is a popular language due to the parallel corpora available. However, there are factors such as language relatedness that can affect the choice of the pivot language for a certain language-pair.

2.3.2 Direct Zero-Shot Translation

Direct zero-shot translation does not require the intermediate step of translating into a pivot language. The multilingual system is trained with multiple source and target languages, and it has the ability of translating between them. Although pivot-based zero-shot translation yield higher BLEU scores than direct translation, recent works [13, 4] suggest that in the near future, direct zero-shot translation is going to be able to perform as good or better than pivot-based zero-shot translation.

Escolano *et al*. [14] and Firat *et al*. [15] performed direct zero-shot translation using a language-specific encoder-decoder architecture but the results were behind the ones achieved with Universal-Encoder approaches. Universal Encoder-Decoder approaches have demonstrated the ability of translating between any language-pair, without using pivot languages and without any special treatment. The shared representation space across languages induces transfer learning. In [6] and [12], the authors obtained reasonable results but they tested their direct zero-shot systems on related languages and large-scale datasets.

2.4. Adapters

Adapters are tiny residual layer that are introduced in the middle of a pre-trained model to improve its performance. This approach shares a large set of parameters across all tasks and introduces a small number of task-specific ones. Adapters have been introduced as an alternative to fine-tune all weights of the pre-trained model.

The main advantage of adapters is that they do not require full fine-tuning of all parameters of the pre-trained model. Once the adapters are introduced, the parameters of the pre-trained model are frozen, and the only parameters that are fine-tuned are the adapters. It allows the model

to converge faster as it is only necessary to train a few numbers of parameters. The main disadvantage of this approach is the necessity of having a component for each task.

2.4.1 Adapters' Architecture

Houlsby *et al*. [16] were the first to propose the use of adapters for NLP tasks. They experimented different architectures and placements and concluded that a two-layer feed-forward neural network with a bottleneck works well. They placed two of these adapters within each transformer layer, one after the multi-head attention and one after the feed-forward layer.

In the field of multilingual NMT, Bapna *et al*. [17] achieved better results using only one adapter (after the feed-forward layer). However, they have introduced a layer normalization and recurrent connections. They introduced a layer normalization in each adapter to avoid retraining all the existing layer normalization layers as was done by [16]. The residual connections are introduced to allow the module to represent a no-operation if necessary. The hidden dimension of the adapter is the hyperparameter that is fine-tuned. They argue that this strategy allows them to adjust the capacity of the adapter easily, depending on the task they want to perform, adjusting only one hyperparameter.

We decided to implement the same architecture as [17]. Our work is different from [17] because we propose the use of adapters for all language-pairs (and not only for high-resource ones) and propose to condition the adapters only on one language instead of a language-pair.

2.4.2 Adapters' Training Process

The injection and training of adapters is a two-step algorithm: firstly, it is necessary to train a fully shared model on all language-pairs, and then there is a fine-tuning using only adapters.

Adapters are trained in the same way as full fine-tuning of the pre-trained model. The data is passed through all layers of the transformer. It is used in the same settings as the fully shared system but the learning rate schedule is reset. All the parameters are frozen, except the adapters.

2.4.3 Proposed Changes: Language-Specific Adapters

In the case of the many-to-many systems, if we want to have adapters for all languages, the naive approach would be injecting language-pair specific adapters (as was suggested in [17]). However, there are a few issues that arise: the first and obvious one is that we would need to have twice the number of adapters compared to the previous experiments ($N-1$ from any language to English and $N-1$ from English to any language).

Moreover, it would not allow us to use adapters to perform direct zero-shot translations. The work developed by [17] did not take into consideration the possibility of performing direct zero-shot translation. As the adapters would be conditioned on English either in the source or in the target, we would have a problem if we wanted to translate between non-English languages.

In an extreme case, we could have an adapter for each possible language combination. However, as we might not have parallel data for every language-pair, we would not be able to train all the adapters and it would imply having $O(N^2)$ adapters. Adapters have a small number of parameters, but if we have such a huge number of adapters, the number of parameters is going to increase quadratically with the number of languages.

To solve this problem we decided that it could be beneficial to have adapters conditioned only on one language, instead of a language-pair. We propose three different architectures based on the language that conditions the choice of the adapters:

- source-specific adapters both in the encoder layers and in the decoder layers;
- target-specific adapters both in the encoder layers and in the decoder layers;
- source-specific adapters in the encoder layers and target-specific adapters in the decoder layers.

In the three different configurations, we are going to have the same number of adapters in each layer. The number of adapters per layer is going to be equal to the number of languages of our dataset, N.

As we said before, [17] did not focus on direct zero-shot translation. The language-pair specific adapters were conditioned on English either on the source or on the target side. To translate between non-English languages, it would not be possible to take advantage of adapters. The language-specific adapters do not have this problem. As they are conditioned only on one language, it is always possible to use them regardless of architecture choice.

3. Experimental Analysis

3.1. Datasets

In our case, the main goal is to cover 24 EU Official Languages: English (en), Bulgarian (bg), Czech (cs), Danish (da), German (de), Greek (el), Spanish (es), Estonian (et), Finnish (fi), French (fr), Hungarian (hu), Italian (it), Latvian (lv), Lithuanian (lt), Dutch (nl), Polish (pl), Portuguese (pt), Romanian (ro), Slovak (sk), Slovenian (sl), Swedish (sv), Irish (ga), Maltese (mt) and Croatian (hr).

For the majority of them, we used the Europarl dataset [18] as the training, development and test corpus. However,

three languages are not covered by Europarl: Irish, Croatian and Maltese, so it was necessary to find alternatives. For Irish we have used *DGT* corpus [19] and for Croatian and Maltese the *TildeMODEL* corpus [20]. As the domain is not exactly the same, the transfer learning process is harder.

In order to be able to compare the results achieved for direct zero-shot Translation and Pivot when using languages of Europarl we have created a test set and a development set common to all languages that Europarl supports. Namely, we used the data from the first semester of 2009 to accomplish this. The development set has 4000 sentences and the test set has 4630 sentences.

3.2. Data Preprocessing and Vocabulary Construction

We tokenized and truecased all the sentences using scripts from the Moses toolkit [21]. We used Byte Pair Encoding (BPE) [22] to segment sentences into subword symbols and to construct all our vocabularies. For the bilingual experiments, the size of the shared vocabulary was 10k tokens and for multilingual experiments, we used a shared vocabulary containing 32k tokens. We decided to use this size based on the work developed by Arivazhagan *et al*. [4].

3.3. Metrics

As we are dealing with a large number of languages, it is important to find clear ways to present the results. Rather than providing the BLEU score for each language pair, we are going to compute some average BLEU scores:

- $BLEU_{20}$ - average BLEU over all 20 language pairs covered by *Europarl*
- $BLEU_{23}$ - average BLEU over all 23 language pairs
- $BLEU_{HR}$ - average BLEU over the high-resource language pairs
- $BLEU_{LR}$ - average BLEU over the low and medium resource language pairs

3.4. Bilingual Baselines

We trained bilingual baselines using the training dataset that we have described before. We performed all our experiments using the open-source *Fairseq* toolkit [23].

The experiments were performed with transformer base settings [24], containing around 75M parameters. For these experiments, we used transformers with 6 layers in both the encoder and the decoder, model dimension set to 512 and 8 attention heads. For optimization, we use Stochastic Gradient Descent with Adam Optimizer [25] ($\beta_1 = 0.9$, $\beta_2 = 0.98$) with label smoothing of 0.1 and scheduled learning rate (warm-up step 16k). We use dropout of 0.3. BLEU scores are calculated on the checkpoint with the best validation BLEU score employing beam search with a beam size of 5.

We plot the two main directions separately in different plots, when translating to or from English, in Figure 1.

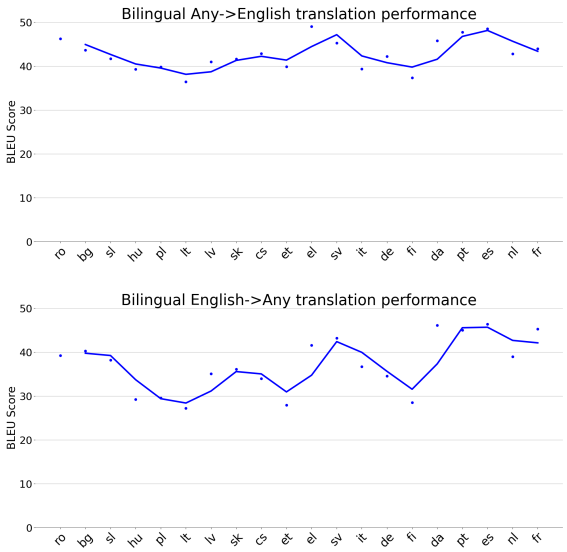

Figure 1: Quality of individual bilingual models on all 20 language pairs covered by Europarl, measured in terms of BLEU score. Languages are arranged in increasing order of available data from left to right. Performance on individual language-pairs is reported using dots and a trailing average is used to show the trend.

3.5. Fully Shared Models

We trained three fully shared multilingual models: 1) many-to-one model from 23 languages to English, 2) one-to-many model from English into 23 languages and 3) many-to-many model trained using all 46 translation directions (to and from English).

For all settings, we train a single transformer base with the same hyper-parameters settings as the bilingual models. The only difference is the use of a shared BPE vocabulary with 32k tokens. Moreover, we followed the approach of [6] and added a target-language token at the end of each source sentence. In all cases, we report test results (in terms of BLEU score) for the checkpoint that performed best on the validation set.

As we have referred before, Europarl does not cover three official European languages: Croatian, Irish and Maltese. The deterioration in these languages is much higher, so we decided not to represent them in the same graphs as the other languages.

3.5.1 Results

We plot the results when translating to and from English in Figure 2. In the first plot, we represent the performance of

the many-to-many model and the many-to-one when translating from all languages to English. In the second one, we represent the performance of the many-to-many and the one-to-many models when translating from English to other languages. We present some average metrics in Table 1.

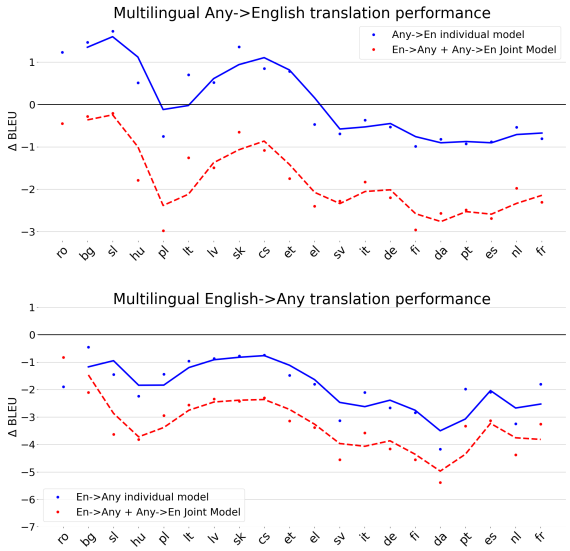

Figure 2: Quality of many-to-many, many-to-one and one-to-many models. Languages are arranged in increasing order of available data from left to right. Results are reported relative to those of bilingual baselines. The colors corresponds to the following strategies : (i) Blue: Models trained in one translation direction (many-to-one and one-to-many respectively) (ii) Red: Model trained in both translation directions (to and from English). Performance on individual language-pairs is reported using dots and a trailing average is used to show the trend.

Any→En	$BLEU_{23}$	$BLEU_{20}$	$BLEU_{LR}$	$BLEU_{HR}$
1.Bilingual	46.32	42.72	41.24	43.66
2.All → En	45.07	42.79	43.66	42.93
4.All → All	42.92	40.93	40.04	41.29
En→Any	$BLEU_{23}$	$BLEU_{20}$	$BLEU_{LR}$	$BLEU_{HR}$
1.Bilingual	36.90	37.14	33.67	40.51
3.En → All	34.25	35.03	32.02	37.84
4.All → All	33.04	33.85	31.05	36.47

Table 1: Average translation quality (BLEU score) of multilingual models trained. All → All reports the performance of the multilingual model trained on all translation directions, En → All reports the performance of the model trained on all language pairs with English as the source and All → En reports the performance on the model trained on all language pairs with English as the target.

The many-to-one shared model achieved worse results than the bilingual baselines if we consider all the languages of our dataset (-1,25 $BLEU_{23}$, 2-1, Table 1), but they were able to outperform the bilingual baselines if we consider only the languages that are covered by Europarl (+0,07 $BLEU_{20}$, 2-1, Table 1). It outperforms bilingual baselines when translation to English for low settings (+0,84 $BLEU_{LR}$ 2-1, Table 1). This phenomenon was already described in previous works like [5] and [26]. A multilingual system with shared weights promotes transfer learning from high-resource languages to low-resource ones. On the other hand, the high-resource language pairs have a decrease in their performance (-0,73 $BLEU_{HR}$ 2-1, Table 1). That might be explained by two different reasons:

- the different language pairs are competing for capacity, and due to the limited model size, the high-resource ones have difficulties to accommodate all the space that they need;
- the model converges before it trains on significant portions of the high-resource data.

Regarding the one-to-many setup, when we compare the results achieved with bilingual baselines, it is easy to realize that none of the language pairs had an improvement in their performance. The deterioration in the performance is higher for high-resource languages (-2,67 $BLEU_{HR}$ 3-1, Table 1), while the performance on low-resource languages does not deteriorate that much (-1,64 $BLEU_{LR}$ 3-1, Table 1).

Analysing the results of Figure 2, we notice both the many-to-one model and the one-to-many model achieve much better results than the many-to-many model. The deterioration is higher when translating to English (-2,15 $BLEU_{23}$ 4-2, Table 1) than when translating from English (-1,21 $BLEU_{23}$ 4-3, Table 1). The many-to-many model must accommodate twice as many translation directions with the same number of parameters. In this case, we have 46 translation directions instead of 23. Due to this, the many-to-many model suffers more relevant capacity issues.

3.6. Adapters

3.6.1 Language-Pair Specific Adapters

We have fine-tuned the many-to-one and the one-to-many with language-pair specific adapters on top. We introduced a variable number of adapters, depending on the training data available for each language-pair, and evaluated the influence on the results.

We tested two different settings: injecting adapters only for high-resource languages and injecting adapters for all languages. We tested it both on the many-to-one and the one-to-many models. We plot results in Figure 3.

Adapters clearly help to improve the performance of the many-to-one model. There is an interesting phenomenon:

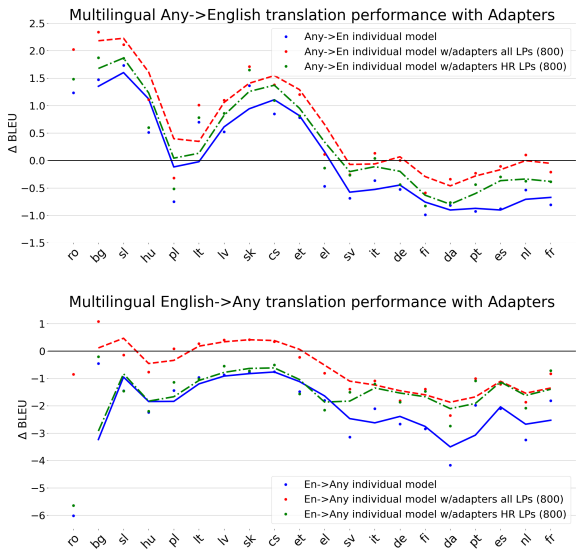

Figure 3: Effects of injecting adapters. Languages are arranged in increasing order of available data from left to right. Results are reported relative to those of bilingual baselines. The colors corresponds to the following strategies : (i) Blue: Fully Shared Model (ii) Green: Fully Shared with adapters for High Resource languages (iii) Red: Fully Shared with adapters for all languages. Performance on individual language-pairs is reported using dots and a trailing average is used to show the trend.

even when we introduce adapters for high-resource languages only, the results for low-resource languages improve. This may be related to the fact that the embedding layer is not frozen. However, the best results are achieved when adapters were injected for all languages. This configuration achieved the best results for every single language.

As we have seen before and unlike the many-to-one model, there is a degradation in the results obtained with the one-to-many model, for all languages pairs when a fully shared model is used. Analysing the graph, it is possible to conclude that when we inject adapters only for high-resource settings, there is a huge improvement for high-resource languages. The introduction of language-pair specific adapters for all language pairs was capable of achieving better results in general. For some low-resource languages, it was even able to surpass bilingual baselines.

3.6.2 Language-Specific Adapters

As described in Section 2.4, injecting language-pair specific adapters on top of the many-to-many model would not be a good choice. It would require a considerable number of adapters and it would not allow us to take advantage of them to perform direct zero-shot translation. To solve this issue, we experimented three different adapter configurations:

1. Source-specific adapters both in the encoder and decoder layers;
2. Target-specific adapters both in the encoder and decoder layers;
3. Source-specific adapter in the encoder layers and Target-specific adapters in the decoder layers.

We plot the results in Figure 4. Further results are summarized in Table 2. Complete results are in appendix A of the thesis.

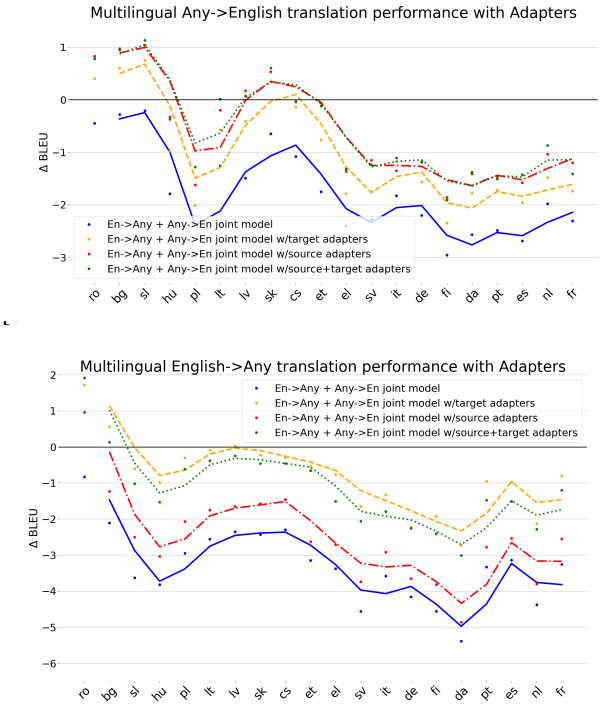

Figure 4: Effects of conditioning the adapters on different conditions. Languages are arranged in increasing order of available data from left to right. Results are reported relative to those of bilingual baselines. The colors corresponds to the following strategies : (i) Blue: Fully Shared Model (ii) Orange: Fully Shared Model with adapters conditioned on the target language (iii) Red: Fully Shared Model with adapters conditioned on the source language (iii) Green: Fully Shared Model with adapters conditioned on the source language in the encoder and conditioned on the target language in the decoder. Performance on individual language-pairs is reported using dots and a trailing average is used to show the trend.

The injection of **target-specific adapters** show the largest benefit for **English**→**X** translation (+2,66 $BLEU_{23}$,

Any→En	$BLEU_{23}$	$BLEU_{20}$	$BLEU_{LR}$	$BLEU_{HR}$
1.Many-to-Many	42.92	40.93	40.04	41.29
2.Many-to-Many w/Source Adapters	44.70	42.10	41.36	42.30
3.Many-to-Many w/Target Adapters	43.75	41.71	40.96	41.94
4.Many-to-Many w/Source+Target Adapters	44.42	42.14	41.41	42.32
En→Any	$BLEU_{23}$	$BLEU_{20}$	$BLEU_{LR}$	$BLEU_{HR}$
1.Many-to-Many	33.04	33.85	31.05	36.47
2.Many-to-Many w/Source Adapters	33.82	34.63	31.97	37.11
3.Many-to-Many w/Target Adapters	35.70	36.33	33.59	38.88
4.Many-to-Many w/Source+Target Adapters	35.28	36.00	33.33	38.51

Table 2: Average translation quality (BLEU score) of multilingual models trained. Many-to-Many reports the performance of the fully shared model trained on all translation directions, Many-to-Many w/Source Adapters reports the performance of the Fully Shared Model with adapters conditioned on the source language, Many-to-Many w/Target Adapters reports the performance of the Fully Shared Model with adapters conditioned on the target language, Many-to-Many w/Source+Target Adapters reports the performance of the Fully Shared Model with adapters conditioned on the source language in the encoder and conditioned on the target language in the decoder.

3-1, Table 2). Furthermore, the results were consistent, if we consider only the low-resource settings, the improvement was of +2,54 $BLEU_{LR}$ and if we consider the high-resource ones it was +2,41 $BLEU_{HR}$. For **X→English** translation, it shows a smaller benefit (+0,83 $BLEU_{23}$, 3-1, Table 2). As the adapters are conditioned on the target language, the capacity is mainly increased for English→X and consequently, the results are better in this translation direction.

Source-specific adapters yield a larger benefit for **X→English** tasks (+1,78 $BLEU_{23}$, 2-1, Table 2) and a smaller benefit for **English→X** tasks (+0,78 $BLEU_{23}$, 2-1, Table 2). For both translation directions, the results have improved for low-resource and high-resource configurations.

Although target and source-specific adapters were able to achieve good results (target adapters are better when translating from English to any language, and source adapters are better when translating from any language to English), a **combination of source-specific adapters in the encoder and target-specific adapters** in the decoder was able to achieve more balanced results, if we consider both translation directions. If we consider only English→X translation, it performs slightly worst than target adapters (-0,42 $BLEU_{23}$, 4-3, Table 2). But if we consider X→English translation, it was able to achieve better results than the exclusive use of source adapters (+0,67 $BLEU_{23}$, 4-2, Table 2).

To sum up, **the use of target adapters both in the encoder and in the decoder was the approach that achieved the best results when translating from English to any language, both for low-resource a high-resource settings. The combination of source adapters (in the encoder side) and target adapters (in the decoder side) was the setting that achieved the best results when translating from any language to English**.

3.7. Translation between Non-English Languages

3.7.1 Results on Pivot-Based zero-shot Translation

Regarding Pivot translation, we did two different experiments: **pivoting using the bilingual models** and **pivoting using a many-to-one model followed by a one-to-many model**. In the case of using the bilingual models, we need to have 46 models if we want to cover all the EU official languages (23 models from different languages to English and 23 models trained in the opposite direction). If we use the many-to-one and the one-to-many models, we only need to have 2 models. We use the one-to-many and the many-to-one models with language-pair specific adapters on top because this was the approach that achieved better results in the previous experiments. In Table 3 we compute the average values when each language is either the source (left side) or the target (right side). It is possible to find complete results in appendix A of the thesis.

The results achieved by the bilingual models were better than the ones obtained by the shared models. However, shared models achieved competitive results, being better for some languages combination.

If we analyse only the left side of the table (source side): for high-resource languages, the results are always better when using bilingual models for pivoting and for low-resource languages the difference is smaller (0.09 BLEU points)

If we take a look at the right-most column of Table 3: the conclusions are very similar, for high-resource settings bilingual models achieve better results. For low-resource settings, the difference is only 0.06 BLEU points.

Source	Pivot Bilingual	Pivot Shared	Target	Pivot Bilingual	Pivot Shared
Average	27.71	27.38	**Average**	27.93	27.60
$BLEU_{LR}$	27.72	27.63	$BLEU_{LR}$	25.86	25.80
$BLEU_{HR}$	27.69	27.06	$BLEU_{HR}$	30.46	29.80

Table 3: Pivot-Based zero-shot translation results.

Taking into consideration the number of models necessary to perform each of the strategies, we believe that using the shared models for pivoting is the best approach. Instead of using 46 models, we only need 2 and the results achieved are quite comparable to the ones obtained with bilingual models.

3.7.2 Results on Direct Zero-Shot Translation

Many-to-many models have the advantage of being able of translating between any pair of supported languages, even when parallel data is not available.

As we have described before, we have trained three different settings using adapters on top of the many-to-many model. We have already concluded that they yield benefits when translating to and from English. But how do they impact the performance of direct zero-shot translation? We performed direct zero-shot translation using the many-to-many fully shared model and the models with adapters on top. The results are in Table 4. Once again, we compute the average values when each language is either the source (left side) or the target (right side). Complete results are in appendix A of the thesis.

If we use the fully shared model to generate direct zero-shot translations, the results that we obtain are poor, especially if we compare with the ones that we obtained using pivot-based techniques (Table 3).

Regarding the use of source-specific adapters, the results that we obtained are also poor. We can say that the translations generated are completely useless. As the results were so poor (around 3-4 BLEU points), we decided not to present them in Table 4.

The combination of source adapters (in the encoder) and target adapters (in the decoder) was able to achieve good results too but not as good as the ones obtained with only target-specific adapters.

Using only target adapters improve a lot the results achieved in direct zero-shot translation. Our results show that there an improvement of +9,71 BLEU points in average when compared with the fully shared model. The improvement is higher for low-resource language pairs. In general, we can say that the use of only target-specific adapters yields a larger benefit.

3.8. Use of Different Pivot Languages

Next we ask ourselves if we can improve the performance if we use a different pivot language. English is the usual choice, but we want to check if the usage of a language that belongs to the same language family helps to improve the translation performance in the case of pivot translation.

We have identified two languages with potential to be used as pivot languages: **German** (de) and **French** (fr). These languages were chosen because they present a big volume of data and are representative of different language families. After having chosen these two languages, we have selected the languages that could be helped with this strategy:

- pivot language **French**: **Portuguese** (pt), **Spanish** (es), **Italian** (it) and **German** (de);
- pivot language **German**: **Polish** (pl), **Dutch** (nl), **Swedish** (sv), **French** (fr) and **Danish** (da).

We trained dedicated bilingual model for each language to and from English, and to and from the desired pivot language (French or German depending on the previous list). To train these models, we used *Europarl* dataset. After having trained the bilingual dedicated models, we used them for pivoting. We compare the translation quality of models of pivot translation in Tables 5 (for French) and 6 (for German).

Somewhat surprisingly, we see that English is the best pivot language. The reasons behind it may be:

- the bilingual models to/from French have a worse performance (in terms of BLEU score) than the bilingual models to/from English. As the models that are used for pivoting are worse, consequently the pivot results are worse too. In the case of German, degradation is even higher;
- French and especially German are morphologically rich languages. The degradation in the case of using German as pivot language (-3,35 BLEU points on average when compared with results obtained using English) is much higher than in the case of French (-1,48 BLEU points).

Source	Many-to-Many Fully Shared	Target Adapters	Source+Target Adapters	Target	Many-to-Many Fully Shared	Target Adapters	Source+Target Adapters
Average	13.40	23.11	25.24	Average	13.55	23.31	22.91
$BLEU_{LR}$	9.03	22.34	21.90	$BLEU_{LR}$	15.01	22.47	21.93
$BLEU_{HR}$	18.75	24.05	23.72	$BLEU_{HR}$	11.76	24.33	24.12

Table 4: Direct zero-shot translation results.

	de		it		pt		es	
	English	French	English	French	English	French	English	French
de	-	-	28.75	27.40	32.73	30.34	33.63	31.45
it	24.47	22.92	-	-	32.80	32.01	33.43	32.46
pt	27.75	25.93	32.31	31.61	-	-	38.12	36.32
es	27.53	25.87	31.87	31.09	36.98	35.26	-	-

Table 5: Average translation quality (BLEU) of pivot translations using different pivot languages: English and French.Rows indicate source language, columns indicate target language.

	da		nl		fr		pl		sv	
	English	German	English	German	English	German	English	German	English	German
da	-	-	32.11	29.93	36.14	33.08	23.70	21.48	34.77	30.90
nl	34.50	31.08	-	-	35.64	33.32	23.45	21.04	32.22	28.48
fr	33.99	29.95	31.10	29.12	-	-	23.25	20.92	31.78	27.73
pl	30.89	28.24	29.00	27.22	33.57	30.74	-	-	29.30	26.67
sv	36.90	32.45	31.31	29.12	35.93	32.57	23.77	21.47	-	-

Table 6: Average translation quality (BLEU) of pivot translations using different pivot languages: English and German.Rows indicate source language, columns indicate target language.

4. Conclusion

We show that it is possible to train multilingual models in large scale settings and that they can improve performance over bilingual baselines, especially for low resource language pairs.

In order to improve the performance of fully-shared multilingual NMT systems, we followed [17] and introduced adapters on top of the three fully shared models. Adapters enable a multilingual model to adapt to multiple target tasks without forgetting the original parameters of the model. In the case of the many-to-one and one-to-many models, we used language-pair specific adapters. In the case of the many-to-many model, we explored different kind of adapters. We tried to use adapters compatible with the idea of direct zero-shot translation. The exclusive use of source adapters was the approach that achieved worst results in all scenarios. When translating from English to other languages, the exclusive use of target adapters was the best choice. When translating from other languages to English, the hybrid approach was the one that had the best performance. In terms of direct zero-shot translation (when translating between non-English languages), target adapters, both in the encoder and decoder, were the ones that achieved the best results.

Regarding pivot-based zero-shot translation, we explored the use of different pivot languages. We explored the use of French and German to translate between Romance and Germanic languages, respectively. We were expecting to have better results when using French or German for pivoting between languages from the same family. However, that did not occur. The best results were obtained when using English as the pivot language.

Regarding future work, Pfeiffer *et al*. [27] proposed to combine different adapters (*AdapterFusion*) for different Natural Language Processing tasks such as sentiment analysis, commonsense reasoning, paraphrase detection, and recognizing entailment. They use a mechanism very similar to the transformer's attention. One interesting direction direction to be explored is the use of this technique for Multilingual Neural Machine Translation. Sharing knowledge from multiple adapters could possible improve, even more, the results obtained due to Transfer Learning.

References

[1] Antonio Toral. Post-editese: an Exacerbated Translationese. *arXiv*, 2019. 1

[2] Samuel Läubli, Chantal Amrhein, Patrick Düggelin, Beatriz Gonzalez, Alena Zwahlen, and Martin Volk. Post-editing productivity with neural machine translation: An empirical

assessment of speed and quality in the banking and finance domain. *arXiv*, 2019. 1

[3] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-Task learning for multiple language translation. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 1:1723–1732, 2015. 1, 2

[4] Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. 2019. 1, 3, 5

[5] Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively Multilingual Neural Machine Translation. pages 3874–3884, 2019. 1, 6

[6] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. 1, 3, 5

[7] Colin G. Drury and Jiao Ma. Do language barriers result in aviation maintenance errors? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 47(1):46–50, 2003. 1

[8] C. Drury and C. V. Marin. Language error in aviation maintenance. 2005. 1

[9] Surafel M. Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. Transfer Learning in Multilingual Neural Machine Translation with Dynamic Vocabulary. 2018. 2

[10] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 866–875, 2016. 2

[11] Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. Investigating Multilingual NMT Representations at Scale. pages 1565–1575, 2019. 3

[12] Graeme Blackwood, Miguel Ballesteros, and Todd Ward. Multilingual Neural Machine Translation with Task-Specific Attention. pages 3112–3122, 2018. 3

[13] Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. 2020. 3

[14] Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders. 2020. 3

[15] Orhan Firat. Zero-Resource Neural Machine Translation with. pages 268–277, 2016. 3

[16] Neil Houlsby, Andrei Giurgiu, Stanisraw Jastrzçbski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4944–4953, 2019. 4

[17] Ankur Bapna and Orhan Firat. Simple, Scalable Adaptation for Neural Machine Translation. pages 1921–1931, 2019. 4, 10

[18] Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. 2019. 4

[19] Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. DGT-TM: A freely available translation memory in 22 languages. *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, pages 454–459, 2012. 5

[20] Roberts Rozis and Raivis Skadinš. Tilde MODEL - Multilingual Open Data for EU Languages. *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa*, (131):263–265, 2017. 5

[21] Philipp Koehn, Hiue Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, and Brooke Cowan. Moses: Open source toolkit for statistical machine translation. *Prague: Proceedings of 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions.*, (June):177–180, 2007. 5

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016. 5

[23] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. pages 48–53, 2019. 5

[24] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in neural information processing systems*, (Nips):5998–6008, 2017. 5

[25] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015. 5

[26] Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 875–880, 2020. 6

[27] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. 2020. 10