Journal of Artificial Intelligence Research 73 (2022) 1535-1552 Submitted 01/2021; published 04/2022

Multilingual Machine Translation: Deep Analysis of Language-Specific Encoder-Decoders

Carlos Escolano CARLOS.ESCOLANO@UPC.EDU
Marta R. Costa-jussà MARTA.RUIZ@UPC.EDU
José A. R. Fonollosa JOSE.FONOLLOSA@UPC.EDU
TALP Research Center, Universitat Politècnica de Catalunya, Barcelona

Abstract

State-of-the-art multilingual machine translation relies on a shared encoder-decoder. In this paper, we propose an alternative approach based on language-specific encoder-decoders, which can be easily extended to new languages by learning their corresponding modules. To establish a common interlingua representation, we simultaneously train N initial languages. Our experiments show that the proposed approach improves over the shared encoder-decoder for the initial languages and when adding new languages, without the need to retrain the remaining modules. All in all, our work closes the gap between shared and language-specific encoder-decoders, advancing toward modular multilingual machine translation systems that can be flexibly extended in lifelong learning settings.

1. Introduction

Multilingual machine translation is the ability to generate translations automatically across a (large) number of languages. Research in this area has attracted much attention in recent years, from both the scientific and the industrial community. With the recent shift ot a neural machine translation paradigm (Bahdanau, Cho, & Bengio, 2015), the opportunities for improvements in this area have dramatically expanded. Thanks to the encoder-decoder architecture, there are viable alternatives to expensive pairwise translations based on classic paradigms [1].

The main proposal in this direction is the shared encoder-decoder (Johnson et al., 2017) with massive multilingual enhancements (Arivazhagan et al., 2019b). While this approach enables zero-shot translation and is beneficial for low-resource languages, it has multiple drawbacks: (i) the entire system has to be retrained when adding new languages or data or alternatively, use an adapter module to add a new language (Bapna, Arivazhagan, & Firat, 2019); (ii) the quality of translation drops when adding too many languages or for those with the most resources (Arivazhagan et al., 2019b); (iii) the shared vocabulary grows when adding a large number of languages (especially when they do not share alphabets); and (iv) the shared encoder is not able to add multiple modalities such as image or speech.

In this paper, we propose a new framework that can be incrementally extended to new languages without the aforementioned limitations (§3). Our proposal is based on language-specific encoders and decoders that rely on a common intermediate representation space. For that purpose, we simultaneously train the initial *N* languages in all translation directions. New languages are naturally added to the system by training a new module coupled with any of the existing languages, while new data can be easily added by training only the module for the corresponding language.

1. http://www.euromatrixplus.net

Escolano, C.; Costa-jussà, M.R.; Fonollosa, J.A.R. Multilingual machine translation: Deep analysis of language-specific encoder-decoders. "Journal of artificial intelligence research", 25 Abril 2022, vol. 73, p. 1535-1552. https://doi.org/10.1613/jair.1.12699

We evaluate our proposal on three experimental configurations: translation for the jointly trained initial languages, translation when incrementally training a new language, and zero-shot translation (§4). Our results show that the proposed method is competitive in the first two configurations, but still lags behind the shared encoder-decoder in zero-shot translation. In order to further understand our model and as an extension of the previous publication by Escolano, Costa-jussà and Fonollosa (2021), we provide a deeper analysis in the following directions. In §4.3, we study the effect of fine-tuning and our approach shows robustness by avoiding catastrophic forgetting. In §4.4, we analyze why our model does not suffer from the attention mismatch, mentioned in previous works, even though the modules do not share parameters (Firat, Cho, & Bengio, 2016a). To perform this analysis we explore the effect of excluding training data from certain language pairs. We observe that when there are four languages in the initial system, when we train with only parallel data from and to one language, our system is not able to learn all the translation directions. However, when adding one more language (parallel data from and to two languages), our system achieves almost full performance compared to training with all the translation directions from the initial languages in the system. Then, to better understand the nature of the learned representations, we run additional experiments on natural language inference, where the language-specific encoder-decoders internal representation is evaluated in all the encoder layers (§5.1). Finally, we visualize these representations in a two dimensional space (§5.2).

Overall, we provide a deep analysis of this new multilingual model based on language-specific encoder-decoders that can incrementally be extended to new languages and that can improve the performance of multilingual machine translation without parameter sharing, closing the existent gap with shared encoder-decoders architecture.

The rest of the paper is organized as follows. Section 2 reviews most related work. Section 3 details the proposed method for multilingual machine translation. Section 4 overviews experiments in machine translation where we compare our proposed method to the shared encoder/decoder which is considered the state-of-the-art in current multilingual approaches. Section 5 provides an in-depth analysis of the intermediate representations created with our proposed method by showing the results in natural language inference and visualizing some intermediate sentence representations. Finally, Section 6 concludes and suggests new research paths for future studies.

2. Related Work

Multilingual neural machine translation can refer to translating from one-to-many languages (Dong et al., 2015), from many-to-one (Zoph & Knight, 2016) and many-to-many (Johnson et al., 2017). In this section, we briefly review the latter, which is the most general approach and the one that we follow.

Within the many-to-many paradigm, the existing approaches can be further subdivided into shared or language-specific encoder-decoders. The latter approaches vary from the sharing of parameters to no sharing at all. Among the common advantages of these approaches is that they allow for zero-shot translations; contrastive properties between total sharing and no-sharing (language-specific) are presented in what follows and are summarized in Table 1.

2.1 Shared Encoder-Decoder

Ha, Niehues, and Waibel (2016) and Johnson et al. (2017) feed a single encoder and decoder with multiple input and output languages. Given a set of languages, a shared architecture has a shared

	Shared	Language-specific
Efficiency	**One single encoder-decoder**	N encoders and N decoders
Continuous learning	Need to retrain or add adapters	**No need to retrain when adding languages**
Transfer Learning (when adding languages)	To initial and added	**Only to added**
Vocabulary	Grow with N	**Independent for each language**

Table 1: A comparison of the approaches shared (Arivazhagan et al (2019)) and the language-specific encoders-decoders. N refers to the number of languages in the system. Advantages are highlighted in bold.

encoder E_u and a shared decoder D_u that are trained on all the initial language pairs at once. The model shares parameters, vocabulary and tokenization among the languages to ensure that no additional ambiguity is introduced in the representation. By sharing a single model across all languages, the system is able to represent all languages in a single space both semantically and lexically. This allows the translation between language pairs never seen during the training process just by sharing a common sentence representation, which is known as a zero-shot translation.This architecture provides a simple framework to develop multilingual systems because it does not require modifications of a standard neural machine translation model, and information is easily shared among the different languages through common parameters. Despite the model's advantages in transfer learning (Aharoni, Johnson, & Firat, 2019), the use of a shared vocabulary and embedding representations forces the model to employ a vocabulary that includes tokens from all the scripts used. Additionally, a recent study (Arivazhagan et al., 2019a), which imposes representational invariance across language, shows improvements in zero-shot translations but a decrease in performance in highly multilingual scenarios for high-resource languages. Therefore, increasing the number of languages varies the quality of the languages already in the system (generally enhancing low-resource pairs but being detrimental for high-resource pairs). Some other disadvantages are that the number of parameters related to vocabulary grow with the number of languages with different scripts and the entire system has to be retrained or add adapters when adding new languages.

2.2 Language-specific Encoder-Decoders

We can classify the different proposed approaches within this category by the degree of parameter sharing between languages.

Sharing parameters. Firat et al. (2016b) proposed extending a bilingual recurrent neural machine translation architecture (Bahdanau et al., 2015) to a multilingual case (Vázquez et al., 2019; Lu et al., 2018) by designing a shared attention-based mechanism between the language-specific encoders and decoders to create a language independent representation. These architectures provide the flexibility for each language to be trained with its own vocabulary, preventing problems related to the addition of several scripts in the same model, especially when some of them are underrepresented. However, as the language-specific components rely on the shared modules, modifying those components to add a new language or add further data to the system would require retraining the whole system (or alternatively, add adapters). Neubing and Hu (2018) proposed a language addition method, based on model fine-tuning, to fast adapt a preexisting model to low-resource source languages. Unlike our work, this work focused on bilingual results and did not consider the effect of the method on previous translation directions. Lakew et al. (2018) proposed a model based on the addition of

new languages to an already trained system by vocabulary adaptation and transfer learning. While limited, it required some retraining to adapt the model to the new task; this resulted in variations in the translation quality of initial languages varying when adding new ones.

No sharing. The system proposed by Escolano, Costa-jussà and Fonollosa (2019) is trained on language-specific encoders and decoders based on joint training without parameter or vocabulary-sharing and on enforcing a compatible representation between the jointly trained languages. The advantage of the approach is that it does not require retraining to add new languages and increasing the number of languages does not affect the quality of the languages already in the system. However, the system has to be trained on a multiway parallel corpus and the system does not scale well when there is a large number of languages in the initial system, since all the encoders and decoders have to be simultaneously trained.

In this study, we are extending the proposal by the same authors (Escolano et al., 2021) which consists of a joint training of language-specific encoders and decoders without forcing an intermediate representation by means of a specific distance. In this sense, our approach does not require a multiway parallel corpus. Compared to the previous study, we are further analyzing the limitations and strengths of our model in terms of fine-tuning, multilingual attention or attention mismatches and the quality of the intermediate representations.

3. Proposed Method

Our proposed approach trains a separate encoder and decoder for each of the N languages available. We do not share any parameter across these modules, which allows us to add new languages incrementally without retraining the entire system. In contrast to Escolano et al. (2019), we do not force the intermediate representation to be the same, and therefore, we do not require multi-parallel corpus to train our system.

We denote the encoder and the decoder for the ith 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. The training algorithm for this procedure is described in Algorithm 1. 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.

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, the basic joint training enables the addition of new languages without the need to retrain the existing modules. Suppose we want to add language $N + 1$. To do so, we must have parallel data between $N + 1$ and any language in the system. As an illustration, let us assume that we have $L_{N+1} - L_i$ parallel data. Then, we can set up a new bilingual system with language L_{N+1} as the source and language L_i 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 decoder (d_{li}) as the decoder of the new $L_{N+1}L_i$ system and we freeze it. During training, we optimize the cross-entropy

Algorithm 1 Multilingual training step

procedure MULTILINGUALTRAININGSTEP
  $N \leftarrow$ Number of languages in the system
  $S = \{s_{0,0}, ..., s_{N,N}\} \leftarrow \{(e_i, d_j)\}$
  $E = \{e_0, ..., e_N\} \leftarrow$ Language-specific encs.
  $D = \{d_0, ..., d_N\} \leftarrow$ Language-specific decs.
  for $i \leftarrow 0$ to N **do**
    for $j \leftarrow 0$ to N **do**
      if $s_{i,j} \in S$ **then**
        $l_i, l_j = get\_parallel\_batch(i, j)$
        $train(s_{i,j}(e_i, d_j), l_i, l_j)$

between the generated tokens and L_i reference data but update only the parameters of to the L_{N+1} encoder ($e_{l_{N+1}}$). By doing so, we train $e_{l_{N+1}}$ 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} decoder can also be trained as a bilingual system by freezing the L_i encoder and training the decoder of the $L_i - L_{N+1}$ system by optimizing the cross-entropy with the L_{N+1} reference data. See Figure 1 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.

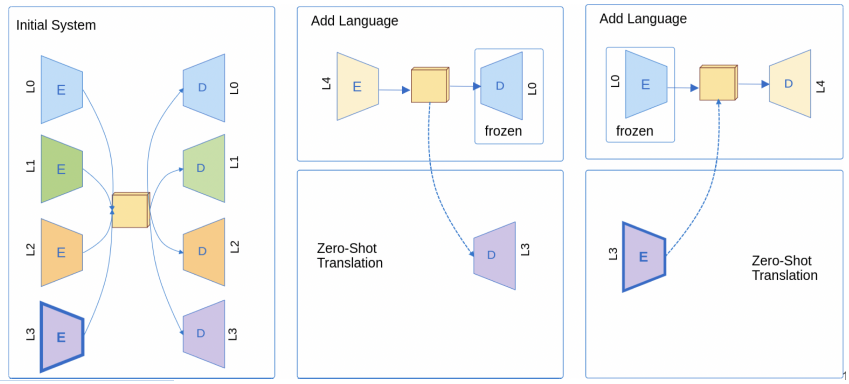

Figure 1: Block Scheme. (Left) Initial Joint Training. (Middle) Adding a new language in the source side with parallel data $L_0 - L_4$ and obtaining zero-shot translation with $L_0 to L_1, L_2, L_3$. (Right) Adding a new language on the target side.

4. Experiments in Multilingual Machine Translation

In this section we review the machine translation experiments in different settings. Since the main difference between the shared and the language-specific encoders-decoders lies in whether they retrain the entire system when adding new languages, we accordingly design our experiments to compare this aspect of the systems.

4.1 Data and Implementation

We used 2 million sentences from the *EuroParl* corpus (Koehn, 2005) in German, French, Spanish and English as training data, which included parallel sentences among all combinations of these four languages. For Russian-English, we used 1 million training sentences from the *Yandex* corpus[2]. As validation and test set, we used *newstest2012* and *newstest2013* from WMT[3]. While the training data is not multiparallel, the validation and test sets are multiparallel across all the above languages and allows for evaluation of zero-shot translation. All data were preprocessed using standard Moses scripts (Koehn et al., 2007)

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 and a vocabulary size of 32k subword tokens with Byte Pair Encoding (Sennrich, Haddow, & Birch, 2016) (in total for the shared encoders/decoders and per pair for the language-specific encoder-decoders). The dropout was 0.1 for the shared approach and 0.3 for language-specific encoders/decoders. Both approaches were trained with an effective batch size of 32k tokens for approximately 200k updates, using the validation loss for early stopping. In all cases, we used Adam (Kingma & Ba, 2015) as the optimizer, with learning rate of 0.001 and 4000 warmup steps. 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. For language-specific encoders-decoders using tied embeddings means that we train by using language-wise word embeddings. The idea is that for one language we use the same word embeddings. Tied embeddings in the shared system means that both encoder and decoder share the same word 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, with slight differences due to each language's subword tokenization.

Another important considerations are the hardware requirements of each architecture. Shared architectures require all parameters to be simultaneously allocated on GPU to be updated. This can become a limitation as more languages require bigger models with a higher capacity. This is not the case for the joint training of our language-specific architecture, where the number of encoders and decoders grows linearly with the number of languages, but the number of parameters of each module remains constant. In conjunction with the proposed training method focused on iteratively training each translation direction, our method allows a constant use of GPU resources equivalent to a single translation direction, which holds independently of the number of languages supported by the system. The cost of this constant computational power is additional CPU memory and disk space and access for the language modules not involved in the current translation direction. Even though the addition usage of system's memory and disk, these components are much more available than GPU in terms of price and capacity of the hardware components, making the architecture a good fit for limited resource scenarios.

2. `https://translate.yandex.ru/corpus?lang=en`
3. `http://www.statmt.org`
4. Release v0.6.0 available at `https://github.com/pytorch/fairseq`

Due to the modularity of our approach, this trade-off between computational resources and storage is not observed during inference and incremental training. Only the required encoder and decoder are loaded during these steps, accounting for the same requirements a bilingual system for that translation direction would have.

4.2 Comparing Training Conditions: Joint Training, Incremental Training and Zero-Shot

We evaluate our approach in 3 different settings: (i) the *joint* training, covering all combinations of German, French, Spanish and English; (ii) the *incremental training* for new languages, tested with Russian-English in both directions; and (iii) the *zero-shot* translation, covering all combinations between Russian and the rest of the languages.

	Shared	LangSpec	Shared^{RU}
de-en	**25.04**	24.54	26.25
de-es	25.01	**25.02**	25.65
de-fr	25.14	**25.49**	25.92
en-de	21.51	**22.01**	22.11
en-es	28.19	**29.53**	29.78
en-fr	28.67	**29.74**	29.63
es-de	20.21	**20.31**	20.52
es-en	26.93	**27.75**	28.72
es-fr	29.59	**30.08**	30.53
fr-de	19.81	**19.97**	19.66
fr-en	26.29	**26.55**	27.98
fr-es	29.03	**29.07**	29.43

Table 2: Joint training. In bold, best global results.

	Shared^{RU}	LangSpec
ru-en	24.62	**27.54**
en-ru	20.03	**23.94**
ru-de	**16.52**	13.77
ru-es	**23.12**	21.08
ru-fr	**22.04**	19.85
de-ru	**17.27**	16.99
es-ru	**18.78**	18.46
fr-ru	**17.83**	17.47

Table 3: Incremental training of new language translation and zero-shot.

In contrast to our proposed approach, 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}). We use the *Shared* version when comparing to our *jointly trained* system in Table 2, and the *Shared*^{RU} version when *incrementally* adding new languages and performing *zero-shot* translations.

Joint training Table 2 shows that our proposed language-specific encoder-decoders outperforms the shared approach in all cases. Note that the gain of our proposed architecture over the shared system corresponds to an average of 0.4 BLEU improvement per language pair.

Incremental training Table 3 shows that, when adding a new language into the system, the language-specific encoder-decoders outperform the shared baseline system by 2.9 BLEU for the direction of Russian-English and 3.9 BLEU points for the opposite direction. 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). As such, the language-specific encoder-decoders are able to outperform the shared architecture when adding a new language and a new domain by learning from the previous information in the frozen modules. The improvement of the system is similar in both directions, even when the decoder has never been trained with the additional data in a different domain.

We observe that adding Russian to the shared system (*Shared*^{RU}) improves its performance on most joint training languages by 0.9 BLEU point on average (see *Shared* vs *Shared*^{RU} in Table 2), except for the French-German pair, so there is both positive and negative transfer learning to the initial languages. This variation, even if being small in our experiments, has been proven to be larger in both positive and negative directions in previous works when the variation of resources for language pairs varies further (Arivazhagan et al., 2019b). This is not the case for the language-specific encoder-decoders, in which all the modules are frozen when adding new languages and there is no transfer from added languages to initial languages by design. Moreover, retraining the shared encoder-decoder to add a new language took an entire week, whereas the incremental training with the language-specific encoder-decoders was performed in only one day. Finally, and to see if there exists positive transfer for new added languages, we trained a bilingual system for Russian-English, and we obtained 26.36 BLEU for Russian-to-English and 22.86 BLEU for the inverse direction. In the case of including Russian in the initial system, our language-specific architecture does not have a negative transfer, whereas the shared system has one for the Russian-English pair.

Zero-Shot The shared encoder-decoder clearly outperforms the language-specific encoder-decoders by 1.3 BLEU points on average. This difference in performance suggests that, while limiting the amount of shared information during training can improve the model's performance, it may also harm zero-shot translations.

4.3 Fine-Tuning

In this section, we want to explore how our proposed architecture re-acts to the effect of fine-tuning on data from new languages added to the system. We want to explore the effect of fine-tuning in the added language pair (Russian-English) and in a language pair already in the initial system (e.g., German-English).

Fine-tuning an added language pair . When 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 in 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,

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**
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**

Effect		Shared^{RU}		LangSpec	
			ft		ft
Transfer	de-en	26.25	**26.74**	24.54	24.56
	en-de	22.11	**22.79**	22.01	22.02
Noise	es-en	28.72	**29.07**	27.75	27.51
	en-es	29.78	**30.36**	29.53	29.44
	fr-en	27.98	**28.29**	26.55	26.42
	en-fr	29.63	**30.21**	29.74	29.57
	ru-en	24.62	25.93	**26.28**	25.77
	en-ru	20.03	21.41	**22.27**	22.12

Table 4: 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.

Table 4 shows how this fine-tuning benefits the Russian-English performance and harms the other directions dramatically in the case of the shared encoder/decoder. These observations on the behavior of fine-tuning on new data resemble the ones obtained for the shared architecture in previous studies (Kudugunta et al., 2019). However, for language-specific encoder/decoder, fine-tuning benefits all pair of languages. Note that Table 4 (top) reports variations only on the results involving English modules, which are the ones modified by this fine-tuning. This fine-tuning has mainly double impact on the entire system. First, it is doing inductive transferring for the Russian-English and therefore, improves its translation quality. Second, it is adding noise/interference to the other language pairs in the system. Inadvertently, this experiment is measuring the robustness of our crosslingual representations. By showing that we can fine-tune and do 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.

Fine-tuning initial language pair . To better understand this behavior we are also replicate fine-tuning on one of the language pairs from the initial training, e.g. German-English. Table 4 (bottom) shows that in this case, neither the shared nor language-specific architecture shows catastrophic forgetting after fine-tuning. An explanation for this difference is the sharing of the embedding table by the shared architecture. By fine-tuning with Russian data, which employs a different set of

tokens from the other languages, we may create an inconsistent token representation for the other languages in the system. On the other hand, as the language-specific architecture has individual embedding tables for each language, its behavior remains constant in both scenarios, even when Russian is added to the already trained system.

4.4 Study on the Impact of Attention Mismatch

We have seen that language-specific encoders-decoders does not suffer from attention mismatch as reported in previous research works (Firat et al., 2016a, 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 are excluding 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 we do not have parallel data among all the language pairs in the initial system. We want to see the impact on performance in these situations. We explore four situations (see Table 5):

1. *(EN)* including parallel data only with English (excluding parallel data from ES-FR, ES-DE, and DE-FR);

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).

From the results in Table 5, we observe that for the first situation, in 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, there is no regularization across languages. As a consequence, there is no information transferred to the intermediate representation. The shared architecture does not have this problem, because, by nature, there is regularization. However, in this situation, we also observe that for the language pairs involving English, our proposed methodology is able to outperform the shared architecture by more than 2 BLEU points in all cases (except for fr-en, where we obtain 0.81 BLEU improvement). In these cases, the fact of not adding regularization at all is aiding the translation quality.

When incrementally adding more languages, for the remaining 3 situations, 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), except for the zero-shot cases, e.g., DE-ES in the column *(EN+FR)*.

	EN		EN+DE		EN+ES		EN+FR	
	Shared	LangSpec	Shared	LangSpec	Shared	LangSpec	Shared	LangSpec
de-en	**24.4**	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	23.8	**24.8**
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**	17.9	**19.84**	**18.53**	16.5
es-en	26.66	**27.15**	26.52	**27.53**	24.78	**27.2**	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**	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**

Table 5: 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*)

4.5 Translation Examples

Table 6 shows some translation examples. In the first examples, we see how for the sentence *Martin was still a teenager when she had him in 1931 during her marriage to lawyer Ben Hagman*, the shared architecture tends to miss some relevant information when translating from Spanish (*she had him in 1931*), which is not the case in the language-specific encoders-decoders. However, when translating from French both shared and language-specific architectures make some translation inaccuracies. For the second example *in recent years , a number of scientists have studied the links between vitamin supplements and cancer .*, the shared architecture generates some term inaccuracies, such as *complexes* or *additives* instead of *supplements*.

5. Analysis of the Intermediate Representations

In this section, we want to better understand the capabilities of our model and we analyze the quality of the intermediate representations by means of a probing classification task. This method has been proposed before as a measure of the cross-lingual capabilities of NMT systems (Eriguchi et al., 2018; Siddhant et al., 2020; McCann et al., 2017; Conneau et al., 2018), using natural language inference (§5.1) and visualization techniques (§5.2).

5.1 Cross-Lingual Natural Language Inference

Given two sentences, a reference and a hypothesis, the natural language inference (NLI) task consists of deciding whether the relationship between them is an *entailment*, *contradiction* or *neutral*. This task has been addressed as a classification problem using the relatedness of the representation of sentences. Following the procedure of Conneau et al. (2018), we train a classifier to perform the task using the encodings of the NMT system as input features. In the original work, the model

System	Languages	Sentence
Reference		Martin was still a teenager when she had him in 1931 during her marriage to lawyer Ben Hagman .
Shared	DE	Martin was a young man when she got Larry during her marriage with Ben Hagman 's lawyer .
	ES	Martin was still a teenager when he was married to Ben Hagman .
	FR	Mary Martin was still a teenager when she gave her birth in 1931 when she was married to Ben Hagman 's lawyer .
	RU	Martin was a teenager when he was born in 1931 during his marriage with Ben Hamman , a lawyer .
Lang-Spec	DE	Martin was still a young boy when she got Larry during her marriage with lawyers Ben Hagman .
	ES	Martin was still a teenager when she had him in 1931 during her marriage with the lawyer Ben Hagman .
	FR	Mary Martin was still a teenager when she was born in 1931 when she was married to the lawyer Ben Hagman .
	RU	Martin was still a teenager when she gave birth to him in 1931 during her marriage with a lawyer Ben Hagman .
Reference		in recent years , a number of scientists have studied the links between vitamin supplements and cancer .
Shared	DE	in recent years , several scientists have studied the link between vitamin additives and cancer .
	ES	in recent years , several scientists have studied the links between vitamin complexes and cancer .
	FR	in recent years , several scientists have studied the links between vitamin supplements and cancer .
	RU	in recent years , many scientists have studied the impact of vitamin additives on cancer development .
Lang-Spec	DE	in recent years , several scientists have studied the link between vitamin supplements and cancer .
	ES	in recent years , a number of scientists have studied the links between vitamin supplements and cancer .
	FR	in recent years , a number of scientists have studied the links between vitamin supplements and cancer .
	RU	in recent years , many scientists have studied the influence of vitamin supplements on cancer development .

Table 6: Translation examples for shared and language-specific architectures.

consisted of a bidirectional recurrent encoder and, as a classifier, two fully connected layers with ReLU and Softmax activation respectively. The classifier is fed with the following combination of the encoding of both the reference and the hypothesis:

$$h = [u, v, |u - v|, u * v] \tag{1}$$

where u is the reference encoding, v is the hypothesis encoding and $*$ is the element multiplication of both vector representations. In that study, encoders were trained specifically on the task of natural language inference, independently for each language and representations were forced to share representation space by means of additional loss terms. For our task, we want to study the shared space

already trained by our proposed multilingual machine translation systems from Section 4. We train a classifier 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 1 while using a Transformer encoder, the contextual embeddings are averaged to create a fixed-sized sentence representation. This approach was previously proposed by Arivazhagan et al. (2019a), where pooling was employed to fix the representation size while not adding extra padding to the data. This was done at the cost of producing an information bottleneck for the classification because all sentence information had to be condensed into a single fixed-size vector, independently of the sentence's length.

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

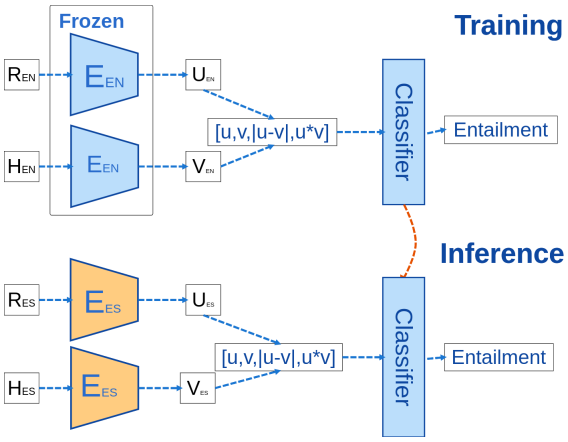

Figure 2: Experiment setup for NLI.

Data and implementation For this task, we use the MultiNLI corpus [5] 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.

We use the exact same encoders trained for the machine translation experiments (§4), 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, which is the only language for which we have training data. Note that in our proposed language-specific encoder-decoders, each language has its own encoder, and both vocabulary and parameters are fully independent.

Results Table 7 shows the results for the XNLI tasks for the output of different encoder layers for the language-specific encoder-decoders. 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 gain a deeper knowledge of it. 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. To better illustrate the model's performance, table 8 show the performance of the proposed methods compared to the shared system with and without Russian. Results show that the method is outperformed by both shared systems,

5. https://cims.nyu.edu/ sbowman/multinli/

showing that sharing parameters may lead to better cross-lingual representations. This difference is more acute for the added Russian encoder that shows more than 10% gap.

From our experiments, we do not observe a correlation between the similarity between language representations and the translation quality for any of the systems on the supervised directions. When comparing tables 8 and 2, German-English results seem to produce worst results than the rest of the tested languages, while it shows the best NLI results overall. This is particularly relevant on the incrementally trained English-Russian language pair, where we observe the most significant difference in NLI performance, even though outperforming the shared model by more than 2 BLEU points. Where these results do correlate is on zero-shot performance, where the Shared architecture outperforms the language-specific in both tasks.

These results indicate that the impact of learning common representation may be more significant for non-supervised tasks such as zero-shot translation and NLI, where parameter sharing acts as an additional regularization step, enforcing better cross-lingual mappings. On the other hand, supervised directions may benefit to some extent from language-specific features and spurious correlations between source and target language that are better captured by the language-specific architecture without parameter sharing.

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.

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

	Shared	Shared^{RU}	LangSpec
en	58,32	**59,96**	54,49
de	59,94	**62,15**	59,52
es	58,4	**60,59**	55,71
fr	59,19	**60,6**	54,81
ru	-	**55,98**	38,94

Table 8: XNLI (en, de, es, fr, ru) accuracy comparison.

5.2 Visualization

In what follows, we use a freely available tool (Escolano et al., 2019) [6], that allows us to visualize intermediate sentence representations. The tool uses the encoder's output fixed-representations as

6. https://github.com/elorala/interlingua-visualization

input data and performs a dimensionality reduction of these data using UMAP (McInnes, Healy, Saul, & Grossberger, 2018). We make the comparison for both Shared^{RU} and language-specific architectures.

Figure 3 shows the intermediate representation of 100 sentences in each languages (German, English, Spanish, French and Russian)of the sentence *recomiendo que se haga el test en cualquier caso* and the corresponding translations. Note that, although all the points seems to be mixed together, the representation of the same sentence in different languages is not placed exactly in the same point in the space, for both shared and language-specific systems. This visualization only pretends to provide a brief qualitative and interpretable analysis of the proposed model.

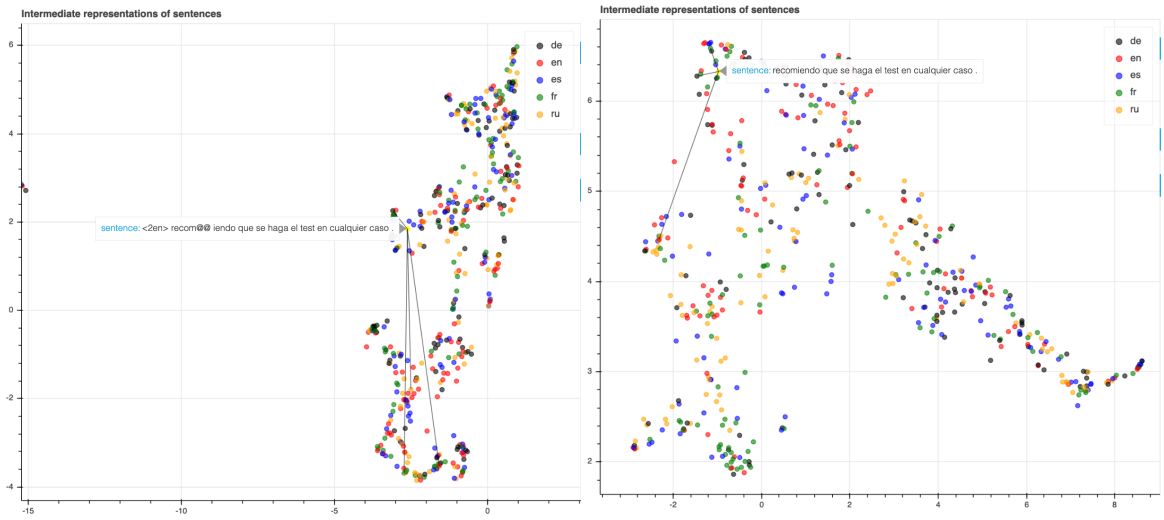

Figure 3: Visualization of sentence representations for the shared (left), language-specific (right) approaches.

6. Conclusions

In this paper, we present a novel method to train language-specific encoders-decoders that allows incremental additions of new languages in the system without having to retrain the entire system or, add 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.

For the initial languages in the system, the language-specific encoder-decoders outperform the shared architecture by 0.4 BLEU points on average. When adding a new language, the language-specific encoder-decoders outperforms the shared ones by 3.4 BLEU points on average and, most importantly, the training of this new language was done in only one day, as opposed to the week taken by the shared system. Additionally, by design, there is no variation in the quality of languages in the initial system when adding a new language.

A further analysis of our model in fine-tuning shows more robustness by avoiding catastrophic forgetting. 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 the application of natural language inference. We see that the higher the encoder layers are, the better the quality. Additionally, an intuitive visualization example shows that the sentences in different languages appear close in the space, but not exactly at the same point. 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 performance of the systems on zero-shot translation.

Our work substantially closes the existing gap between the language-specific and the shared encoders-decoders, while maintaining the flexibility that results from not sharing parameters. Similar to Arivazhagan et al. (2019b), our results suggest that the shared architecture is beneficial for languages that share the same script because of their joint vocabulary and is detrimental for languages that do not share the same script, due to negative transfer between languages in the system. This behaviour is not observed on the language-specific system as each language has its own vocabulary and embeddings.

In the future, we would like to further compare the shared and language-specific encoders-decoders in cases where the languages do not share scripts (e.g. Chinese, Arabic, Russian and Greek) to see if our model has even more advantages over the shared system under these conditions.

Acknowledgments

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 947657).

References

Aharoni, R., Johnson, M., & Firat, O. (2019). Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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., & Wu, Y. (2019b). Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, *abs/1907.05019*.

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

Bapna, A., Arivazhagan, N., & Firat, O. (2019). Simple, scalable adaptation for neural machine translation..

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., & Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pp. 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Dong, D., Wu, H., He, W., Yu, D., & Wang, H. (2015). Multi-task 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)*, pp. 1723–1732, Beijing, China. Association for Computational Linguistics.

Eriguchi, A., Johnson, M., Firat, O., Kazawa, H., & Macherey, W. (2018). Zero-shot cross-lingual classification using multilingual neural machine translation. *CoRR*, *abs/1809.04686*.

Escolano, C., Costa-jussà, M. R., & Fonollosa, J. A. R. (2019). 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*, pp. 236–242, Florence, Italy. Association for Computational Linguistics.

Escolano, C., Costa-jussà, M. R., & Fonollosa, J. A. R. (2021). Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. In *Proceedings of the EACL*.

Escolano, C., Costa-jussà, M. R., Lacroux, E., & Vázquez, P.-P. (2019). 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*, pp. 151–156, Hong Kong, China. Association for Computational Linguistics.

Firat, O., Cho, K., & Bengio, Y. (2016a). Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 866–875, San Diego, California. Association for Computational Linguistics.

Firat, O., Sankaran, B., Al-Onaizan, Y., Yarman Vural, F. T., & Cho, K. (2016b). Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 268–277, Austin, Texas. Association for Computational Linguistics.

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

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., & Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, *5*, 339–351.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y., & LeCun, Y. (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, Vol. 5, pp. 79–86. Citeseer.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (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*, pp. 177–180.

Kudugunta, S., Bapna, A., Caswell, I., Arivazhagan, N., & Firat, O. (2019). Investigating multilingual nmt representations at scale. In *EMNLP/IJCNLP*.

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

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

McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

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

Neubig, G., & Hu, J. (2018). Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 875–880, Brussels, Belgium. Association for Computational Linguistics.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Siddhant, A., Johnson, M., Tsai, H., Ari, N., Riesa, J., Bapna, A., Firat, O., & 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*, pp. 8854–8861. AAAI Press.

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

Zoph, B., & 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*, pp. 30–34, San Diego, California. Association for Computational Linguistics.