Transfer Learning in Multilingual Neural Machine Translation with Dynamic Vocabulary

Surafel M. Lakew[†+], *Aliia Erofeeva*[†], *Matteo Negri*[+], *Marcello Federico*[⋆], *Marco Turchi*[+]

[†]University of Trento, [+]Fondazione Bruno Kessler, Trento, Italy
[⋆]Amazon AI, East Palo Alto, CA 94303, USA
[†]name.surname@unitn.it, [+]surname@fbk.eu, [⋆]marcfede@amazon.com

Abstract

We propose a method to transfer knowledge across neural machine translation (NMT) models by means of a shared dynamic vocabulary. Our approach allows to extend an initial model for a given language pair to cover new languages by adapting its vocabulary as long as new data become available (i.e., introducing new vocabulary items if they are not included in the initial model). The parameter transfer mechanism is evaluated in two scenarios: *i)* to adapt a trained single language NMT system to work with a new language pair and *ii)* to continuously add new language pairs to grow to a multilingual NMT system. In both the scenarios our goal is to improve the translation performance, while minimizing the training convergence time. Preliminary experiments spanning five languages with different training data sizes (i.e., 5k and 50k parallel sentences) show a significant performance gain ranging from +3.85 up to +13.63 BLEU in different language directions. Moreover, when compared with training an NMT model from scratch, our transfer-learning approach allows us to reach higher performance after training up to 4% of the total training steps.

1. Introduction

Neural Machine Translation (NMT) has shown to surpass phrase based Machine Translation approaches not only in high-resource language settings, but also with low-resource [1] and zero-resource translation tasks [2, 3]. Although recent approaches yield promising results, training models in low-resource settings remains a challenge for MT research [4]. [2] have shown that a multilingual NMT (M-NMT) model that utilizes a concatenation of data covering multiple language pairs (including high-resourced ones) can result in better performance in the low-resource translation task. Alternatively, [5] proposed a transfer-learning approach from an NMT "*parent-model*" trained on a high-resource language to initialize a "*child-model*" in a low-resource setting showing consistent translation improvements on the latter task.

Though effective, training models on a concatenation of data covering multiple language pairs or initializing them by transferring knowledge from a parent model does not consider the dynamic nature of new language vocabularies. In relation to how and when model vocabularies are built, there can be two distinct scenarios. In the first one, all the training data for all the language pairs are available since the beginning. In this case, either separate or joint sub-word segmentation models can be applied on the training material to build vocabularies that represent all the data [6, 7]. In the second scenario, training data covering different language directions are not available at the same time (most real-world MT training scenarios fall in this category, in which new data or new needs in terms of domains or language coverage emerge over time). In such cases, either: *i)* new MT models are trained from scratch with new vocabularies built from the incoming training data, or *ii)* the word segmentation rules of a prior (parent) model are applied on the new data to continue the training as a fine-tuning task. In all the scenarios, accurate word segmentation is crucial to avoid out-of-vocabulary (OOV) tokens. However, different strategies for the different training conditions can result in longer training time or performance degradations. More specifically, limiting the target task with the initial model vocabulary will result in: *i)* a word segmentation that is unfavorable for the new language directions and *ii)* a fixed vocabulary/model dimension despite the varying language and training dataset size.

NMT models are not only data-demanding, but also require considerable time to be trained, optimized, and put into use. In particular real-word scenarios, strict time constraints prevent the possibility to deploy and use NMT technology (consider, for instance, emergency situations that require to promptly enable communication across languages [8]). On top of this, when the available training corpora are limited in size, delivering usable NMT systems (i.e., systems that can be used with the requirement of not making severe errors [9]) becomes prohibitive. In summary: *i)* on the data side, acquiring new training material for x undefined languages is costly and not always possible, and *ii)* on the model side, building an NMT system from scratch when new data become available raises efficiency and performance issues that are particularly relevant in low-resource scenarios.

We address these issues by introducing a method to transfer knowledge across languages by means of a dynamic vo-

(*) Work conducted while this author was at FBK.

cabulary. Starting from an initial model, our method allows to build new NMT models, either in a single or multiple language translation directions, by dynamically updating the initial vocabulary to new incoming data. For instance, given a trained German-English NMT system (L_1), the learned parameters can be transferred across models, while adopting new language vocabularies. In our experimental setting we test two transfer approaches:

- `progAdapt`: train a chain of consecutive M-NMT models by transferring the parameters of an initial model for L_1 to new language pairs $L_2 \dots L_N$. In this scenario, the goal is to maximize performance on the new language pairs.
- `progGrow`: progressively introduce new language pairs to the initial model L_1 to create a growing M-NMT model covering N translation directions. In this scenario, the goal is to maximize performance on all the language pairs.

Our experiments are carried out with Italian−English, Romanian−English, and Dutch−English training data sets of different size, ranging from low-resource (50k) to extremely low-resource (5k) conditions.

As such, in a rather different way from previous work [5], we show our transfer-learning approach in a multilingual NMT model with dynamic vocabulary both in the source and target directions. Our contributions are as follows:

- we develop a transfer-learning technique for NMT based on a dynamic vocabulary, which adapts the parameters learned on a parent task (language direction) to cover new target tasks;
- through experiments in different scenarios, we show that our approach improves knowledge transfer across NMT models for different languages, particularly in low-resource conditions;
- we show that, with our transfer learning approach, it is possible to train a faster converging model that achieves better performance than a system trained from scratch.

2. Related work

2.1. Transfer Learning

Recent efforts [10, 11] in natural language processing (NLP) research have shown promising results when transfer-learning techniques are applied to leverage existing models to cope with the scarcity of training data in specific domains or language settings. The advancements in NLP came following a much larger impact of transfer-learning in computer vision tasks, such as classification and segmentation, either using features of ImageNet [12] or by fine-tuning the last layers of a deep neural network [13]. Specific to NLP, pre-trained word embeddings [14] used as input to the first layer of the network have become a common practice. In a broader sense, pre-trained models have been successfully exploited for several NLP tasks. [15] used an MT model as a pre-training step to further contextualize word vectors for downstream tasks like sentiment analysis, question classification, textual entailment, and question answering. In a similar way, a language model is utilized for pre-training in sequence labeling tasks [16], question answering, textual entailment, and sentiment analysis [17].

Close to our approach, [5] explored techniques for transfer-learning across two NMT models. First, a "parent" model is trained with a large amount of available data. Then the encoder-decoder components are transferred to initialize the parameters of a low-resourced "child" model. In this parent-child setting, the decoder parameters of the child model are fixed at the time of fine-tuning. Later, in [18], the parent-child approach has been extended to analyze the effect of using related languages on the source side.

Although this work shares a related approach with [5], we diverge by our hypothesis not to selectively update only the encoder, allowing all the parameters to be updated as a beneficial strategy in our setting. Our strategy is based on both the source→target and target→source translation directions that we consider as transferable. Moreover, our transfer-learning approach relies on a dynamic vocabulary that enforces changes in the trainable parameters of the network in contrast to fixing them[1].

2.2. Multilingual NMT

In a one-to-many multilingual translation scenario, [19] proposed a multi-task learning approach that utilizes a single encoder for the source language and separate attention mechanisms and decoders for each target language. [20] used distinct encoder and decoder networks for modeling multiple language pairs in a *many-to-many* setting. Later, [21] introduced a way to share the attention mechanism across multiple languages. Aimed at avoiding translation ambiguities on the decoder side, a *many-to-one* character level NMT setup [22] and a two/multi-source NMT [23] were also proposed. Inspired by [24], who automatically annotated the source side with artificial flags to manage the politeness level of the output, other works focused on controlling the grammatical voice [25], the text domain [26, 27], and enforcing gender agreement [28]. Simplified yet efficient multilingual NMT approaches have been proposed by [2] and [3]. The approach in [3] applies a language-specific code to words from different languages in a mixed-language vocabulary. The approach in [2], by prepending a *language flag* to the input string, greatly simplified multilingual NMT eliminating the need of having separate encoder/decoder networks and attention mechanism for each new language pair. In this work we follow a similar strategy by incorporating an artificial language flag.

[1] In future work, we plan to further study which parameters are more beneficial if transferred and which part of the network to selectively update.

3. Transfer Learning in M-NMT

In this work, we cast transfer-learning in a multilingual neural machine translation (M-NMT) task as the problem of dynamically changing/updating the vocabulary of a trained NMT system. In particular, transfer-learning across models is assumed to: *i)* include a strategy to add new language-specific items to an existing NMT vocabulary, and *ii)* be able to manage a number of new translation directions in different transfer rounds, either by covering them one at a time (i.e., in a chain where new languages are covered stepwise) or simultaneously (i.e., pursuing all directions at each step). Our investigation focuses on two aspects. The first one is how the parameters of an existing model can be transferred to a target one for a new language pair. The second aspect is how to limit the impact of parameters' transfer on the performance of the initial model as long as new language directions are added. For convenience, we refer to our approach as `TL-DV` (*Transfer-Learning using Dynamic Vocabulary*).

As shown in Figure 1, our transfer-learning approach is evaluated in two conditions:

- `progAdapt`, in which progressive updates are made on the assumption that new target NMT task data become available for one language direction at a time (i.e., new language directions are covered sequentially). In this condition, our goal is to maximize performance on the new target tasks by taking advantage of parameters learned in their parent task;

- `progGrow`, in which progressive updates are made on the same assumption of receiving new target task data as in `progAdapt`, but with the additional goal of preserving the performance of the previous language directions.

We discuss these two scenarios below in §3.2 and §3.3.

3.1. Dynamic Vocabulary

In the defined scenarios, we update the vocabulary V_p of the previous model with the current language direction vocabulary V_c. The approach simply keeps the intersection (same entries) between V_p and V_c, whereas replacing V_p entries with V_c if the entries of the former vocabulary do not exist in the latter. At training time, these new entries are randomly initialized, while the intersecting items maintain the embeddings of the former model. The alternative approach to dynamic vocabulary in a continuous model training is to use the initial model vocabulary V_p, which we refer to as static-vocabulary.

3.2. Progressive Adaptation to New Languages

In this scenario, starting from the `init` model (L_1), we perform progressive adaptation by initializing the training of a model at each step (L_n) with the previous model (L_{n-1}). At time of reloading the model from L_{n-1}, a `TL-DV` update is performed as described in §3. In this approach, the dataset of the initial model is not included at the current training stage. This allows the adaptation to the new language without unnecessary word segmentation that may arise by applying the initial model's segmentation rules. As shown in Figure 1 (left), the adaptation on any of the L_n stages is language independent, though subject to the available training dataset. We refer to the application of this approach in the experimental settings and discussion as `progAdapt`.

3.3. Progressive Growth of Translation Directions

In this scenario, an initial model L_1 is simultaneously adapted to an incremental number of translation directions, under the constraint that the level of performance on L_1 has to be maintained. For a simplified experimental setup, we will incorporate a single language pair (source→target) at a time, when adapting to L_n from L_{n-1} (see Figure 1 (right)). We refer to the application of this approach in the experimental settings and discussion as `progGrow`.

4. Experimental Setting

4.1. Dataset and Preprocessing

Our experimental setting includes the `init` model language pair (German-English) and three additional language pairs (Italian-English, Romanian-English, and Dutch-English) for testing the proposed approaches. We use publicly available datasets from the WIT[3] TED corpus [29]. Table 1 shows the summary of the training, dev, and test sets. To simulate an extremely low-resource (M_{ELR}) and low-resource (M_{LR}) model settings, 5K and 50K sentences are sampled from the last three language pairs' training data.

At the preprocessing step, we first tokenize the raw data and remove sentences longer than 70 tokens. As in [2], we prepend a "language flag" on the source side of the corpus for all multilingual models. For instance, if a German source is paired with an English target, we append `<2ENG>` at the beginning of source segments. Next, a shared byte pair encoding (BPE) model [6] is trained using the union of the source and target sides of each language pair. Following [30], the number of BPE segmentation rules is set to $8,500$ for the data size used in our experimental setting. At different levels of training (L_i), a BPE model with respect to the language pairs is then used to segment the training, dev, and test data into sub-word units. While, the vocabulary size of the `init` is fixed, the vocabulary varies in the consecutive training stages depending on the overlap of sub-word units and lexical similarity between two language pairs.

4.2. Experimental Settings

All systems are trained using the Transformer [31] model implementation of the OpenNMT-tf sequence modeling framework[2] [32]. At training time, to alternate between dynamic

[2]https://github.com/OpenNMT/OpenNMT-tf

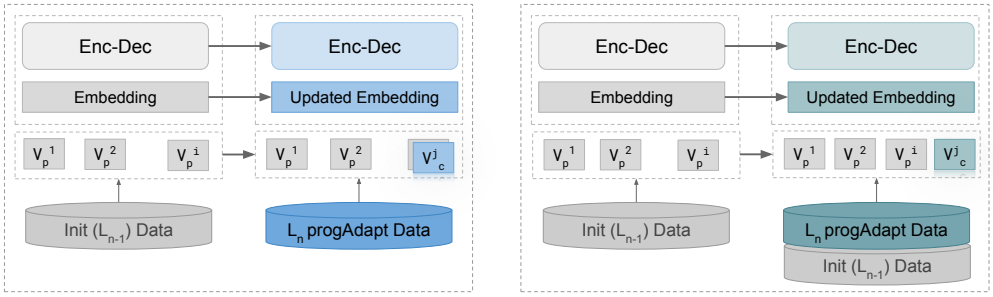

Figure 1: *Transfer-Learning; (left) from an initial NMT model to a new language pair, model is applied after inserting the new vocabulary entries, for instance, the initial model L_{n-1} parameters are transfered to L_n with the updated embedding space (i.e., keeping V_p^1, V_p^2 as overlapping entries, while replacing the non-overlapping V_p^i with V_c^j new language vocabularies), and (right) from an initial model L_{n-1} to L_n, but incorporating both the previous and new language pair data and vocabulary entries.*

Table 1: *Languages and dataset sizes for train, dev, and test sets of the* `init` *model for De-En direction and other pairs assumed to be received progressively (It-En, Ro-En, Nl-En).*

Language	Train	Dev	Test	Received
German(De)-En	200k	1497	1138	`init`
Italian(It)-En	5k/50k	1501	1147	L_2
Romanian(Ro)-En	5k/50k	1633	1129	L_3
Dutch(Nl)-En	5k/50k	1726	1181	L_4

and static vocabulary, we utilized an updated version of the script within the same framework. For all trainings, we use LazyAdam, a variant of the Adam optimizer [33], with an initial learning rate constant of 2 and a dropout [34, 35] of 0.3. The learning rate is increased linearly in the early stages (*warmup_training_steps*=16, 000), and afterwards it is decreased with an inverse square root of the training step.

To train our models using Transformer, we employ a uniform setting with 512 hidden units and embedding dimension, and 6 layers of self-attention encoder-decoder network. The training batch size is of 4096 sub-word tokens. At inference time, we use a beam size of 4 and a batch size of 32. Following [31] and for a fair comparison, all baseline experiments are run for 100k training steps, i.e., all models are observed to converge within these steps. The consecutive experiments converge in variable training steps. However, to make sure a convergence point is reached, all restarted experiments on L_i are run for additional 50K steps. All models are trained on a `GeForce-GTX-1080` machine with a single GPU. Systems are compared in terms of BLEU [36] using the *multi-bleu.perl* implementation[3], on the single references of the official IWSLT test sets.

4.3. Baseline Models

To evaluate and compare with our approach, we train single language pair baseline models corresponding to the newly introduced language pairs at each L_i training stage. The baseline models, referred to as `Bi-NMT`, are separately trained from scratch in a bi-directional setting (i.e., source $\leftrightarrow$ target). In addition, we report scores from a multilinugal (`M-NMT`) model trained with the concatenation of all available data in each training stage. The alternative baseline are built by fine-tuning the `init` model. These models use the vocabulary (word segmentation rules) of the `init` model, avoiding the proposed dynamic vocabulary. This fine-tuning approach is prevalent in continued model trainings, for adapting NMT models [37, 38] or improving zero-shot and low-resource translation tasks [39, 40, 41]. For the alternative baseline where we fine-tune `init` with its static-vocabulary, we observed that results were mostly analogous to `Bi-NMT` models. Hence, we avoided this comparison in this work and relied on the former baselines.

[3] A script from the Moses SMT toolkithttp://www.statmt.org/moses

5. Results and Discussion

Experiments are performed using the `progAdapt` (see §3.2) and `progGrow` (§3.3) approaches. The experimental results with the associated discussion are presented in Table 2 for models characterized by relatively low-resource data (M_{LR}), and in Table 3 for an extremely low-resource condition (M_{ELR}). In both dataset conditions, the performance of the proposed approaches is compared with the baseline systems (`Bi-NMT` and `M-NMT`, see §4.3).

The `init` model which is trained with a data size 4X larger than M_{LR} and 40X the size of M_{ELR}, achieves BLEU scores of 26.74 and 23.30, respectively, for the De-En and En-De directions. In Table 2 and 3, the `progAdapt` is reported for each training stage (i.e., L_2, L_3, and L_4), whereas the `progGrow` is reported for the final stage L_4. Moreover, Table 4 analyzes the effect of language relatedness and training stage reordering in our `TL-DV` approach. Bold highlighted BLEU scores show the best performing approach, while the $^{\uparrow\downarrow}$ arrows indicate statistically significant differences of the hypothesis against the better performing baseline (`M-NMT`) using bootstrap resampling ($p < 0.05$) [42].

Table 2: *M_{LR} models performance i) at L_1 for the* `init` *De-En direction and baseline (*`Bi-NMT`*) It-En, Ro-En, and Nl-En directions, ii) at $L_{2/3/4}$ for* `progAdapt`*, and iii) at L_4 for the* `progGrow` *approach.*

	Dir	De-En	It-En	Ro-En	Nl-En
`Init/Bi-NMT`	>	**26.74**	25.21	10.80	21.75
	<	23.30	22.39	12.94	19.75
`M-NMT`	>	24.14	26.42	22.17	24.00
	<	21.80	23.57	17.35	21.25
`ProgAdapt`	>	-	↑**30.08**	↑**24.43**	↑**26.36**
	<	-	↑**26.24**	↑**20.31**	↑**25.52**
`ProgGrow`	>	26.22	↑**29.61**	**23.23**	**24.78**

Table 3: *M_{ELR} models performance i) at L_1 for the* `init` *De-En direction and baseline (*`Bi-NMT`*) It-En, Ro-En, and Nl-En directions, ii) at $L_{2/3/4}$ for* `progAdapt`*, and iii) at L_4 for the* `progGrow` *approach.*

	Dir	De-En	It-En	Ro-En	Nl-En
`Init/Bi-NMT`	>	**26.74**	7.64	4.56	5.69
	<	23.30	5.25	3.86	5.14
`M-NMT`	>	24.96	**16.26**	**12.67**	**15.59**
	<	21.67	10.38	8.67	12.72
`ProgAdapt`	>	-	↓15.16	↓11.03	↓11.52
	<	-	↑**14.40**	↑**11.10**	**13.57**
`ProgGrow`	>	25.61	↓15.02	↓11.20	↓13.56

5.1. Low-Resource Setting

For each language pair (i.e., It-En, Ro-En, and Nl-En), the results of the baseline models `Bi-NMT` trained using the available 50K parallel data (M_{LR} setting) are presented in the first two rows of Table 2. The `progAdapt` results are reported from three consecutive adaptations to new language directions. These include the `init` to It-En, followed by the adaptation to Ro-En, and then to Nl-En. Compared to the corresponding `Bi-NMT` and M-NMT models, all of the three progressive adaptations using the dynamic vocabulary technique achieved a higher performance gain.

If we look at the specific level of adaption (L_i) against the `Bi-NMT`, we observe that the It-En direction showed a +4.87 and +3.85 gain for the En and It target, respectively. When we take this model and continue the adaptation to Ro-En and Nl-En, we see a similar trend where the highest gain is observed on L_3 for the Ro-En direction with +13.63 and +7.37 points. These significant improvements over the baseline models tell us that transfer-learning using dynamic vocabulary in a multilingual setting is a viable direction. Its capability to quickly tune the representation space of the `init` model to deliver improved results is an indication of the importance of using different word representations for each language pair[4].

In case of the `progGrow`, we observed a similar improvement trend as in the `progAdapt` approach. The results are reported from the final stage (L_4) of the model growth, but improvements are consistent throughout the L_2 and L_3 stages. The `M-NMT` outperformed the `Bi-NMT` models except for De-En pair. However, compared to the multilingual model as an alternative method for achieving cross-lingual transfer-learning, our approach shows improvements in the consecutive training stages. Overall, our observation is that the suggested `progGrow` model can accommodate new translation directions when the data are received. Most importantly, improvements are observed for these newly introduced languages without altering the performance of the `init` model in the De-En direction.

Specific to each language direction, It-En shows a comparable performance with the `progAdapt` approach, whereas in case of Ro-En and Nl-En a small degradation ranging from 0.47 (De-En) to 1.58 (Nl-En) is observed. The loss in performance is likely due to the increased ambiguities in the encoder side of the `progGrow` model, where at both training and inference time there does not exist a disambiguation mechanism between languages except the prepended language flag. This observation, which sheds a light on our initial expectation of more data aggregation benefiting the model performance, requires further investigation.

5.2. Extremely Low-Resource Setting

In a similar way with what we observed in the M_{LR} experiments, the baseline models in the extremely low-resource setting demonstrate poor performance. Looking at our approaches, we observe a relatively higher gain at the first stage of `progAdapt` and `progGrow`. For instance, for the It-En pair there is a +7.52 improvement compared to the +4.87 in the M_{LR} models (see Table 2) over the `Bi-NMT` model. In the subsequent additional language directions (i.e., Ro-En and Nl-En), we also observe a similar trend. However, in comparison with the M-NMT, both of our approach perform poorly when translating to the En target. The main reason for this could be the aggregation of all the available data for a single run in the `M-NMT` model, while our approaches exploit data when it becomes available in a continuous training. Alternatively the distance between each language pair could play a significant role when we adapt in an extremely sparse data.

prog-Adapt/Grow with Related Languages. When related language pairs are consecutively added (L_{n-1} and L_n) at each training stages, our `TL-DV` approach showed the best performance. For instance, for the Nl-En experiments, we changed the sequence of the added language pair

[4]We reserve the adaptation from the `init` model directly to all the three new language pairs and the comparison with the current setting for future work.

Table 4: *M_{LR} and M_{ELR} models performance at L_1 for* `progAdapt` *and* `progGrow` *approaches in a closely related De-En (*`init`*) and Nl-En language pairs setting.*

		M_{LR}		M_{ELR}	
	Dir	De-En	Nl-En	De-En	Nl-En
`ProgAdapt`	$>$	-	↑**27.23**		**16.21**
	$<$	-	↑25.51		**15.86**
`ProgGrow`	$>$	26.62	↑**26.41**	26.52	↑**15.52**

moving from a random order to a sequence based on the similarity to the `init` model. Table 4 shows the results from `progAdapt` and `progGrow`, when the Nl-En pair is used at the L_1 training stage. The M_{LR} results confirm the trend observed in Table 2, however, with a relatively better performance when translating in to English. Most importantly, the M_{ELR} results show a consistent and larger gain of $+4.69$ (Nl-En) and $+2.29$ (En-Nl) with the `progAdapt`, and $+1.96$ (Nl-En) with `progGrow` compared to the corresponding results in Table 3. Thus, we emphasize on the degree of language similarity as a direct influencing factor when incorporating a new language pair both in `progAdapt` and `progGrow` approaches. .

Prog-Adapt/Grow with Faster Convergence. The other main advantage of our `TL-DV` approach comes from the time a model takes to restart from the `init` model and reach a convergence point with better performance. In all experiments with our `TL-DV` approach a converged model is found within 10K steps for M_{ELR} and 20K for M_{LR} training settings. Compared to ≈100K steps needed by a model trained from scratch to reach good performance, our approach takes only 4% to 20% of training steps with significantly higher performance. For instance, taking into consideration the M_{ELR} models, Figure 2 illustrates the steps required for the baseline systems to converge (Table 3), in comparison with our approach where `progGrow` shows to converge slightly faster than `progAdapt`. However, with the relatively larger data of the M_{LR} models, the `progAdapt` approach proves to converge much faster than `progGrow`, for the reason that the newly introduced vocabulary and training dataset sizes are smaller compared to the concatenation of the `init` and L_i data.

We further analyzed the influence of shared vocabularies between models L_i and L_{i+1} on the performance of `TL-DV`. For this discussion, we took the `progAdapt` M_{LR} model from all stages. Figure 3 summarizes the improvement differences from consecutive models in relation to the percentage of shared vocabularies. For instance, `init` and the L_2 (It-En) model vocabularies have a 47% overlap, whereas L_3 and L_4 share 53% and 51% with the previous model. The interesting aspect of the shared vocabulary comes from the increase in model performance with a higher fraction of shared vocabulary entires. Thus, a larger number of shared parameters between two consecutive models allows for a better gain in performance of the latter.

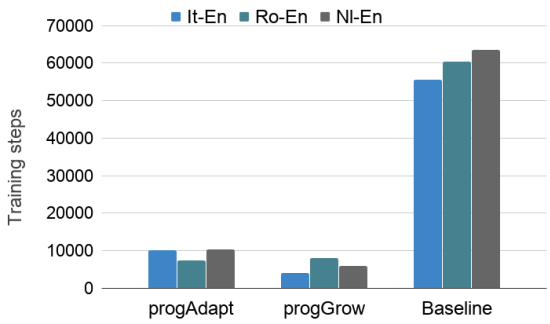

Figure 2: *Model training steps number comparison for the three different language pairs between the baseline (rightmost) and the proposed approaches in the M_{ELR} setting.*

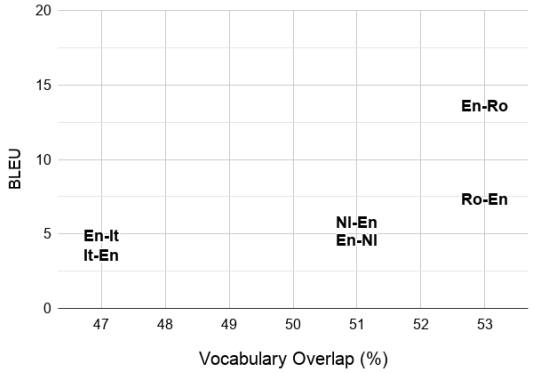

Figure 3: *The difference in performance between the baseline and* `progAdapt` *models (Tgt→Src and Src→Tgt directions) in relation with the shared vocabulary between model L_i and new language pair model L_{i+1}.*

The results achieved by the transfer-learning with dynamic vocabulary approach in two different training size conditions show that: *i)* adapting a trained NMT model to a new language pair improves performance on the target task significantly, and *ii)* it is possible to train a model faster to achieve better performance. Overall, the capability of injecting new vocabularies for new language pairs in the initial model is a crucial aspect for efficient and fast adaptation steps.

6. Conclusions

In this work, we proposed a transfer-learning approach within a multilingual NMT. Experimental results show that our dynamic vocabulary based transfer-learning improves model performance in a significant way of up to 9.15 in an extremely low-resource and up to 13.0 BLEU in a low-resource setting over a bilingual baseline model.

In future work, we will focus on finding the optimal way of transferring model parameters. Moreover, we plan to test our approach for various languages and language varieties.

7. Acknowledgments

This work has been partially supported by the EC-funded project ModernMT (H2020 grant agreement no. 645487). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Moreover, we thank the Erasmus Mundus European Program in Language and Communication Technology.

8. References

[1] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multilingual neural machine translation," *arXiv preprint arXiv:1606.04164*, 2016.

[2] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *arXiv preprint arXiv:1611.04558*, 2016.

[3] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[4] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *arXiv preprint arXiv:1706.03872*, 2017.

[5] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[6] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[7] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[8] W. Lewis, "Haitian creole: How to build and ship an mt engine from scratch in 4 days, 17 hours, & 30 minutes," in *14th Annual conference of the European Association for machine translation*. Citeseer, 2010.

[9] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french," *Computer Speech & Language*, vol. 49, pp. 52–70, 2018.

[10] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 328–339.

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2017.

[12] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[14] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" *arXiv preprint arXiv:1804.06323*, 2018.

[15] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.

[16] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[18] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," *arXiv preprint arXiv:1708.09803*, 2017.

[19] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation." in *ACL (1)*, 2015, pp. 1723–1732.

[20] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[21] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[22] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *arXiv preprint arXiv:1610.03017*, 2016.

[23] B. Zoph and K. Knight, "Multi-source neural translation," *arXiv preprint arXiv:1601.00710*, 2016.

[24] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 35–40.

[25] H. Yamagishi, S. Kanouchi, T. Sato, and M. Komachi, "Controlling the voice of a sentence in japanese-to-english neural machine translation," in *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, 2016, pp. 203–210.

[26] W. Chen, E. Matusov, S. Khadivi, and J.-T. Peter, "Guided Alignment Training for Topic-Aware Neural Machine Translation," in *Association for Machine Translation in the Americas (AMTA)*, jul 2016. [Online]. Available: http://arxiv.org/abs/1607.01628

[27] C. Chu, R. Dabre, and S. Kurohashi, "An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 385–391. [Online]. Available: http://arxiv.org/abs/1701.03214%0Ahttp://aclweb.org/anthology/P17-2061

[28] M. Elaraby, A. Y. Tawfik, M. Khaled, and A. Osama, "Gender Aware Spoken Language Translation Applied to English-Arabic," in *Proceedings of the Second International Conference on Natural Language and Speech processing*, 2018. [Online]. Available: https://arxiv.org/pdf/1802.09287.pdf

[29] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[30] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*, 2017, pp. 18–27.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[32] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[35] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[37] P. Michel and G. Neubig, "Extreme adaptation for personalized neural machine translation," *arXiv preprint arXiv:1805.01817*, 2018.

[38] D. Vilar, "Learning hidden unit contribution for adapting neural machine translation models," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, vol. 2, 2018, pp. 500–505.

[39] S. M. Lakew, Q. F. Lotito, N. Matteo, T. Marco, and F. Marcello, "Improving zero-shot translation of low-resource languages," in *14th International Workshop on Spoken Language Translation*, 2017.

[40] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.

[41] J. Gu, H. Hassan, J. Devlin, and V. O. Li, "Universal neural machine translation for extremely low resource languages," *arXiv preprint arXiv:1802.05368*, 2018.

[42] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 4, 2004, pp. 388–395.