

# Unsupervised Neural Machine Translation Initialized by Unsupervised Statistical Machine Translation

Benjamin Marie      Atsushi Fujita

National Institute of Information and Communications Technology  
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan  
{bmarie, atsushi.fujita}@nict.go.jp

## Abstract

Recent work achieved remarkable results in training neural machine translation (NMT) systems in a fully unsupervised way, with new and dedicated architectures that rely on monolingual corpora only. In this work, we propose to define unsupervised NMT (UNMT) as NMT trained with the supervision of synthetic bilingual data. Our approach straightforwardly enables the use of state-of-the-art architectures proposed for supervised NMT by replacing human-made bilingual data with synthetic bilingual data for training. We propose to initialize the training of UNMT with synthetic bilingual data generated by unsupervised statistical machine translation (USMT). The UNMT system is then incrementally improved using back-translation. Our preliminary experiments show that our approach achieves a new state-of-the-art for unsupervised machine translation on the WMT16 German-English news translation task, for both translation directions.

## 1 Introduction

Machine translation (MT) systems usually require a large amount of bilingual data, produced by humans, as supervision for training. However, finding such data remains challenging for most language pairs, as it may not exist or may be too costly to manually produce.

In contrast, a large amount of monolingual data can be easily collected for many languages, for instance from the Web.<sup>1</sup> Previous work proposed many ways for taking advantage of the monolingual data in order to improve translation models trained on bilingual data. These methods usually exploit existing accurate translation models and have shown to be useful especially when targeting

low-resource language pairs and domains. However, they usually fail when the available bilingual data is too noisy or too small to train useful translation models. In such scenarios, the use of pivot languages or unsupervised machine translation are possible alternatives.

Recent work has shown remarkable results in training MT systems using only monolingual data in the source and target languages. Unsupervised statistical (USMT) and neural (UNMT) machine translation have been proposed (Artetxe et al., 2018b; Lample et al., 2018b). State-of-the-art USMT (Artetxe et al., 2018b; Lample et al., 2018b) uses a phrase table induced from source and target phrases, extracted from the monolingual data, paired and scored using bilingual word, or  $n$ -gram, embeddings trained without supervision. This phrase table is plugged in a standard phrase-based SMT framework that is used to translate target monolingual data into the source language, i.e., performing a so-called back-translation. The translated target sentences and their translations in the source language are paired to form synthetic parallel data and to train a source-to-target USMT system. This back-translation/re-training step is repeated for several iterations to refine the translation model of the system.<sup>2</sup> On the other hand, state-of-the-art UNMT (Lample et al., 2018b) uses bilingual sub-word embeddings. They are trained on the concatenation of source and target monolingual data in which tokens have been segmented into sub-word units using, for instance, byte-pair-encoding (BPE) (Sennrich et al., 2016b). This method can learn bilingual embeddings if the source and target languages have in common some sub-word units. The sub-word embeddings are then used to initialize the lookup tables in the encoder and decoder of the UNMT system. Follow-

<sup>1</sup>See for instance the Common Crawl project: <http://commoncrawl.org/>

<sup>2</sup>Previous work did not address the issue of convergence and rather fixed the number of iterations to perform for these refinement steps.

ing this initialization step, UNMT mainly relies on denoising autoencoder as language model during training and on latent representation shared across the source and target languages for the encoder and the decoder.

While the primary target of USMT and UNMT is low-resource language pairs, their possible applications for these language pairs remain challenging, especially for distant languages,<sup>3</sup> and have yet to be demonstrated. On the other hand, unsupervised MT achieves impressive results on resource-rich language pairs, with recent and quick progresses, suggesting that it may become competitive, or more likely complementary, to supervised MT in the near future.

In this preliminary work, we propose a new approach for unsupervised MT to further reduce the gap between supervised and unsupervised MT. Our approach exploits a new framework in which UNMT is bootstrapped by USMT and uses only synthetic parallel data as supervision for training. The main outcomes of our work are as follows:

- We propose a simplified USMT framework. It is easier to set up and train. We also show that using back-translation to train USMT is not suitable and underperform.
- We propose to use supervised NMT framework for the unsupervised NMT scenarios by simply replacing true parallel data with synthetic parallel data generated by USMT. This strategy enables the use of well-established NMT architectures with all their features, without assuming any relatedness between source and target languages in contrast to previous work.
- We empirically show that our framework leads to significantly better UNMT than USMT on the WMT16 German–English news translation task, for both translation directions.

## 2 What is truly unsupervised in this paper?

Since the term “unsupervised” may be misleading, we present in this section what aspects of this work are truly unsupervised.

<sup>3</sup>Mainly due to the difficulty of training accurate unsupervised bilingual word/sub-word embeddings for distant languages (Søgaard et al., 2018).

As previous work, we define “unsupervised MT” as MT that does not use human-made translation pairs as bilingual data for training. Nonetheless, MT still needs some supervision for training. Our approach uses as supervision synthetic bilingual data generated from monolingual data.

“Unsupervised” qualifies only the training of MT systems on bilingual parallel data of which at least one side is synthetic. For tuning, it is arguably unsupervised in some of our experiments or supervised using a small set of human-made bilingual sentence pairs. We discuss “unsupervised tuning” in Section 3.2. For evaluation, it is fully supervised, as in previous work, since we use a human-made test set to evaluate the translation quality.

Even if our systems are trained without human-made bilingual data, we can still argue that the monolingual corpora used to generate synthetic parallel data have been produced by humans. Source and target monolingual corpora in our experiments (see Section 5.1) could include some comparable parts. Moreover, we cannot ensure that they do not contain any human-made translations from which our systems can take advantage during training. Finally, we use SMT and NMT architectures, set and use their hyper-parameters (for instance, the default parameters of the Transformer model) in our framework that have already shown to give good results in supervised MT.

## 3 Simplified USMT

Our USMT framework is based on the same architecture proposed by previous work (Artetxe et al., 2018b; Lample et al., 2018b): a phrase table is induced from monolingual data and used to compose the initial USMT system that is then refined iteratively using synthetic parallel data. We propose the following improvements and discussions to simplify the framework and make it faster with lighter models (see also Figure 1):

- Section 3.1: we propose several modifications to rely more on compositional phrases and to simplify the phrase table induction compared to the method proposed by Artetxe et al. (2018b)
- Section 3.2: we discuss the feasibility of unsupervised tuning.
- Section 3.3: we propose to replace the back-translation in the refinement steps with for-

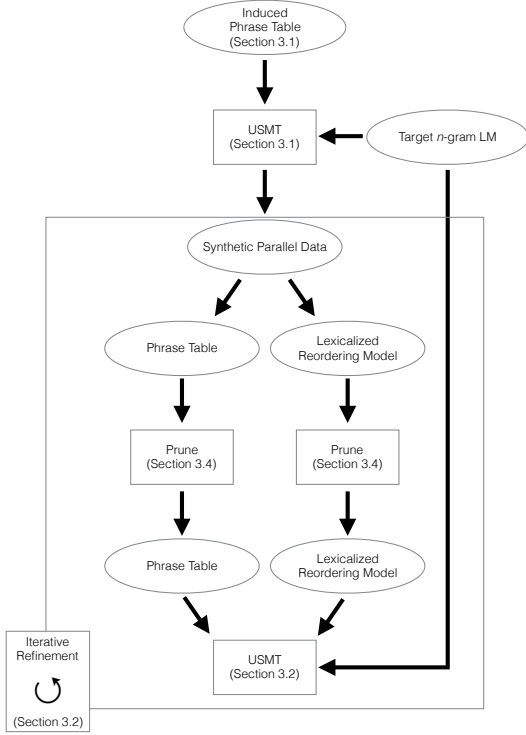


Figure 1: Our USMT framework.

ward translation to improve translation quality and to remove the need of simultaneously training models for both translation directions.

- Section 3.4: we propose to prune the phrase table to speed up the generation of synthetic parallel data during the refinement steps.

### 3.1 Phrase table induction

As proposed by Artetxe et al. (2018b) and Lample et al. (2018b), the first step of our approach for USMT is an unsupervised phrase table induction that only takes as inputs a set of source phrases, a set of target phrases, and their respective embeddings, as illustrated by Figure 2. Artetxe et al. (2018b) regarded the most frequent uni-grams, bi-grams, and tri-grams in the monolingual data as phrases. The embedding of each n-gram is computed with a generalization of the skip-gram algorithm (Mikolov et al., 2013). Then, source and target n-gram embedding spaces are aligned in the same bilingual embedding space without supervision (Artetxe et al., 2018a). Lample et al. (2018b)’s method also works at n-gram level, but computes phrase embeddings as proposed by Zhao et al. (2015): performing the element-wise addition of the embeddings of the

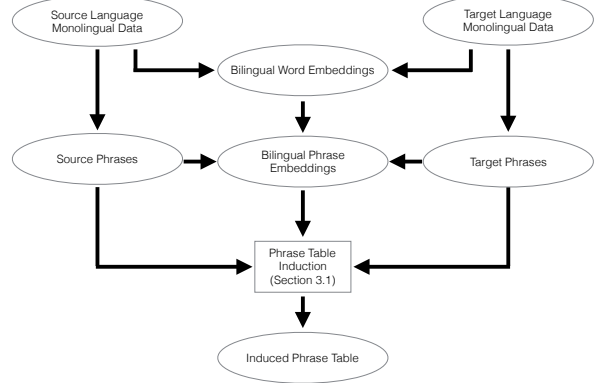


Figure 2: Phrase table induction.

component words of the phrase, also trained on the monolingual data and aligned in the same bilingual embedding space. This method can estimate embedding for compositional phrases but not for non-compositional phrases unlike Artetxe et al. (2018b)’s method. Interestingly, Artetxe et al. (2018b)’s method yields significantly better results at the first iteration of USMT, that uses the induced phrase table, but performs similarly to Lample et al. (2018b)’s method after several refinement steps (see Section 3.3).

We choose to build USMT with an alternative method for phrase table induction. We adopt the method proposed by Marie and Fujita (2018), except that we remove the supervision using a bilingual word lexicon. First, phrases are collected using the following equation (Mikolov et al., 2013):

$$score(w_i w_j) = \frac{freq(w_i w_j) - \delta}{freq(w_i) \times freq(w_j)}, \quad (1)$$

where  $w_i$  and  $w_j$  are two consecutive tokens or phrases in the monolingual data,  $freq(\cdot)$  the frequency of the given token or phrase, and  $\delta$  a discounting coefficient for preventing the retrieval of phrases composed of very infrequent tokens. Consecutive tokens/phrases having a higher score than a pre-defined threshold are regarded as new phrases,<sup>4</sup> and a new pass is performed to obtain longer phrases. The iteration results in the collection of much longer and meaningful phrases, i.e., not only very frequent sequences of grammatical words, rather than only short n-grams. In our experiments, we perform 6 iterations to collect phrases of up to 6 tokens.<sup>5</sup> Equation (1) was

<sup>4</sup>This transformation is performed by simply replacing the space between the two tokens/phrases with an underscore.

<sup>5</sup>We chose a maximum phrase length of 6, since this value

originally proposed to identify non-compositional phrases. However, we choose to enforce the collection of more compositional phrases with a low  $\delta^6$  for the following reasons:

- very few phrases are actually non-compositional in standard SMT systems (Zens et al., 2012),
- most of them are not very frequent, and
- useful representation of compositional phrases can easily be obtained compositionally (Zhao et al., 2015).

To obtain the pairs of source and target phrases that populate the induced phrase table, we used the Equation proposed by Lample et al. (2018b):<sup>7</sup>

$$p(t_j|s_i) = \frac{\exp(\beta \cos(\text{emb}(t_j), \text{emb}(s_i)))}{\sum_k \exp(\beta \cos(\text{emb}(t_k), \text{emb}(s_i)))}, \quad (2)$$

where  $t_j$  is the  $j$ -th phrase in the target phrase list and  $s_i$  the  $i$ -th phrase in the source phrase list,  $\beta$  a parameter to tune the peakiness of the distribution<sup>8</sup> (Smith et al., 2017), and  $\text{emb}(\cdot)$  a function returning the bilingual embedding of a given phrase.

In this work, for a reasonably fast computation, we retained only the 300k most frequent phrases in each language and retained for each of them the 300-best target phrases according to Equation (2).

Standard phrase-based SMT uses the following four translation probabilities for each phrase pair.

- (a)  $p(t_j|s_i)$ : forward phrase translation probability
- (b)  $p(s_i|t_j)$ : backward phrase translation probability
- (c)  $\text{lex}(t_j|s_i)$ : forward lexical translation probability
- (d)  $\text{lex}(s_i|t_j)$ : backward lexical translation probability

is usually used as the maximum length in most state-of-the-art SMT frameworks.

<sup>6</sup>We set  $\delta = 10$  in all our experiments.

<sup>7</sup>We could not obtain results similar to the results reported in Lample et al. (2018b) (the second version of their arXiv paper) by using their Equation (3) with  $\beta = 30$  as they proposed. We have confirmed through personal communications with the authors that Equation (2), as we wrote, with  $\beta = 30$ , generates the expected results. We did not use the Equation computing  $\phi$  in Artetxe et al. (2018b), since it produces negative value as a probability when cosine similarity is negative.

<sup>8</sup>We set  $\beta = 30$  since it is the default value proposed in the code released by Smith et al. (2017): [https://github.com/Babylonpartners/fastText\\_multilingual](https://github.com/Babylonpartners/fastText_multilingual)

These probabilities, except (a), need to be computed only for the 300-best target phrases for each source phrase that are already determined using (a). (b) is given by switching  $s_i$  and  $t_j$  in Equation (2). To compute lexical translation probabilities, (c) and (d), given the significant filtering of candidate target phrases, we can adopt a more costly but better similarity score. In this work, we compute them using word embeddings as proposed by Song and Roth (2015):

$$\text{lex}(t_j|s_i) = \frac{1}{L} \prod_{l=1}^L \max_{k=1}^K p(t_j^k|s_i^l) \quad (3)$$

where  $K$  and  $L$  are the number of words in  $t_j$  and  $s_i$ , respectively, and  $p(t_j^k|s_i^l)$  the translation probability of the  $k$ -th target word  $t_j^k$  of  $t_j$  given the  $l$ -th source word  $s_i^l$  of  $s_i$  given by Equation (2). This phrase-level lexical translation probability is computed for both translation directions. Note that, unlike Song and Roth (2015) and Kajiwara and Komachi (2016), we do not use a threshold value under which  $p(t_j^k|s_i^l)$  is ignored, since it would require some supervised fine-tuning to be set according to the translation task. In practice, even without this threshold value, our preliminary experiments showed significant improvements of translation quality by incorporating  $\text{lex}(t_j|s_i)$  and  $\text{lex}(s_i|t_j)$  into the induced phrase table.

After the computation of the above four scores for each phrase pair in the induced phrase table, the phrase table is plugged in an SMT system to perform what we denote in the remainder of this paper as iteration 0 of USMT.

Computing lexicalized reordering models for the phrase pairs in the induced phrase table from monolingual data is feasible and helpful as shown by Klementiev et al. (2012). However, for the sake of simplicity, we do not compute these lexical reordering models for iteration 0.

### 3.2 Discussion about unsupervised tuning

State-of-the-art supervised SMT performs the weighted log-linear combination of different models (Och and Ney, 2002). The model weights are tuned given a small development set of bilingual sentence pairs. For completely unsupervised SMT, we cannot assume the availability of this development set. In other words, model weights must be tuned without the supervision of manually produced bilingual data.

Lample et al. (2018b) used some pre-existing default weights that work reasonably well. On the other hand, Artetxe et al. (2018b) obtained better results by using 10k monolingual sentences paired with their back-translations as a development set. Nonetheless, to create this development set, they also relied on the same pre-existing default weights used by Lample et al. (2018b). To be precise, both used the default weights of the Moses framework (Koehn et al., 2007). In this preliminary work, we present results with supervised tuning and with the Moses’s default weights.

However, regarding the use of default weights as “unsupervised tuning” is arguable, since these default weights have been determined manually to work well for European languages. For translation between much more distant languages,<sup>9</sup> these default weights would likely result in a very poor translation quality. We argue that unsupervised tuning remains one of the main issues in current approaches for USMT.

Note that while creating large training bilingual data manually for a particular language pairs is very costly, which is one of the fundamental motivations of unsupervised MT, we can assume that a small set of sentence pairs required for tuning can be created at a reasonable cost.

### 3.3 Refinement without back-translation

Artetxe et al. (2018b) and Lample et al. (2018b) presented the same idea of performing so-called refinement steps. Those steps use USMT to generate synthetic parallel data to train a new phrase table, with refined translation probabilities. This can be repeated for several iterations to improve USMT. The initial system at iteration 0 uses the induced phrase table (see Section 3.1), while the following iterations use only a phrase table and a lexicalized reordering model trained on the synthetic parallel data generated by USMT. They both fixed the number of iterations.

Artetxe et al. (2018b) and Lample et al. (2018b) generated the synthetic parallel data through back-translation: a target-to-source USMT system was used to back-translate sentences in the target language, then the pairs of each sentence in the target language and its USMT output in the source language were used as synthetic parallel data to train a new source-to-target USMT system. This way

of using back-translation has originally been proposed to improve NMT systems (Sennrich et al., 2016a) with a specific motivation to enhance the decoder by exploiting fluent sentences in the target language. In contrast, however, using back-translation for USMT lacks motivation. Since the source side of the synthetic parallel data, i.e., decoded results of USMT, is not fluent, USMT will learn a phrase table with many ungrammatical source phrases, or foreign words, that will never be seen in the source language, meaning that many phrase pairs in the phrase table will never be used. Moreover, possible and frequent source phrases, or even source words, may not be generated via back-translation and will be consequently absent from the trained phrase table.

We rather consider that the language model already trained on a large monolingual corpus in the target language can play a much more important role in generating more fluent translations. This motivates us to perform the refinement steps on synthetic parallel data made of source sentences translated into the target language by the source-to-target system, i.e., “forward translation,” as opposed to back-translation. In fact, the idea of re-training an SMT system on synthetic parallel data generated by a source-to-target system has already been proven beneficial (Ueffing et al., 2007).

At each iteration, we randomly sample new  $N$  source sentences from the monolingual corpus and translate them with the latest USMT system to generate synthetic parallel data.

### 3.4 Phrase table pruning

Generating synthetic parallel data through decoding millions of sentences is one of the most computationally expensive parts of the refinement steps, requiring also a large memory to store the whole phrase table.<sup>10</sup> In SMT, decoding speed can be improved by reducing the size of the phrase table. The phrase tables trained during the re-

<sup>9</sup>For instance, Lample et al. (2018b) presented for Urdu–English only the results with supervised tuning.

<sup>10</sup>To decode a particular test set, usually consisting of thousands of sentences, the phrase table can be drastically filtered by keeping only the phrase pairs applicable to the source sentences to translate. For the refinement steps of USMT, this filtering is impractical since we need to translate a very large number of sentences. In other words, it would still remain a large number of phrase pairs. Another alternative is to binarize the phrase table so that the system can load only applicable phrase pairs on-demand at decoding time. However, we did not consider it in our framework since the binarization is itself very costly to perform, and more importantly, the phrase table of each refinement step is used only once.

finement steps are expected to be very noisy and very large since they are trained on noisy parallel data. Therefore, we assume that a large number of phrase pairs can be removed without sacrificing translation quality. On this assumption, we use the well-known algorithm for pruning phrase table (Johnson et al., 2007), which has shown good performance in removing less reliable phrase pairs without any significant drop of the translation quality. This pruning can be done for each refinement step to reduce the phrase table size, and consequently to speed up the decoding. Note that we cannot prune the induced phrase table used at iteration 0, since it was not learned from parallel data: we do not have co-occurrence statistics for the phrase pairs.

#### 4 UNMT as NMT trained exclusively on synthetic parallel data

To make NMT able to learn how to translate from monolingual data only, previous work on UNMT (Artetxe et al., 2018c; Lample et al., 2018a,b; Yang et al., 2018) proposed dedicated architectures, such as denoising autoencoders, shared latent representations, weight sharing, pre-trained sub-word embeddings, and adversarial training.

In this paper, we propose to train UNMT systems exclusively on synthetic parallel data, using existing frameworks for supervised NMT. Specifically, we train the first UNMT system on synthetic parallel data generated by USMT through back-translating monolingual sentences in the target language, expecting that they are of a better quality than those generated by existing UNMT frameworks.

Our approach is significantly different from Lample et al. (2018b)’s “PBSMT+NMT” configuration in the following two aspects. First, while it uses synthetic parallel data generated by USMT only to further tune their UNMT system, ours uses it for initialization. Second, they assumed certain level of relatedness between source and target languages, which is a prerequisite to jointly pre-train bilingual sub-word embeddings. Our approach does not make this assumption.

However, training an NMT system only on synthetic parallel data generated by USMT, as we proposed, will hardly make an UNMT system significantly better than USMT systems. To obtain better UNMT systems, we propose the following (see also Figure 3).

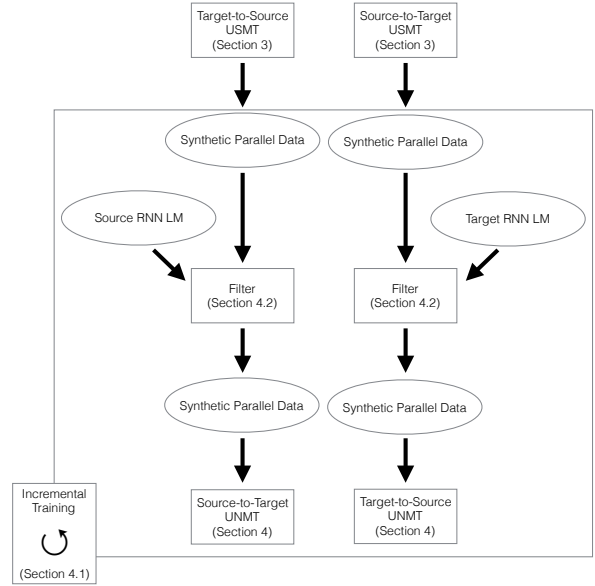


Figure 3: Our UNMT framework.

- Section 4.1: we propose an incremental training strategy for UNMT that gradually increases the quality and the quantity of synthetic parallel data.
- Section 4.2: we propose to filter the synthetic parallel data to remove before training the sentence pairs with the noisiest synthetic sentences, aiming at speeding up training and improving translation quality.

##### 4.1 Incremental training

To train UNMT, we first use the synthetic parallel data generated by the last refinement step of our USMT system. Since it has been shown that back-translated monolingual data significantly improves translation quality in NMT, as opposed to the refinement of our USMT (see Section 3.3), we train source-to-target and target-to-source UNMT systems on synthetic parallel data respectively generated by a target-to-source and source-to-target USMT systems.

In contrast to supervised NMT where synthetic parallel data are used in combination with human-made parallel data, we can presumably use as much synthetic parallel data as possible, since seeing more and more fluent target sentences will be helpful to train a better decoder while we can assume that the quality of synthetic source side remains constant. In practice, generating a large quantity of synthetic parallel data is costly. Therefore, to train the first UNMT system, we use the



same number,  $N$ , of synthetic sentence pairs generated by the final USMT system.

Since the source side of the synthetic parallel data is generated by USMT, it is expected to be of worse quality than those that state-of-the-art supervised NMT can generate. Therefore, we propose to refine UNMT through gradually increasing the quality and quantity of synthetic parallel data. First, we back-translate a new set of  $N$  monolingual sentences using our UNMT systems at iteration 1 in order to generate new synthetic parallel data. Then, new UNMT systems at iteration 2 are trained from scratch on the  $2N$  synthetic sentence pairs consisting of the new  $N$  synthetic data and  $N$  synthetic data generated by USMT. Note that we do not re-back-translate the monolingual data used at iteration 1 but keep them as they are for iteration 2 to reduce the computational cost. Similarly to the refinement steps of USMT, we can again perform this back-translation/re-training step for a pre-defined number of iterations to keep improving the quality of the source side of the synthetic data while increasing the number of new target sentences. At each iteration  $i$ ,  $(N \times i)$  synthetic sentence pairs are used for training.

This can be seen as an extension of Hoang et al. (2018)’s work, which performs a so-called iterative back-translation to improve NMT. The difference is that we introduce better synthetic parallel data, with new target sentences, at each iteration.

#### 4.2 Filtering of synthetic parallel data

Our UNMT system is trained on purely synthetic parallel data in which a large proportion of source sentences may be very noisy. We assume that removing the sentence pairs with the noisiest source sentences will improve translation quality. Inevitably it also reduces the training time.

Each sentence pair in the synthetic parallel data is evaluated by the following normalized source language model score:

$$ppl(S) = \frac{lm(S)}{len(S) + 1} \quad (4)$$

where  $S$  is a (synthetic) source sentence,  $lm(\cdot)$  the language model score, and  $len(\cdot)$  a function returning the number of tokens in the sentence. We add 1 to the number of tokens to account for the special token used by NMT that marks the end of a sentence. This scoring function has a negligible computational cost, but has shown satisfying performances in our preliminary experiments. While

we do not limit the language model to be specific type, in our experiment, we use a recurrent neural network (RNN) language model trained on the entire source monolingual data.

There are many ways to make use of the above score during NMT training. For instance, weighting the sentence pairs with this score during training is a possible alternative, and this idea is close to one used by Cheng et al. (2017) in their joint training framework for NMT. However, given that many of the source sentences would be noisy, we rather choose to discard potentially noisy pairs for training. It would also remove potentially useful target sentences, but we assume that the impact of this removal could be compensated at the succeeding iterations of UNMT, where we incrementally introduce new target sentences.

At each iteration  $i$  of incremental training, we keep only the cleanest  $(\alpha \times N \times i)$  synthetic sentence pairs<sup>11</sup> selected according to the score computed by Equation (4), where  $\alpha$  ( $0 < \alpha \leq 1$ ) is the filtering ratio.<sup>12</sup> This aggressive filtering will speed up training while relying only on the most fluent sentence pairs.

## 5 Experiments

In this section, we present experiments for evaluating our USMT and UNMT systems.

### 5.1 Experimental settings

For these preliminary experiments, we chose the language pair English–German (en-de) and the evaluation task WMT16 (newstest2016) for both translation directions, following previous work (Artetxe et al., 2018b; Lample et al., 2018b). To train our USMT and UNMT, we used only monolingual data: English and German News Crawl corpora respectively containing around 238M and 237M sentences.<sup>13</sup> All our data were tokenized and truecased with Moses’s tokenizer<sup>14</sup> and truecaser, respectively. The statistics for truecasing were learned from 10M sentences randomly sampled from the monolingual data.

<sup>11</sup>We considered both the sentence pairs used to initialize UNMT and all the sentence pairs generated by each iteration of UNMT in the set of sentence pairs to filter.

<sup>12</sup>We used  $\alpha = 0.5$  in our experiments.

<sup>13</sup><http://www.statmt.org/wmt18/translation-task.html>

<sup>14</sup>We escaped special characters but did not use the option for “aggressive” tokenization.

For the phrase table induction, the source and target word embeddings were learned from the entire monolingual data with the default parameters of `fasttext` (Bojanowski et al., 2017),<sup>15</sup> except that we set to 200 the number of dimensions.<sup>16</sup> For a reasonably fast computation, we retained only the embeddings for the 300k most frequent words. Word embeddings for two languages were then aligned in the same space using the `-unsupervised` option of `vecmap`.<sup>17</sup> From the entire monolingual data, we also collected phrases of up to 6 tokens in each language using `word2phrase`.<sup>18</sup> To maintain the experiments feasible and to make sure that we have a word embedding for all of the constituent words, we retained only 300k most frequent phrases made of words among the 300k most frequent words. We conserved the 300-best target phrases for each source phrase, according to Equation (2), consequently resulting in the initial phrase table for USMT containing 90M ( $300k \times 300$ ) phrase pairs.

We used Moses and its default parameters to conduct experiments for USMT. The language models used by our USMT systems were 4-gram models trained with `LMPLZ` (Heafield et al., 2013) on the entire monolingual data. In each refinement step, we trained a phrase table and a lexicalized reordering model on synthetic parallel data using `mgiza`.<sup>19</sup> We compared USMT systems with and without supervised tuning. For supervised tuning, we used `kb-mira` (Cherry and Foster, 2012) and the WMT15 newstest (newstest2015). For the configurations without tuning, we used Moses’s default weights as in previous work.

For UNMT, we used the Transformer (Vaswani et al., 2017) model implemented in Marian (Junczys-Dowmunt et al., 2018)<sup>20</sup> with the hyperparameters proposed by Vaswani et al. (2017).<sup>21</sup>

We reduced the vocabulary size by using byte-pair-encoding (BPE) with 8k symbols jointly learned for English and German from 10M sentences sampled from the monolingual data. BPE was then applied to the entire source and target monolingual data.<sup>22</sup> We used the same BPE vocabulary throughout our UNMT experiments.<sup>23</sup> We validated our model during UNMT training as proposed by Lample et al. (2018b): we did a supervised validation using 100 human-made sentence pairs randomly extracted from newstest2015. We consistently used the same validation set throughout our UNMT experiments. To filter the synthetic parallel sentences (see Section 4.2), we used an RNN language model trained on the entire monolingual data, without BPE, with a vocabulary size of 100k.<sup>24</sup>

For each of USMT and UNMT, we performed 4 refinement iterations. USMT has one more system in the beginning, which exploits an induced phrase table. At each iteration, we sampled new 3M monolingual sentences: i.e.,  $N = 3000000$ .<sup>25</sup>

For reference, we also trained supervised NMT with Marian on 5.6M, 2.8M, and 1.4M human-made parallel sentences provided by the WMT18 conference for the German–English news translation task.<sup>26</sup>

We evaluated our systems with detokenized and detruccased BLEU-cased (Papineni et al., 2002). Note that our results should not be directly compared with the tokenized BLEU scores reported in Artetxe et al. (2018b) and Lample et al. (2018b).

## 5.2 Results

Our results for USMT and UNMT are presented in Table 1.

We can first observe that supervised tuning for USMT improves translation quality, with 2.0 BLEU points of improvements, for instance between systems #5 and #6. Another interesting observation is that this improvement is carried on until the final iteration of UNMT (#11 and #12). These results show the importance of development data for tuning that could be created at a reason-

<sup>15</sup><https://fasttext.cc/>

<sup>16</sup>While Artetxe et al. (2018b) and Lample et al. (2018b) used 300 and 512 dimensions, respectively, we chose a smaller number of dimensions for faster computation, even though this might lead to lower quality.

<sup>17</sup><https://github.com/artetxem/vecmap>

<sup>18</sup><https://code.google.com/archive/p/word2vec/>

<sup>19</sup>`fast_align` (Dyer et al., 2013) is a significantly faster alternative for a similar performance on en-de (Durrani et al., 2014). We used `mgiza` since it is integrated in Moses.

<sup>20</sup><https://marian-nmt.github.io/>, version 1.6.

<sup>21</sup>Considering the computational cost of our approach for UNMT, we did not experiment with the “big” version of the Transformer model while it would probably have resulted in a better translation quality.

<sup>22</sup>We did not use BPE for USMT.

<sup>23</sup>Re-training BPE at each iteration of UNMT on synthetic data did not improve the translation quality in our preliminary experiments.

<sup>24</sup>We used also Marian to train the RNN language models.

<sup>25</sup>Artetxe et al. (2018b) and Lample et al. (2018b) respectively sampled 2M and 5M monolingual sentences.

<sup>26</sup>We did not use the ParaCrawl corpus.



System	USMT Tuning	de→en	en→de	#
Lample et al. (2018b) USMT	No	22.7*	17.8*	1
Artetxe et al. (2018b) USMT	back-translation	23.1*	18.2*	2
USMT (this work) w/ back-translation	Supervised	20.5	17.0	3
	No	19.5	15.0	4
USMT (this work)	Supervised	22.1	17.4	5
	No	20.2	15.5	6
Lample et al. (2018b) UNMT	No	21.0*	17.2*	7
Lample et al. (2018b) USMT+UNMT	No	25.2*	20.2*	8
UNMT (this work) w/o filtering	Supervised	28.2	21.3	9
	No	27.0	19.6	10
UNMT (this work)	Supervised	28.8	21.6	11
	No	26.7	20.0	12
Supervised NMT (1.4M sent. pairs)	Supervised	32.5	29.9	13
Supervised NMT (2.8M sent. pairs)	Supervised	33.8	31.6	14
Supervised NMT (5.6M sent. pairs)	Supervised	34.9	32.3	15

Table 1: Results of our USMT and UNMT systems (denoted “this work”) evaluated with BLEU for the WMT16 German–English news translation task. We present results for USMT with back-translation (#3 and #4) and forward translation (#5 and #6) during the refinement steps. Results for UNMT are presented without (#9 and #10) and with (#11 and #12) filtering of synthetic parallel data. “\*” indicates the scores shown in the original paper for indicative purpose only, since they are tokenized BLEU scores and thus not directly comparable with our results.

able cost (see Section 3.2).

Our USMT systems benefited more from forward translation (#5 and #6) than back-translation (#3 and #4) during the refinement steps, with an improvement of 1.6 and 0.4 BLEU points for de→en and en→de (with supervised tuning), respectively. Pruning the phrase table (see Section 3.4) did not hurt translation quality but removed around 93% of the phrase pairs in the phrase tables for each refinement step. Nonetheless, our USMT systems seem to significantly underperform the state-of-the-art USMT proposed by Lample et al. (2018b) (#1) and Artetxe et al. (2018b) (#2). This is potentially the consequence of the following: we used much lower dimensions for our word embeddings and much less phrases (300k source and target phrases), than in Artetxe et al. (2018b) (1M source and target phrases). In our future work, we will investigate whether their parameters improve the performance of our USMT systems.

While our USMT systems do not seem to outperform previous work, we can observe that the synthetic parallel data that they generated are of

sufficient quality to initialize our UNMT. Incremental training improved significantly translation quality. To the best of our knowledge, we report the best results of unsupervised MT for this task which is, for de→en, only 3.7 BLEU points lower (#11) than a supervised NMT system trained on 1.4M parallel sentences (#13).<sup>27</sup> Our best UNMT systems (#11 and #12) significantly outperformed our USMT systems (#5 and #6) by more than 6.0 BLEU points, for de→en. Filtering the synthetic parallel sentences at each iteration significantly improved the training speed<sup>28</sup> for a comparable or better translation quality for both translation directions. The results confirm the importance of filtering the very noisy synthetic source sentences generated by back-translation.

<sup>27</sup>A fair supervised NMT baseline should also use, in addition to human-made parallel sentences, back-translated data for training.

<sup>28</sup>For instance, for the last iteration of UNMT for de→en, the training using 4 GPUs consumed 30 hours with filtering while it took 52 hours without filtering.

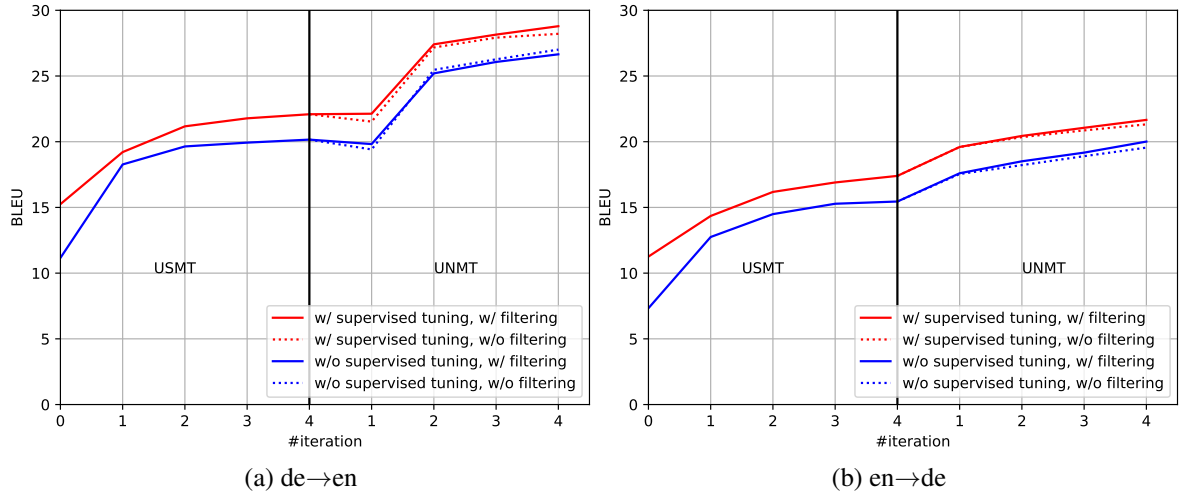


Figure 4: Learning curves of our USMT (#5 and #6) and UNMT (#9, #10, #11, and #12) systems presented in Section 5.

### 5.3 Learning curves

In this section, we present the evolution of the translation quality during training of USMT and UNMT.

The learning curves of our systems, for the same experiments presented in Section 5.1, are given in Figures 4a and 4b for  $de \rightarrow en$  and  $en \rightarrow de$ , respectively. Iteration 0 of our USMT, using an induced phrase table, performed very poorly; for instance systems without supervised tuning (leftmost points of blue lines) achieved only 11.2 and 7.3 absolute BLEU points for  $de \rightarrow en$  and  $en \rightarrow de$ , respectively. Iterations 1 and 2 of USMT were very effective and covered most of the improvements between iteration 0 and iteration 4. After 4 iterations, we observed improvements of 9.0 and 8.1 BLEU points for  $de \rightarrow en$  and  $en \rightarrow de$ , respectively.

The learning curves of UNMT were very different for the two translation directions. The first iteration of UNMT, trained on the synthetic parallel data generated by USMT, performed slightly lower than USMT for  $de \rightarrow en$  while for  $en \rightarrow de$  we observed around 2.0 BLEU points of improvements. This confirms the ability of NMT in generating significantly better sentences than SMT for morphologically-rich target languages (Bentivogli et al., 2016). Then, the second iteration of UNMT improved the translation quality significantly for  $de \rightarrow en$ , but much more moderately for  $en \rightarrow de$ . For instance, in the configuration without supervised tuning and with language model filtering (blue solid lines), we observed 5.4 and 0.9 BLEU

points of improvements for  $de \rightarrow en$  and  $en \rightarrow de$ , respectively. Succeeding iterations continued to improve translation quality but more moderately.

For both translation directions, the learning curves highlighted that improving the synthetic parallel data generated by USMT, and used to initialize UNMT, is critical to improve UNMT: synthetic parallel data generated with tuned USMT were consistently more useful for UNMT than the synthetic parallel data of lower quality generated by USMT without tuning.

## 6 Conclusion and future work

We proposed a new approach for UNMT that can be straightforwardly exploited with well-established architectures and frameworks used for supervised NMT without any modifications. It only assumes for initialization the availability of synthetic parallel data that can be, for instance, easily generated by USMT. We showed that improving the quality of the synthetic parallel data used for initialization is crucial to improve UNMT. We obtained with our approach a new state-of-the-art performance for unsupervised MT on the WMT16 German–English news translation task.

For future work, we will extend our experiments to cover many more language pairs, including distant language pairs for which we expect that our approach will perform better than previous work that assumes the relatedness between source and target languages. We will also analyze the impact of using synthetic parallel data of a much better quality to initialize UNMT. Moreover, we

would like to investigate the use of much noisier and not comparable source and target monolingual corpora to train USMT and UNMT, since we consider it as a more realistic scenario when dealing with truly low-resource languages. We will also study our approach in the semi-supervised scenario where we assume the availability of some human-made bilingual sentence pairs for training.

## References

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. *CoRR*, abs/1809.01272.
- Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018c. Unsupervised neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations*.
- Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, USA. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Proceedings of the 26th International Joint Conferences on Artificial Intelligence*, pages 3974–3980.
- Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.
- Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. 2014. Edinburgh’s phrase-based machine translation systems for WMT-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104. Association for Computational Linguistics.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, USA. Association for Computational Linguistics.
- Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696. Association for Computational Linguistics.
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.
- Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt, 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. Association for Computational Linguistics.
- Tomoyuki Kajiwara and Mamoru Komachi. 2016. Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158. The COLING 2016 Organizing Committee.
- Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. 2012. Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 130–140, Avignon, France. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.
- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *Proceedings of the 6th International Conference on Learning Representations*.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.
- Benjamin Marie and Atsushi Fujita. 2018. Phrase table induction using monolingual data for low-resource statistical machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17(3):16:1–16:25.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119, Lake Tahoe, USA. Curran Associates Inc.
- Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, USA. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.
- Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for*

*Computational Linguistics (Volume 1: Long Papers)*, pages 778–788. Association for Computational Linguistics.

Yangqiu Song and Dan Roth. 2015. Unsupervised sparse vector densification for short text similarity. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1275–1280. Association for Computational Linguistics.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55. Association for Computational Linguistics.

Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 972–983. Association for Computational Linguistics.

Kai Zhao, Hany Hassan, and Michael Auli. 2015. Learning translation models from monolingual continuous representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1527–1536. Association for Computational Linguistics.