Efficiently Reusing Old Models Across Languages via Transfer Learning

Tom Kocmi **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{kocmi,bojar}@ufal.mff.cuni.cz

Abstract

Recent progress in neural machine translation is directed towards larger neural networks trained on an increasing amount of hardware resources. As a result, NMT models are costly to train, both financially, due to the electricity and hardware cost, and environmentally, due to the carbon footprint. It is especially true in transfer learning for its additional cost of training the "parent" model before transferring knowledge and training the desired "child" model. In this paper, we propose a simple method of reusing an already trained model for different language pairs where there is no need for modifications in model architecture. Our approach does not need a separate parent model for each investigated language pair, as it is typical in NMT transfer learning. To show the applicability of our method, we recycle a Transformer model trained by different researchers and use it to seed models for different language pairs. We achieve better translation quality and shorter convergence times than when training from random initialization.

1 Introduction

Neural machine translation (NMT), the current prevalent approach to automatic translation, is known to require large amounts of parallel training sentences and an extensive amount of training time on dedicated hardware. The total training time significantly increases, especially when training strong baselines, searching for best hyperparameters or training multiple models for various language pairs.

Schwartz et al. (2019) analyzed 60 papers from top AI conferences and found out that 80% of them target accuracy over efficiency, and only a small portion of papers argue for a new efficiency result. They also noted that the increasing financial cost of the computations could make it difficult for researchers to engage in deep learning research or limit training strong baselines. Furthermore, increased computational requirements have also an environmental cost. Strubell et al. (2019) estimated that training a single Transformer "big" model produces 87 kg of CO_2 and that the massive Transformer architecture parameter search produced 298 tonnes of CO_2.[1]

However, a lot of research has been already invested into cutting down the long training time by the design of NMT model architectures, promoting self-attentive (Vaswani et al., 2017) or convolutional (Gehring et al., 2017) over recurrent ones (Bahdanau et al., 2014) or the implementation of heavily optimized toolkits (Junczys-Dowmunt et al., 2018).

In this paper, we propose a novel view on reusing already trained "parent" models without the need to prepare a parent model in advance or modify its training hyper-parameters. Furthermore, we propose a second method based on a vocabulary transformation technique that makes even larger improvements, especially for languages using an alphabet different from the re-used parent model. Our transfer learning approach leads to better performance as well as faster convergence speed of the "child" model compared to training the model from scratch. We document that our methods are

© 2020 The authors. This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC-BY-ND.

[1] The paper reports numbers based on the U.S. energy mix.

not restricted only to low-resource languages, but they can be used even for high-resource ones.

Previous transfer learning techniques (Neubig and Hu, 2018; Kocmi and Bojar, 2018) rely on a shared vocabulary between the parent and child models. As a result, these techniques separately train parent model for each different child language pair. In contrast, our approach can re-use one parent model for multiple various language pairs, thus further lowering the total training time needed.

In order to document that our approach is not restricted to parent models trained by us, we re-use parent model trained by different researchers: we use the winning model of WMT 2019 for Czech-English language pair (Popel et al., 2019).

The paper is organized as follows: Section 2 describes the method of Direct Transfer learning, including our improvement of vocabulary transformation. Section 3 presents the model, training data, and our experimental setup. Section 4 describes the results of our methods followed by the analysis in Section 5. Related work is summarized in Section 6 and we conclude the discussion in Section 7.

2 Transfer Learning

In this work, we present the use of transfer learning to reduce the training time and improve the performance in comparison to training from random initialization even for high-resource language pairs.

Transfer learning is an approach of using training data from a related task to improve the accuracy of the main task in question (Tan et al., 2018). One of the first transfer learning techniques in NMT was proposed by Zoph et al. (2016). They used word-level NMT and froze several model parts, especially embeddings of words that are shared between parent and child model.

We build upon the work of Kocmi and Bojar (2018), who simplified the transfer learning technique thanks to the use of subword units (Wu et al., 2016) in contrast to word-level NMT transfer learning (Zoph et al., 2016) and extended the applicability to unrelated languages.

Their only requirement, and also the main disadvantage of the method, is that the vocabulary has to be shared and constructed for the given parent and child languages jointly, which makes the parent model usable only for the particular child language pair. This substantially increases the overall training time needed to obtain the desired NMT system for the child language pair.

The method of Kocmi and Bojar (2018) consists of three steps: (1) construct the vocabulary from both the parent and child corpora, (2) train the parent model with the shared vocabulary until convergence, and (3) continue training on the child training data.

Neubig and Hu (2018) call such approaches warm-start, where we use the child language pair to influence the parent model. In our work, we focus on the so-called cold-start scenario, where the parent model is trained without a need to know the language pair in advance. Therefore we cannot make any modifications of the parent training to better handle the child language pair. The cold-start transfer learning is expected to have slightly worse performance than the warm-start approach. However, it allows reusing one parent model for multiple child language pairs, which reduces the total training time in comparison to the use of warm-start transfer learning.

We present two approaches: Direct Transfer that ignores child-specific vocabulary altogether; and Transformed Vocabulary, which modifies vocabulary of the already trained parent. Thus, one parent model can be used for multiple child language pairs.

2.1 Direct Transfer

Direct Transfer can be seen as a simplification of Kocmi and Bojar (2018). We ignore the specifics of the child vocabulary and train the child model using the parent vocabulary. We suppose that the subword vocabulary can handle the child language pair, although it is not optimized for it.

We take an already trained model and use it as initialization for a child model using a different language pair. We continue the training process without any change to the vocabulary or hyper-parameters. This applies even to the training parameters, such as the learning rate or moments.

This method of continued training on different data while preserving hyper-parameters is used under the name "continued training" or "fine-tuning" (Hinton and Salakhutdinov, 2006; Miceli Barone et al., 2017), but it is mostly used as a domain adaptation within a given language pair.

Direct Transfer relies on the fact that the current NMT uses subword units instead of words. The subwords are designed to handle unseen words or even characters, breaking the input into shorter units, possibly down to individual bytes as implemented, for example, by Tensor2Tensor (Vaswani et al., 2018).

Avg. # per:	Child-specific Sent.	Word	EN-CS vocab. Sent.	Word
Odia	95.8	3.7	496.8	19.1
Estonian	26.0	1.1	56.2	2.3
Finnish	22.9	1.1	55.9	2.6
German	27.4	1.3	55.4	2.5
Russian	33.3	1.3	134.9	5.3
French	42.0	1.6	65.7	2.5

Table 1: Average number of tokens per sentence (column "Sent.") and average number of tokens per word (column "Word") when the training corpus is segmented by child-specific or parent-specific vocabulary. "Child-specific" represents the effect of using vocabulary customized for examined language. "EN-CS" corresponds to the use of English-Czech vocabulary.

	Segmented sentence
Original	Сьерра-Леоне
EN-RU	Сьерра_▌-_▌Леоне_
EN-CS	С▌ь▌ер▌ра▌_▌-_▌\▌10▌51▌;▌е▌о▌не_

Figure 1: Illustration of segmentation of Russian phrase (gloss: Sierra Leone) with English-Czech and English-Russian vocabulary from our experiments. The character ▌ represents splits.

This property ensures that the parent vocabulary can, in principle, serve for any child language pair, but it can be highly suboptimal, segmenting child words into too many subwords.

We present an example of a Russian phrase and its segmentation based on English-Czech or English-Russian vocabulary in Figure 1. When using child-specific vocabulary, the segmentation works as expected, splitting the phrase into three tokens. However, when we use a vocabulary that contains only the Cyrillic alphabet[2] and not many longer sequences of characters, the sentence is split into 13 tokens. We can notice that English-Czech wordpiece vocabulary is missing a character "Л", thus it breaks it into the byte representation "`\1051;`".

We examine the influence of parent-specific vocabulary on the training dataset of the child. Table 1 documents the segmenting effect of different vocabularies. If we compare the child-specific and parent-specific ("EN-CS") vocabulary, the average number of tokens per sentence or per word increases more than twice. For example, German has twice as many tokens per word compared to its child-specific vocabulary, and Russian has four times more tokens due to Cyrillic. Odia is affected even more.

[2]This happened solely due to noise in the Czech-English parent training data.

Input: Parent vocabulary (an ordered list of parent subwords) and the training corpus for the child language pair.
Generate child-specific vocabulary with the maximum number of subwords equal to the parent vocabulary size;
for *subword S in parent vocabulary* **do**
  if *S in child vocabulary* **then**
    continue;
  else
    Replace position of S in the parent vocabulary with the first unused child subword not contained in the parent;
  end
end
Result: Transformed parent vocabulary

Algorithm 1: Transforming parent vocabulary to contain child subwords and match positions for subwords common for both of language pairs.

Thus, we see that ignoring the vocabulary mismatch introduces a problem for NMT models in the form of an increasing split ratio of tokens. As expected, this is most noticeable for languages using different scripts.

2.2 Vocabulary Transformation

Using parent vocabulary roughly doubles the number of subword tokens per word, as we showed in the previous section. This problem would not happen with child-specific vocabulary. However, we are using an already trained parent with its vocabulary. Therefore, we propose a vocabulary transformation method that replaces subwords in the parent wordpiece (Wu et al., 2016) vocabulary with subwords from the child-specific vocabulary.

NMT models associate each vocabulary item with its vector representation (embedding). When transferring the model from the parent to the child, we decide which subwords should preserve their embedding as trained in the parent model and which embeddings should be remapped to new subwords from the child vocabulary. The goal is to preserve embeddings of subwords that are contained in both parent and child vocabulary. In other words, we reuse embeddings of subwords common to both parent and child vocabularies and reuse the vocabulary entries of subwords not occurring in the child

data for other, unrelated, subwords that the child data need. Obviously, the embeddings for these subwords will need to be retrained.

Our Transformed Vocabulary method starts by constructing the child-specific vocabulary with the size equal to the parent vocabulary size (the parent model is trained, thus it has a fixed number of embeddings). Then, as presented in Algorithm 1, we generate an ordered list of child subwords, where subwords known to the parent vocabulary are on the same positions as in the parent vocabulary, and other subwords are assigned arbitrarily to places where parent-only subwords were stored.

We experimented with several possible mappings between the parent and child vocabulary. We tried to assign subwords based on frequency, by random assignment, or based on Levenshtein distance of parent and child subwords. However, all the approaches reached comparable performance; neither of them significantly outperformed the others. One exception is when assigning all subwords randomly, even those that are shared between parent and child. This method leads to worse performance, having several BLEU points lower than other approaches. Another approach would be to use pretrained subword embeddings similarly as proposed Kim et al. (2019). However, in this paper, we focus on showing, that transfer learning can be as simple as not using any modifications at all.

3 Experiments

In this section, we first provide the details of the NMT model used in our experiments and the examined set of language pairs. We then discuss the convergence and a stopping criterion and finally present the results of our method for recycling the NMT model as well as improvements thanks to the vocabulary transformation.

3.1 Parent Model and its Training Data

In order to document that our method functions in general and is not restricted to our laboratory setting, we do not train the parent model ourselves. Instead, we recycle two systems trained by Popel et al. (2019), namely the English-to-Czech and Czech-to-English winning models of WMT 2019 News Translation Task. It is important to note, that we use two parent models and for experiments we always use the parent model with English on the same side, e.g. English-to-Russian child has English-to-Czech as a parent. We leave experimenting with different parents or various combinations for future works, because the goal of this work is to make approach most simple.

We decided to use this model for several reasons. It is trained to translate into Czech, a high-resource language that is dissimilar from any of the languages used in this work.[3] At the same time, it is trained using the state-of-the-art Transformer architecture as implemented in the Tensor2Tensor framework.[4] (Vaswani et al., 2018). We use Tensor2Tensor in version 1.8.0.

The model is described in Popel (2018). It is based on the "Big GPU Transformer" setup as defined by Vaswani et al. (2017) with a few modifications. The model uses reverse square root learning rate decay with 8000 warm-up steps and a learning rate of 1. It uses the Adafactor optimizer, the batch size of 2900 subword units, disabled layer dropout.

Due to the memory constraints, we drop training sentences longer than 100 subwords. We use child hyper-parameter setting equal to the parent model. However, some hyper-parameters like learning rate, dropouts, optimizer, and others could be modified for the training of the child model. We leave these experiments for future work.

We train models on single GPU GeForce 1080Ti with 11GB memory. In this setup, 10000 training steps take on average approximately one and a half hours. Popel et al. (2019) trained the model on 8 GPUs for 928k steps, which means that on the single GPU, the parent model would need at least 7424k steps, i.e. more than 45 days of training.

In our experiments, we train all child models up to 1M steps and then take the model with the best performance on the development set. Because some of the language pairs, especially the low-resource ones, converge within first 100k steps, we use a weak early stopping criterion that stops the training whenever there was no improvement larger than 0.5% of maximal reached BLEU over the past 50% of training evaluations (minimum of training steps is 100k). This stopping criterion makes sure that no model is stopped prematurely.

[3]The linguistically most similar language of our language selection is Russian, but we do not transliterate Cyrillic into Latin script. Therefore, the system cannot associate similar Russian and Czech words based on appearance.

[4]`https://github.com/tensorflow/tensor2tensor`

Language pair	Pairs	Training set	Development set	Test set
EN - Odia	27k	Parida et al. (2018)	Parida et al. (2018)	Parida et al. (2018)
EN - Estonian	0.8M	Europarl, Rapid	WMT dev 2018	WMT 2018
EN - Finnish	2.8M	Europarl, Paracrawl, Rapid	WMT 2015	WMT 2018
EN - German	3.5M	Europarl, News commentary, Rapid	WMT 2017	WMT 2018
EN - Russian	12.6M	News Commentary, Yandex, and UN Corpus	WMT 2012	WMT 2018
EN - French	34.3M	Commoncrawl, Europarl, Giga FREN, News commentary, UN corpus	WMT 2013	WMT dis. 2015

Table 2: Corpora used for each language pair. The names specify the corpora from WMT 2018 News Translation Task data. Column "Pairs" specify the total number of sentence pairs in training data.

Language pair	Baseline		Direct Transfer		Transformed Vocab			
	BLEU	Steps	BLEU	Steps	BLEU	Steps	Δ BLEU	Speed-up
English-to-Odia	3.54	45k	0.26	47k	**6.38** ‡*	**38k**	2.84	16 %
English-to-Estonian	16.03	95k	**20.75** ‡	**75k**	20.27 ‡	**75k**	4.24	21 %
English-to-Finnish	14.42	420k	16.12 ‡	**255k**	**16.73** ‡*	270k	2.31	36 %
English-to-German	36.72	270k	38.58 ‡	190k	**39.28** ‡*	**110k**	2.56	59 %
English-to-Russian	27.81	1090k	27.04	630k	**28.65** ‡*	**450k**	0.84	59 %
English-to-French	33.72	820k	34.41 ‡	**660k**	**34.46** ‡	720k	0.74	12 %
Estonian-to-English	21.07	70k	24.36 ‡	**30k**	**24.64** ‡*	60k	3.57	14 %
Russian-to-English	30.31	980k	23.41	**420k**	**31.38** ‡*	700k	1.07	29 %

Table 3: Translation quality and training time. "Baseline" is trained from scratch with its own vocabulary and child corpus only. "Direct Transfer" is initialized with parent model using the parent vocabulary and continues training. "Transformed Vocab" has the same initialization but merges the parent and child vocabulary as described in Section 2.2. Best score and lowest training time in each row in bold. The statistical significance is computed against the baseline (‡) or against "Direct Transfer" (*). Last two columns show improvements of Transformed Vocabulary in comparison to the baseline.

3.2 Studied Language Pairs

We use several child language pairs to show that our approach is useful for various sizes of corpora, language pairs, and scripts. To cover this range of situations, we select languages in Table 2. Future works could focus also on languages outside from Indo-European family, such as Chinese.

Another decision behind selecting these language pairs is to include language pairs reaching various levels of translation quality. This is indicated by automatic scores of the baseline setups ranging from 3.54 BLEU (English-to-Odia) to 36 BLEU (English-to-German)[5], see Table 3.

The sizes of corpora are in Table 2. The smallest language pair is English-Odia, which uses the Brahmic writing script and contains only 27 thousand training pairs. The largest is the high-resource English-French language pair.

For most of the language pairs, we use training data from WMT (Bojar et al., 2018).[6] We use the training data without any preprocessing, not even tokenization.[7] See Table 2 for the list of used corpora for each language pair. For some languages, we have opted out from using all available corpora in order to experiment on languages containing various magnitudes of parallel sentences.

For high-resource English-French language pair, we perform a corpora cleaning using language detection Langid.py (Lui and Baldwin, 2012). We drop all sentences that are not recognized as the correct language. It removes 6.5M (15.9 %) sentence pairs from the English-French training corpora.

4 Results

All reported results are calculated on the test data and evaluated with SacreBLEU (Post, 2018). The results are in Table 3. We discuss separately the training time, automatically assessed translation quality using the parent and the Transformed Vocabulary, and comparison to Kocmi and Bojar (2018) in the following sections.

Baselines use the same architecture, and they are trained solely on the child training data with the use of child-specific vocabulary. We compute

[5]The systems submitted to WMT 2018 for English-to-German translation have better performance than our baseline due to the fact, that we decided not to use Commoncrawl, which artificially made English-German parallel data less resourceful.

[6]http://www.statmt.org/wmt18/

[7]While the recommended best practice in past WMT evaluations was to use Moses tokenizer. It is not recommended for Tensor2Tensor with its build-in tokenizer any more.

statistical significance with a paired bootstrap resampling (Koehn, 2004). We use 1000 samples and a confidence level of 0.05. Statistically significant improvements are marked by ‡.

4.1 Direct Transfer Learning

First, we compare the Direct Transfer learning in contrast to the baseline. We see that Direct Transfer learning is significantly better than the baseline in both translation directions in all cases except for Odia and Russian, which we will discuss later. We get improvements for various language types, as discussed in Section 3.2. The largest improvement is of 4.72 BLEU for the low-resource language pair of Estonian-English, but we also get an improvement of 0.69 BLEU for the high-resource pair French-English.

The results are even more surprising when we take into account the fact that the model uses the parent vocabulary, and it is thus segmenting words into considerably more subwords. This suggests that the Transformer architecture generalizes very well to short subwords. However, the worse performance of English-Odia and English-Russian can be attributed to the different writing script. The Odia script is not contained in the parent vocabulary at all, leading to segmenting of each word into individual bytes, the only common units with the parent vocabulary. Therefore, to avoid problems with filtering, we increase the filtering limit of long sentences during training from 100 to 500 subwords for these two language pairs (see Section 3.1).

4.2 Results with Transformed Vocabulary

As the results in Table 3 confirm, Transformed Vocabulary successfully tackles the problem of the child language using a different writing script. We see "Transformed Vocab" delivering the best performance for all language pairs except for English-to-Estonian, significantly improving over baseline and even over "Direct Transfer" in most cases.

4.3 Training Time

In the introduction, we discussed that recent development in NMT focuses mainly on the performance over efficiency (Schwartz et al., 2019). Therefore, in this section, we discuss the amount of training time required for our method to converge. We are reporting the number of updates (i.e. steps) needed to get the model used for evaluation.[8]

[8]Another possibility would be to report wall-clock time. However, that is influenced by server load and other factors. The number of steps is better for the comparison as long as the batch size stays the same across experiments.

	Language pair	Baseline	Transf. vocab	Warm Start
BLEU	To Estonian	16.03	20.27	**20.75**
	To Russian	27.81	28.65	**29.03** ‡
	From Estonian	21.07	24.64	**26.00** ‡
	From Russian	30.31	**31.38**	31.15
Steps	To Estonian	95k	**75k**	735k
	To Russian	1090k	**450k**	1510k
	From Estonian	70k	**60k**	700k
	From Russian	980k	**700k**	1465k

Table 4: Comparison of our Transformed Vocabulary method with Kocmi and Bojar (2018) (abridged as "Warm Start"). The top half of table compares results in BLEU, the bottom half the number of steps needed to convergence. Steps of Kocmi and Bojar (2018) method are reported as the sum of parent and child training, due to the nature of the method.

We see in Table 3 that both our methods converged in a lower number of steps than the baseline. For the Transformed Vocabulary method, we get a speed-up of 12–59 %. The reduction in the number of steps is most visible in English-to-German and English-to-Russian. It is important to note that the number of steps to the convergence is not precisely comparable, and some tolerance must be taken into account. It is due to the fluctuation in the training process. However, in neither of our experiments, Transformed Vocabulary is slower than baseline. Thus we conclude that our Transformed Vocabulary method takes fewer training steps to finish training than training a model from scratch.

4.4 Comparison to Kocmi and Bojar (2018)

We replicated the experiments of Kocmi and Bojar (2018) with the identical framework and hyperparameter setting in order to compare their method to ours. We experiment with Estonian-English and Russian-English language pair in both translation directions. Their approach needs an individual parent for every child model, so we train four models: two English-to-Czech and two Czech-to-English on the same parent training data as Kocmi and Bojar (2018). All vocabularies contain 32k subwords. We compare their method with our Transformed Vocabulary. Furthermore, the results of Direct Transfer in Table 3 are also comparable with this experiment.

In Table 4, we see that their method reaches a slightly better performance in three translation models, where English-to-Russian and Estonian-to-English are significantly (‡) better than Transformed Vocabulary technique; the other two are on par with our method, which is understandable. The Transformed Vocabulary cannot outperform

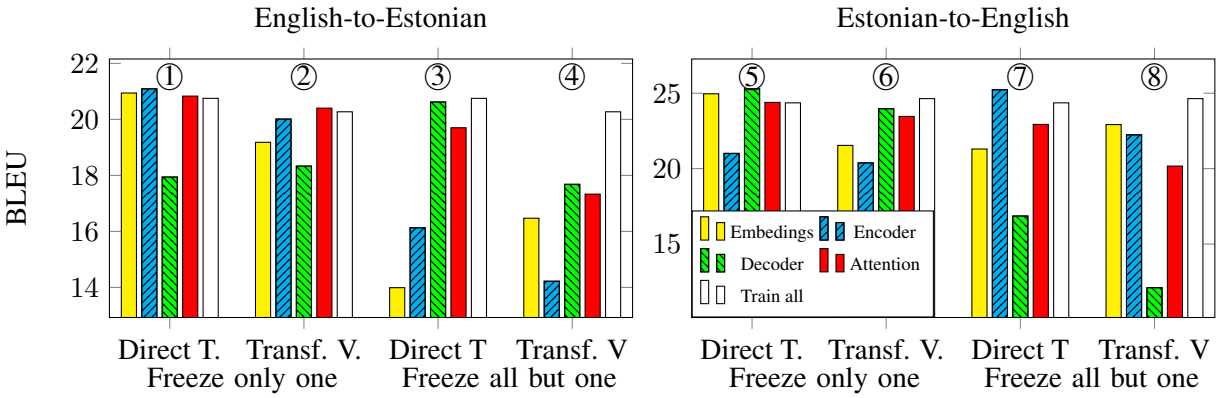

Figure 2: Child BLEU scores when trained with some parameters frozen. The left plot shows English-to-Estonian and the right is Estonian-to-English. In both plots, the first two groups are experiments where one component is frozen and the second two are when all components but one are frozen.

the warm-start technique since the warm-start parent model has the advantage of being trained with the vocabulary prepared for the investigated child.

However, when we compare the total number of steps needed to reach the performance, both our approaches are significantly faster than Kocmi and Bojar (2018). The most substantial improvements are roughly ten times faster for Estonian-to-English, and the smallest difference for English-to-Russian is two times faster. This is mostly because their method first needs to train the parent model that is specialized for the child, while our method can directly re-use any already trained model. Moreover, we can see that their method is even slower than the baseline model.

5 Analysis by Freezing Parameters

To discover which transferred parameters are the most helpful for the child model and which need to be changed the most, we follow the analysis used by Thompson et al. (2018): When training the child, we freeze some of the parameters.

Based on the internal layout of the Transformer model in Tensor2Tensor, we divide the model into four components. (i) Word embeddings (shared between encoder and decoder) map each subword unit to a dense vector representation. (ii) The encoder component includes all the six feed-forward layers converting the input sequence to the deeper representation. (iii) The decoder component consists again of six feed-forward layers preparing the choice of the next output subword unit. (iv) The multi-head attention is used throughout encoder and decoder, as self-attention layers interweaved with the feed-forward layers.

We run two sets of experiments: either freeze only one out of the four components and leave the rest of the model updating or freeze everything but the examined component. We also test it on two translation directions: English-to-Estonian in the left hand part of Figure 2 and Estonian-to-English in the right hand part. In both cases, English-Czech (in the corresponding direction, i.e. with English on the correct side) serves as the parent. We discuss individual components separately, indexing the experiments ① to ⑧.

Similarly to Thompson et al. (2018) in domain adaptation, we observe that parent embeddings serve well in Direct Transfer, freezing them has a minimal impact compared to the baseline in ① and ⑤. The frozen embeddings in Transformed Vocabulary (②, ⑥) results in significant performance drops which can be attributed to the arbitrary assignment of embeddings to new subwords.

The comparison of all but embeddings frozen in ④ and ⑧ (Transformed Vocabulary) is interesting. In ⑧, the performance of the network can be recovered close to the baseline by retraining either parent source embeddings or the encoder. These two components can compensate for each other. This differs from the case with English reused in the source (④) where updating embeddings to the child language is insufficient: the decoder must be updated to produce fluent output in the new target language and even with the decoder updated, the loss compared to the baseline is quite substantial.

The most important component for transfer learning is generally the component handling the new language: decoder in English-to-Estonian and encoder in the reverse. With this component fixed, the performance drops the most with this component fixed (①, ②, ⑤, ⑥) and among the least with this

component free to update (③, ④, ⑦, ⑧). This confirms that at least for examined language pair, the Transformer model lends itself very well to encoder or decoder re-use.

Other results in Figure 2 reveal that the architecture can compensate for some of the training deficiencies. Freezing the encoder ①, ② (resp. decoder for Estonian-to-English ⑤, ⑥) or attention is not that critical as the frozen decoder (resp. encoder). The bad result of the encoder ③, ④ (resp. decoder ⑦, ⑧) being the only non-frozen component shows that model is not capable of providing all the needed capacity for the new language, unlike the self-attention where the loss is not that large. This behaviour correlates with our intuition that the model needs to update the most the component that handles the differing language with the parent model (in our case Czech).

All in all, these experiments illustrate the robustness of the Transformer model that it is able to train and reasonably well utilize pre-trained weights even if they are severely crippled.

6 Related Work

This paper focuses on re-using an existing NMT model in order to improve the performance in terms of training time and translation quality without any need to modify the model or pre-trained weights.

Lakew et al. (2018) presented two model modifications for multilingual MT and showed that transfer learning could be extended to transferring from the parent to the first child, followed by the second child and then the third one. They achieved improvements with dynamically updating embeddings for the vocabulary of a target language.

The use of other language pairs for improving results for the target language pair has been approached from various angles. One option is to build multilingual models (Liu et al., 2020), ideally so that they are capable of zero-shot, i.e. translating in a translation direction that was never part of the training data. Johnson et al. (2017) and Lu et al. (2018) achieve this with a unique language tag that specifies the desired target language. The training data includes sentence pairs from multiple language pairs, and the model implicitly learns translation among many languages. In some cases, it achieves zero-shot and can translate between languages never seen together. Gu et al. (2018) tackled the problem by creating universal embedding space across multiple languages and training many-to-one MT system. Firat et al. (2016) propose multi-way multi-lingual systems. Their goal is to reduce the total number of parameters needed to train multiple source and target models. In all cases, the methods are dependent on a special training schedule.

The lack of parallel data in low-resource language pairs can also be tackled by unsupervised translation (Artetxe et al., 2018; Lample et al., 2018). The general idea is to train monolingual autoencoders for both source and target languages separately, followed by mapping both embeddings to the same space and training simultaneously two models, each translating in a different direction. In an iterative training, this pair of NMT systems is further refined, each system providing training data for the other one by back-translating monolingual data (Sennrich et al., 2016).

For very closely related language pairs, transliteration can be used to generate training data from a high-resourced pair to support the low-resourced one as described in Karakanta et al. (2018).

7 Conclusion

In this paper, we focus on a setting where existing models are re-used without any preparation for knowledge transfer of original model ahead of its training. This is a relevant and prevailing situation in academia due to computing restrictions, and industry, where updating existing models and scaling to more language pairs is essential. We evaluate and propose methods of re-using Transformer NMT models for any "child" language pair regardless of the original "parent" training languages and especially showing, that no modification is better than training from scratch.

The techniques are simple, effective, and applicable to models trained by others which makes it more likely that our experimental results will be replicated in practice. We showed that despite the random assignment of subwords, the Transformed Vocabulary improves the performance and shortens the training time of the child model compared to training from random initialization.

Furthermore, we showed that this approach is not restricted to low-resource languages, and we documented that the highest improvements are (expectably) due to the shared English knowledge. Moreover, we confirmed the robustness of the Transformer and its ability to achieve good results in adverse conditions like very fragmented subword units or parts of the network frozen.

The warm-start approach by Kocmi and Bojar (2018) performs slightly better than our Transformed Vocabulary, but it needs to be trained for a significantly longer time. This leaves room for approaches that also focus on the efficiency of the training process. We perceive our approach as a technique for increasing the performance of a model without an increase in training time. Thus, re-using older models in cold-start scenario of transfer learning can be used in standard NMT training pipelines without any performance or speed losses instead of random initialization as is the common practice currently.

Acknowledgements

This study was supported in parts by the grants 18-24210S of the Czech Science Foundation and 825303 (Bergamot) of the European Union. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071).

References

Artetxe, Mikel, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Bojar, Ondřej, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels, October. Association for Computational Linguistics.

Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. 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*, pages 866–875, San Diego, California, June. Association for Computational Linguistics.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Gu, Jiatao, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hinton, Geoffrey E. and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Johnson, Melvin, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Vigas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July. Association for Computational Linguistics.

Karakanta, Alina, Jon Dehdari, and Josef van Genabith. 2018. Neural machine translation for low-resource languages without parallel corpora. *Machine Translation*, 32(1):167–189, Jun.

Kim, Yunsu, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In Korhonen, Anna, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1246–1257. Association for Computational Linguistics.

Kocmi, Tom and Ondřej Bojar. 2018. Trivial Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, November.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Lakew, Surafel M, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. *IWSLT*.

Lample, Guillaume, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Lu, Yichao, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Belgium, Brussels, October. Association for Computational Linguistics.

Lui, Marco and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July. Association for Computational Linguistics.

Miceli Barone, Antonio Valerio, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark, September. Association for Computational Linguistics.

Neubig, Graham and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium, October. Association for Computational Linguistics.

Parida, Shantipriya, Ondrej Bojar, and Satya Ranjan Dash. 2018. Odiencorp: Odia-english and odia-only corpus for machine translation. In *Smart Computing and Informatics*. Springer.

Popel, Martin, Dominik Machaček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. 2019. English-czech systems in wmt19: Document-level transformer. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 342–348, Florence, Italy, August. Association for Computational Linguistics.

Popel, Martin. 2018. CUNI Transformer Neural MT System for WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 486–491, Belgium, Brussels, October. Association for Computational Linguistics.

Post, Matt. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation, Volume 1: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

Schwartz, Roy, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *arXiv preprint arXiv:1907.10597*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July. Association for Computational Linguistics.

Tan, Chuanqi, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer.

Thompson, Brian, Huda Khayrallah, Antonios Anastasopoulos, Arya D. McCarthy, Kevin Duh, Rebecca Marvin, Paul McNamee, Jeremy Gwinnup, Tim Anderson, and Philipp Koehn. 2018. Freezing subnetworks to analyze domain adaptation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 124–132, Belgium, Brussels, October. Association for Computational Linguistics.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Vaswani, Ashish, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA, March. Association for Machine Translation in the Americas.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, November. Association for Computational Linguistics.