Empirical Regularization for Synthetic Sentence Pairs in Unsupervised Neural Machine Translation

Xi Ai, Bin Fang
College of Computer Science,Chongqing University
barid.x.ai@gmail.com, fb@cqu.edu.cn

Abstract

UNMT tackles translation on monolingual corpora in two required languages. Since there is no explicitly cross-lingual signal, pre-training and synthetic sentence pairs are significant to the success of UNMT. In this work, we empirically study the core training procedure of UNMT to analyze the synthetic sentence pairs obtained from back-translation. We introduce new losses to UNMT to regularize the synthetic sentence pairs by training the UNMT objective and the regularization objective jointly. Our comprehensive experiments support that our method can generally improve the performance of currently successful models on three similar pairs $\{French, German, Romanian\} \leftrightarrow English$ and one dissimilar pair $Russian \leftrightarrow English$ with acceptably additional cost.

Introduction

UNMT (unsupervised neural machine translation) leverages language modeling, e.g., denoising language modeling (Hill, Cho, and Korhonen 2016; Dai and Le 2015; Lample et al. 2018a; Artetxe et al. 2018), to both the two required languages, learning to reconstruct sentences in the two languages. The idea is, language knowledge can facilitate UNMT to decompose representation for required languages, and such language knowledge can be transferred and eventually help UNMT to translate fluently due to shared layers/weights. Given the nature of translation, although shared layers/weights are employed to work as a pivot language, some weak cross-lingual signals are expected at the very least. Therefore, to train UNMT for a true translation task without violating the constraint of using nothing but monolingual corpora, back-translation (Sennrich, Haddow, and Birch 2016a) is jointly used in training. Significantly, this on-the-fly back-translation generates synthetic sentence pairs to provide synthetic supervision for training.

(Artetxe, Labaka, and Agirre 2016; Zhang et al. 2017; Artetxe, Labaka, and Agirre 2017, 2018; Lample et al. 2018b) first observe that the recently successful UBWE (unsupervised bilingual word embeddings) can provide UNMT required word-level cross-lingual knowledge in the initialization. On the other hand, the objective of BERT (Devlin et al. 2019) or MLM (masked language modeling) encourages a model to find multilingual properties (Wu and Dredze 2020; Karthikeyan et al. 2020; Pires, Schlinger, and Garrette 2019) by inputting multilingual corpora. Thus, XLM (Lample et al. 2018b), MASS (Song et al. 2019), BART (Lewis et al. 2020) and mBART (Liu et al. 2020) are proposed to adapt MLM for UNMT in pre-training and training, hence encouraging UNMT to build a robustly multilingual space upon shared layers/weights. The robustly multilingual space eventually and implicitly provides cross-lingual knowledge.

Although a large body of the previous study shows the significance of pre-training, we are aware that the quality of the synthetic sentence pairs is *not* guaranteed. Compared to NMT, which leverages the synthetic sentence pairs for further improvement through back-translation, the synthetic sentence pairs significantly provide cross-lingual knowledge to UNMT, facilitating training in a pseudo NMT scenario. Meanwhile, NMT generates the synthetic sentence pairs by typically reusing a trained translation model in finetuning, whereas UNMT generates the synthetic sentence pairs in a zero-shot (Johnson et al. 2017) style or a few-shot style (Brown et al. 2020), which only pre-trains the model on monolingual corpora at the most.

In this work, to guarantee the quality of the synthetic sentence pairs, we tackle the challenge with pure neural settings. Concretely, we present regularization models to regularize the synthetic sentence pairs. In this way, UNMT can be jointly trained with the new objective of regularization. Intuitively, the regularization should have three properties: *1) Low-cost :* it should be very simple to be implemented with a little additional cost in time because training UNMT is time-consuming; *2) Data free:* the model does not need additional data to regularize the synthetic sentence pairs; *3) Efficient decoding:* the method should not hurt the efficiency of decoding. To explore this idea, we have three main works:

- We present a method to regularize the shared semantics of a synthetic sentence pair, regardless of word semantics somewhat. This method adds a new loss to UNMT based on the high-level meaning of the sentence.
- We empirically study the regularization word-wise. Concretely, instead of regularizing the shared semantic between the two sentences from a synthetic sentence pair, we present a method to regularize similar/close words in

a synthetic sentence pair. This method does not eventually enable the model to learn word translation (Lample et al. 2018b; Artetxe, Labaka, and Agirre 2018) but adds a new objective into UNMT for joint training.

- We conduct comprehensive experiments to evaluate our methods in different configurations.

Note that, although there have been successful models (Lample et al. 2018c; Artetxe, Labaka, and Agirre 2019; Ren et al. 2019) employing phrase-based models, statistical models and their variants, in this paper, we only study pure neural models without any benefits from these models, e.g., SMT or PBSMT (Lample and Conneau 2019; Ren et al. 2019; Artetxe, Labaka, and Agirre 2019; Koehn, Och, and Marcu 2003). Our method is general and can be applied to any UNMT/NMT architecture, e.g., LSTM (Wu et al. 2016) and Transformer (Vaswani et al. 2017). Besides, we focus on the training phase instead of the pre-training phase. In the evaluation section, we conduct comprehensive experiments to show how our method performs on pre-trained models with different configurations and on random models.

Background and Related Work

NMT (neural machine translation) (Bahdanau, Cho, and Bengio 2015; Wu et al. 2016; Vaswani et al. 2017; Sutskever, Vinyals, and Le 2014) can be studied in an unsupervised learning manner. Concretely, UNMT models are based on the assumption that the two languages can be reconstructed from shared encodings (Lample et al. 2018a; Artetxe et al. 2018). In other words, the shared encoding works as a pivot language that is translated to the required language regardless of the input language. Typically, the recently successful UNMT models build upon denoising language modeling (Dai and Le 2015; Hill, Cho, and Korhonen 2016) for the two languages, respectively, with shared layers between the two languages (Artetxe et al. 2018; Lample et al. 2018a; Lample and Conneau 2019; Lample et al. 2018c; Sun et al. 2019; Yang et al. 2018; Liu et al. 2020; Lewis et al. 2020; Song et al. 2019), as:

$$\begin{aligned}\mathcal{L}_{lm}(X) = \mathbb{E}_{X\sim\phi L_1}[-logP(X|X';\theta_{Enc_{L_1}},\theta_{Dec_{L_1}})]\\ \mathcal{L}_{lm}(Y) = \mathbb{E}_{Y\sim\phi L_2}[-logP(Y|Y';\theta_{Enc_{L_2}},\theta_{Dec_{L_2}})]\end{aligned} \tag{1}$$

where X' and Y' are corrupted X and Y in language L_1 and language $L2$ respectively and $(\theta_{Enc_{L_1}} \cup \theta_{Dec_{L_1}}) \cap (\theta_{Enc_{L_2}} \cup \theta_{Dec_{L_2}}) \neq \phi$. Nevertheless, this idea only accounts one language without considering the translation between the two languages when training the objective of denoising language modeling only, i.e., the input and the output are in the same language. To facilitate translation training without violating the constraint of using nothing but monolingual corpora, on-the-fly back-translation (Sennrich, Haddow, and Birch 2016a) is used to generate synthetic sentence pairs. Concretely, given two input sentences (X, Y) in the two languages respectively, we obtain two synthetic sentence pairs $X \rightarrow \tilde{Y}$ and $Y \rightarrow \tilde{X}$ in inference mode. UNMT learns translation knowledge on both the language sides by simultaneously modeling $\tilde{Y} \rightarrow X$ and $\tilde{X} \rightarrow Y$ in the NMT scenario. Hence, we jointly optimize two translation losses for the two input sentences:

$$\begin{aligned}\mathcal{L}_{bt}(X,\tilde{Y}) = \mathbb{E}_{X\sim\phi L_1}[-logP(X|\tilde{Y};\theta_{Enc_{L_2}},\theta_{Dec_{L_1}})]\\ \mathcal{L}_{bt}(Y,\tilde{X}) = \mathbb{E}_{Y\sim\phi L_2}[-logP(Y|\tilde{X};\theta_{Enc_{L_1}},\theta_{Dec_{L_2}})]\end{aligned} \tag{2}$$

where $\{X, \tilde{Y}\}$ and $\{\tilde{X}, Y\}$ are synthetic sentence pairs. Thus, UNMT learns to jointly optimize the loss:

$$\begin{aligned}&\mathcal{L}_{UNMT} =\\ &\mathcal{L}_{lm}(X) + \mathcal{L}_{lm}(Y) + \mathcal{L}_{bt}(X,\tilde{Y}) + \mathcal{L}_{bt}(Y,\tilde{X})\end{aligned} \tag{3}$$

To improve the performance of UNMT, successful UNMT models (Liu et al. 2020; Lewis et al. 2020; Song et al. 2019; Lample and Conneau 2019; Lample et al. 2018c) pay attention to pre-train the encoder and the decoder, i.e., $\theta_{Enc_{L_1}}, \theta_{Dec_{L_1}}, \theta_{Enc_{L_2}}$ and $\theta_{Dec_{L_2}}$, in multilingual modeling settings, i.e., $\theta_{Enc_{L_1}} = \theta_{Enc_{L_2}}$ and $\theta_{Dec_{L_1}} = \theta_{Dec_{L_2}}$. The pre-trained encoder-decoder eventually facilitates UNMT training and improves the quality of translation. Meanwhile, pre-trained bilingual word embeddings that are learned in an unsupervised manner (Lample et al. 2018b; Artetxe, Labaka, and Agirre 2016, 2017, 2018), i.e., UBWE, can facilitate UNMT training (Lample et al. 2018a,c; Artetxe, Labaka, and Agirre 2019; Artetxe et al. 2018). In this scenario, all the lookup tables are initialized from pre-trained bilingual word embeddings.

Although there have been successful models (Lample et al. 2018c; Artetxe, Labaka, and Agirre 2019; Ren et al. 2019) employing phrase-based models, e.g., phrase-based statistical machine translation, to improve and guarantee the quality of the synthetic sentence pairs, we present neural models in this work. That is, given the loss Eq.3 of UNMT, we use a regularization model to regularize the synthetic sentence pairs. In this way, UNMT can be jointly trained with the new objective of regularization.

Besides, there has been a topic to search potentially aligned sentences (Grover and Mitra 2017; Munteanu, Fraser, and Marcu 2004; Hangya and Fraser 2020; Hangya et al. 2018) that can be indirectly leveraged for UNMT. The idea is more or less similar to using synthetic sentence pairs, but additional models are introduced so that the efficiency of UNMT training degrades significantly. Thus, it is not prevalent in the UNMT scenario.

Train with Regularization

Notation We use x and y to denote the word embedding/vector in language L_1 and language L_2, respectively. d_{model} is the model dimension, and d_{we} is the word embedding dimension. $X = (x_1, x_2, ..., x_n) \in R^{N\times d_{we}}$ and $Y = (y_1, y_2, ..., y_m) \in R^{M\times d_{we}}$ are the sentences sampled from corpora in language L_1 and language L_2 respectively, where N and M are the sequence length. The synthetic sentence $\tilde{X}$ and $\tilde{Y}$ are similar to X and Y. Besides, $\{X, \tilde{Y}\}$ and $\{\tilde{X}, Y\}$ denote synthetic sentence pairs. Voc denotes the last layer that outputs a probability over a vocabulary. *For notational simplicity, in most presentation of this paper, we use* $\{\tilde{X}, Y\}$ *as an example to discuss and present our* *idea. However, all the operations are simultaneously applied to both $\{X, \tilde{Y}\}$ and $\{\tilde{X}, Y\}$ in training.*

Framework

Given $\{\tilde{X}, Y\}$, we assume an implicit error $E_{synthetic}$ that indicates the semantic distance [1] between $\tilde{X}$ and Y as:

$$E_{synthetic} = \|\mathcal{F}(\tilde{X}) - \mathcal{F}(Y)\| \quad (4)$$

where latent $\mathcal{F}$ extracts high-level semantic features for distance measurement and $\{X, \tilde{Y}\}$ is similar to $\{\tilde{X}, Y\}$. We anticipate three main properties of $E_{synthetic}$: **1)** the value of $E_{synthetic}$ in the NMT scenario is smaller than in the UNMT scenario because NMT generates the synthetic sentence pairs by reusing the trained translation model in fine-tuning; **2)** training on $\{\tilde{X}, Y\}$ with small $E_{synthetic}$ can improve the performance of translation because $\tilde{X}$ and Y are aligned tightly; **3)** $\mathcal{F}$ should be a soft function [2] that does not degrade training efficiency significantly. We then define a regularization loss of UNMT $\mathcal{L}_{reg}$ as:

$$\mathcal{L}_{reg} = \mathcal{L}_{\mathcal{F}}(\tilde{X}, Y) + \mathcal{L}_{\mathcal{F}}(X, \tilde{Y}) \quad (5)$$

where $\mathcal{L}_{\mathcal{F}}$ is the loss of our regularization model implying the implicit error $E_{synthetic}$.

To optimize $\mathcal{L}_{reg}$, we propose to introduce $\mathcal{L}_{reg}$ to UNMT, adding the new loss $\mathcal{L}_{reg}$ into the loss of UNMT Eq.3 for joint optimizing:

$$\mathcal{L}_{UNMT} = \mathcal{L}_{lm}(X) + \mathcal{L}_{lm}(Y) + \mathcal{L}_{bt}(X, \tilde{Y}) + \mathcal{L}_{bt}(Y, \tilde{X}) + \lambda \mathcal{L}_{reg} \quad (6)$$

where λ is the weight for $\mathcal{L}_{reg}$. $\mathcal{L}_{reg}$ is minimized during joint training on monolingual corpora. Significantly, our method does not affect pre-training. In this work, we assume UNMT has been pre-trained completely or initialized randomly. We will experiment with both configurations in § Experiment and Empirical Study.

Sentence-wise Regularization

Preprocess We first present the sentence-wise regularization. Since both $\tilde{X}$ and Y are a sequence of vector, we aggregate all the vectors with position encodings to obtain a vector of sentence semantic, similar to that is leveraged in GNMT (generative NMT) (Shah and Barber 2018; Bowman et al. 2016). The procedure is formally described as:

$$\tilde{X}_s = \frac{1}{N}\sum_{i=0}^{N} FFN(\tilde{x_i} + P_i); Y_s = \frac{1}{M}\sum_{i=0}^{M} FFN(y_i + P_i) \quad (7)$$

where $\tilde{X}_s, Y_s \in R^{d_{we}}$, FFN is a two-layer feed-forward network (Vaswani et al. 2017) and P_i is a static position encoding (Vaswani et al. 2017). $\tilde{X}_s$ and Y_s are encouraged to model sentence semantics naively. For $\{X, \tilde{Y}\}$, we do similar preprocess.

Auto-encoder Regularization Significantly, $\tilde{X}_s$ and Y_s have to share some latent features because we expect a shared semantic between $\{\tilde{X}, Y\}$ for translation. Inspired by (Vincent 2010), we adapt denoising auto-encoder with a drop probability 0.1 for each element in $\tilde{X}_s$ and Y_s, obtaining bottleneck features for the regularization. Concretely, we employ a 3-layer denoising auto-encoder outputing bottleneck features of size $d_{we}/2$ as our auto-encoder regularization. The auto-encoder is simultaneously trained with UNMT. Then, we increase the similarity between the bottleneck features from $auto\text{-}encoder(\tilde{X}_s)$ and $auto\text{-}encoder(Y_s)$. Therefore, $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$ can be written as:

$$\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y) = 1 - \cos(BF(AE(\tilde{X}_s)), BF(AE(Y_s))) \quad (8)$$

where $AE(*)$ denotes the denoising auto-encoder and $BF(*)$ denotes the bottleneck features obtained from $AE(*)$. $\mathcal{L}_{\mathcal{F}}(X, \tilde{Y})$ is similar to $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$. Significantly, AE is multilingual, discussed in the following comparison.

BOW Regularization Formally, given the general process in Eq.7, we extend the idea significantly as:

$$\tilde{X}_s = \sigma\sum_{i=0}^{N} Voc(FFN(\tilde{x_i})); \; Y_s = \sigma\sum_{i=0}^{N} Voc(y_i) \quad (9)$$

where σ is a $sigmoid$ activation layer, Voc is the word generator (see §Notation) and $\tilde{X}_s, Y_s \in R^{vocabulary\_size}$. Specifically, we aggregate all the outputs of the word generator and then perform $sigmoid$ activation that outputs the BOW (bag-of-words) scores $\tilde{X}_s$ and Y_s (or sentence semantics) for $\tilde{X}$ and Y respectively, where $\tilde{X}$ is preprocessed by FFN position-wise. In other words, the index of $\tilde{X}_s$ or Y_s represents the index of a word in the lookup table, and the $sigmoid$ value of each element/index in $\tilde{X}_s$ or Y_s represents the probability of a word that appears in the sentence regardless of the position in the sentence. Intuitively, we expect the two BOW scores are as the same as possible, hence encouraging UNMT to minimize:

$$\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y) = \mathbb{E}_{\tilde{X} \sim \phi L_1}(-log P_{L_2 \to L_1}(\tilde{X}_s | Y_s)) \quad (10)$$

$\mathcal{L}_{\mathcal{F}}(X, \tilde{Y})$ is similar to $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$.

Comparison We have noticed some close ideas. **1)** For auto-encoder regularization, since we process both $\tilde{X}_s$ and Y_s to the same auto-encoder, the auto-encoder is trained as a simply multilingual encoder somewhat. Compared to the close idea BERT (Devlin et al. 2019) and its variants (Liu et al. 2020; Lewis et al. 2020; Song et al. 2019; Lample and Conneau 2019), which consider word semantics, our method encourages the auto-encoder to extract latent features of sentence semantic and makes latent features as similar as possible because a high-quality synthetic sentence pair has to share the same sentence semantic. **2)** For BOW regularization, previous works (Mikolov et al. 2013) study the

[1] If $\tilde{X}$ and Y are parallel or perfectly aligned, the semantic distance is 0, otherwise > 0.

[2] The hard function can be described as a translation model that translates $\tilde{X}$ and Y to a pivot language.

Algorithm 1 Local Alignment

Input: $\{\tilde{Z}, Z\}$, $\tilde{Z} \in (\tilde{z}_0, ..., \tilde{z}_N)$, $Z \in (z_0, ..., z_M)$
INDEX = list
for $i = 0$ **to** N **do**
  $candidate = double\_cos(\tilde{z}_i, Z)$
  $c = get\_the\_index\_of\_the\_largest\_value(candidate)$
  INDEX.append(c)
end for
Output: $\{\tilde{Z}, Z[\text{INDEX}]\}$

word distribution for one language based on a BOW score, whereas we study the word distribution for two languages based on two BOW scores, preprocessing one of the two languages by FFN.

Word-wise Regularization

Preprocess Both UBWE pre-training and encoder-decoder pre-training can provide high-quality bilingual word embeddings for UNMT, especially at the beginning of UNMT training. Furthermore, (Sun et al. 2019; Lample et al. 2018c; Artetxe et al. 2018) study the correlation between the quality of bilingual word embeddings and the performance of UNMT, reporting the degradation of the quality of bilingual word embeddings during training. Therefore, they propose to update bilingual word embeddings periodically or, more aggressively, fix bilingual word embeddings in training, which regularizes the synthetic sentence pairs statically and globally. On the contrary, we regularize synthetic sentence pairs dynamically and locally. Specifically, given $\{\tilde{X}, Y\}$, $\tilde{x_k}$ has to be close to y_i in the space of bilingual word embedding, where i and k are positions of the corresponding words. Intuitively, we only need to regularize $\tilde{x_k}$ and y_i in order to regularize $\{\tilde{X}, Y\}$.

However, this idea faces a local alignment problem. Specifically, different languages do not perfectly share the same word order. Therefore, it is difficult to decide i and k for the word-wise regularization. For instance, given Y = (I, like, to, drink, coffee, in, the, morning.) and $\tilde{X}$ = (J'aime, boire, du, café, le, matin.), $(y_4 = coffee)$ is not parallel or close to $(\tilde{x}_4 = le)$ if we simply set $i = k = 4$. To solve this problem, we fix $\tilde{X}$ and reconstruct Y. Concretely, we run Algorithm 1, which is based on $double\_cos$ score (Lample et al. 2018b) [3], to search y_i for $\tilde{x_k}$, hence matching Y to $\tilde{X}$ at every position and reconstructing Y to the length of $\tilde{X}$. After this operation, x_i and y_i are potentially aligned, i.e., $i = k$ in our previous example. *Note that the original version of synthetic sentence pairs is still used for the UNMT objective without any change.*

Naive Regularization Intuitively, we can improve the quality of a synthetic sentence pair by maximizing the similarity between word embeddings from $\tilde{X}$ and corresponding word embeddings from Y. Formally, we aim to minimize the objective function:

$$\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y) = 1.0 - similarity(\tilde{X}, Y) \quad (11)$$

where $similarity(\tilde{X}, Y) = \frac{1}{N}\sum_{k=0}^{N} \cos(\tilde{x}_k, y_k)$ and N is the length of $\tilde{X}$. $\mathcal{L}_{\mathcal{F}}(X, \tilde{Y})$ is similar to $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$.

GAN Regularization Inspired by works of (Lample et al. 2018a; Sun et al. 2019; Mikolov, Le, and Sutskever 2013; Kim, Gao, and Ney 2019), which study the linear transformation between two languages, we introduce a transformation $W_{L_1toL_2}$ to synthetic sentence pairs, constructing a generative model G. And, we use a discriminator D to predict which language word embeddings belong. Concretely, we define $W_{L_1toL_2} \in R^{d_{we} \times d_{we}}$ that constructs a generative mode $G = W_{L_1toL_2}\tilde{x}$ for any word embedding in $\tilde{X}$. To learn G (or $W_{L_1toL_2}$) and D, we simply utilize GAN (Goodfellow et al. 2014) architecture, optimizing the objective as:

$$\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y) = \mathcal{L}_D + \mathcal{L}_G \quad (12)$$

where $\mathcal{L}_D(D|G) = \frac{1}{N}\sum_{k=0}^{N}(\mathbb{E}_{\tilde{x_k}}[-\log(1 - D(G(\tilde{x_k})))] + \mathbb{E}_{y_k}[-\log D(y_k)])$, $\mathcal{L}_G(G|D) = \frac{1}{N}\sum_{k=0}^{N}(\mathbb{E}_{\tilde{x_k}}[-\log D(G(\tilde{x_k}))] + \mathbb{E}_{y_k}[-\log(1 - D(y_k))])$ and N is the length of $\tilde{X}$. $\mathcal{L}_{\mathcal{F}}(X, \tilde{Y})$ is similar to $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$.

Trick In practice, given $\{\tilde{X}, Y\}$, we search k nearest neighbors to obtain the mean score as the input of $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$. In other words, we consider a word embedding and its k nearest neighbors in the space of bilingual word embedding. By this method, intuitively, we encourage the model to tolerate word choices in synthetic sentence pairs. We empirically set $k = 3$. $\mathcal{L}_{\mathcal{F}}(X, \tilde{Y})$ is similar to $\mathcal{L}_{\mathcal{F}}(\tilde{X}, Y)$.

Comparison The idea of the word-wise regularization is very close to word translation (Artetxe, Labaka, and Agirre 2016, 2017, 2018; Lample et al. 2018b) and its application, but we have two main differences. **1) Objective:** compared to word translation, which tries to minimize the distance between two word-embedding matrixes, the word-wise regularization pays attention to two possibly aligned word embeddings from a synthetic sentence pair. **2) Training:** word translation is trained on a synthetic vocabulary, or on a collection of selected words at the very least, which is formed from common words, e.g., numbers (Artetxe, Labaka, and Agirre 2017) or frequent words (Lample et al. 2018b), whereas our method does not need the synthetic vocabulary because bilingual word embeddings have been pre-trained in pre-training.

Experiment and Empirical Study

We adapt our methods for UNMT initialized from UBWE pre-training or encoder-decoder pre-training to show our method can generally improve the performance of UNMT regardless of the pre-training methods. For further evaluation, we also observe the performance of random UNMT and

[3] Readers can refer to (Lample et al. 2018b) for more details.

discuss some important aspects of our methods. Note that, there have been some successful statistics-based/phrase-based methods (Lample et al. 2018c; Artetxe, Labaka, and Agirre 2019) that are out of the scope of this work. We leave the adaptation with these technics for future work.

Dataset and Tokenization To be comparable, we train the model on the same dataset used in previous work (Liu et al. 2020; Lewis et al. 2020; Song et al. 2019; Lample and Conneau 2019; Lample et al. 2018c). Specifically, we first retrieve monolingual corpora $\{French, German, English, Russian\}$ from WMT 2018 [4] (Bojar et al. 2018) including all available $NewsCrawl$ datasets from 2007 through 2017 and monolingual corpora $Romanian$ from WMT 2016 [5] (Bojar et al. 2016) including $NewsCrawl$ 2016. We then train the model on *Similar* pairs: $\{French, German, Romanian\} \leftrightarrow English$ and one *Dissimilar* pair: $Russian \leftrightarrow English$. We report case-sensitive BLEU computed by *multi-BLEU.perl*[6] for $Fr \leftrightarrow En$ on *newstest2014* and $\{Ru, De, Ro\} \leftrightarrow En$ on *newstest2016*. Meanwhile, we use BPE (Sennrich, Haddow, and Birch 2016b) tokens, selecting the most frequent 60K tokens from concatenated corpora of language pairs by applying the same criteria in (Lample and Conneau 2019).

Training Setting We implement our experiments on Tensorflow 2.0 (Abadi et al. 2016) and will open our source code on GitHub. We set $\lambda = 1$ to obtain a balanced attention between the UNMT loss and the regularization loss in Eq.6. Adam optimizer (Kingma and Ba 2015) is used with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and a dynamic learning rate over the course of training (Vaswani et al. 2017) ($warmup\_steps = 5000$). We set dropout regularization with a drop rate $rate = 0.1$ and label smoothing with $gamma = 0.1$ (Mezzini 2018).

Reimplementation Principle To be fair, we reimplement some models on our machine with a smaller batch size. We compare the reimplemented results to the reported results on the same test set to ensure the difference less than 5% (or 1.5) in BLEU. Then, we can confirm the correctness.

Adaptation with UBWE Pre-training

UNMT Configuration The UNMT configuration is identical to the baseline model (Lample et al. 2018c). Specifically, UNMT has four layers in both the encoder and the decoder for each language, and three out of the four encoder and decoder layers are shared between the two languages. All the lookup tables are initialized from UBWE.

Pre-training Configuration Given the monolingual corpora, we independently train word embeddings on each language side by using fastText [7] (Bojanowski et al. 2017). We then use the public VecMap[8] (Artetxe, Labaka, and Agirre 2018) to map trained word embeddings to shared space, using the recommended configuration and setting $dim = d_{we}$.

Result Table 1 shows the performance of our methods on the $\{De, Fr, Ru\} \leftrightarrow En$ test sets. Based on the experiment with only monolingual corpora, we have three observations. **1)** Our method significantly outperforms the previous methods in all the language pairs. **2)** The naive regularization shows the weakest performance. Intuitively, the naive regularization just introduces a new loss to UNMT, whereas other objectives are jointly trained with UNMT, having more interaction with UNMT. On the other hand, compared to the naive regularization, which is parameter-free, other regularizations slightly increase the size of the parameter, but we do not observe any significant degradation of training efficiency. **3)** The word-wise regularization generally outperforms the sentence-wise regularization on similar pairs $\{Fr, De\} \leftrightarrow English$, but the sentence-wise regularization shows better performance on the dissimilar pair $Ru \leftrightarrow En$. We explain that the improvement gaining from the word-wise regularization is proportional to the performance of bilingual word embeddings. Generally, the performance of bilingual word embeddings is better on similar pairs than on dissimilar pairs (Lample et al. 2018b; Artetxe, Labaka, and Agirre 2018, 2017, 2016) so that the word-wise regularization shows better performance on similar pairs. Compared to that, the sentence-wise regularization gives UNMT a semantic prototype UNMT can get benefits from, not relying on the performance of bilingual word embeddings heavily.

Adaptation with Encoder-decoder Pre-training

UNMT Configuration The UNMT configuration is identical to XLM (Lample and Conneau 2019) that has a 6-layer encoder and a 6-layer decoder. All encoder layers, decoder layers, and lookup tables are shared by the two languages.

Pre-training Configuration We pre-train the encoder-decoder by reimplementing baseline models: XLM(Lample and Conneau 2019), MASS(Song et al. 2019) and mBART(Liu et al. 2020) that gain significant benefits from large mini-batches. Based on the official code[9], we reimplement these baseline models that only process $approx.4k$ tokens per mini-batch.

Result Table 2 shows that our method can generally improve the performance of baseline models. Meanwhile, we believe our method can also get benefits from larger mini-batches. We will leave it for future work.

Random Initialization

UNMT Configuration We use the same configuration of XLM in the **Encoder-decoder Pre-training** experiment.

[4]http://www.statmt.org/wmt18/translation-task.html

[5]http://www.statmt.org/wmt16/translation-task.html

[6]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-BLEU.perl

[7]https://github.com/facebookresearch/fastText

[8]https://github.com/artetxem/vecmap

[9]https://github.com/facebookresearch/XLM

Model	$De \to En$	$En \to De$	$Fr \to En$	$En \to Fr$	$Ru \to En$	$En \to Ru$
baseline (Lample et al. 2018c)	21.34	17.89	24.20	25.83	9.19	8.08
+AL (Yang et al. 2018)	22.23	18.11	25.50	27.97	9.38	8.22
+UBWE Agreement (Sun et al. 2019)	22.67	18.29	25.87	28.38		
+Naive	23.01	18.57	26.01	28.51	9.42	8.31
+Auto-encoder	23.46	18.91	26.57	29.55	10.11	9.04
+GAN	24.12	19.87	27.24	30.53	9.89	8.71
+BOW	23.87	19.22	26.96	30.17	10.42	9.35

Table 1: Performance of 4-layer transformer UNMT with UBWE pre-training (baseline). AL: adversarial learning. Agreement: static and global maintenance. The baseline model and the "baseline + AL" model are reimplemented.

Model	$De \to En$	$En \to De$	$Fr \to En$	$En \to Fr$	$Ro \to En$	$En \to Ro$
XLM (Lample et al. 2018c)	33.81	26.32	32.87	32.94	31.12	32.81
+ Naive	34.01	26.49	33.17	33.33	31.54	33.12
+ Auto-encoder	34.34	26.78	33.51	33.64	31.92	33.61
+ GAN	34.94	27.12	33.92	34.24	32.53	34.01
+ BOW	34.73	26.90	33.65	33.90	32.30	33.69
MASS (Song et al. 2019)	34.91	28.03	34.42	37.02	32.75	34.82
+ Naive	35.32	28.27	34.82	37.44	33.01	35.21
+ Auto-encoder	35.59	28.60	35.09	37.72	33.35	35.52
+ GAN	36.04	28.98	35.61	38.29	34.01	35.91
+ BOW	35.75	28.81	35.27	37.84	33.68	35.74
mBART (Liu et al. 2020)	33.65	29.37	32.75	34.12	30.01	34.54
+ Naive	33.87	29.61	32.90	34.63	30.32	34.81
+ Auto-encoder	34.49	30.19	33.11	35.03	30.21	35.00
+ GAN	34.94	30.82	33.56	35.55	30.94	35.46
+ BOW	34.71	30.47	33.41	35.19	30.60	35.14

Table 2: Performance of 6-layer transformer UNMT with encoder-decoder pre-training. All the baseline models are reimplemented by using smaller mini-batches.

Model	$De \to En$	$En \to De$
random (Lample et al. 2018c)	20.99	17.01
random + Naive, $\lambda = 1$	21.11	17.36
random + Auto-encoder, $\lambda = 1$	21.23	17.51
random + GAN, $\lambda = 1$	21.61	17.96
random + BOW, $\lambda = 1$	21.39	17.71
random + Naive, $annealing\ \lambda$	22.03	17.85
random + Auto-encoder, $annealing\ \lambda$	22.33	18.18
random + GAN, $annealing\ \lambda$	22.81	18.62
random + BOW, $annealing\ \lambda$	22.62	18.31

Table 3: Performance of 6-layer transformer UNMT with random initialization.

Pre-training Configuration All the parameters of UNMT including the lookup tables and the encoder-decoder are randomly initialized by Xavier initialization (Glorot and Bengio 2010) without pre-training.

Result Table 3 shows that our method can generally improve the performance of random UNMT even the improvement is marginal. We explain that random initialization does not provide reliable bilingual word embeddings for UNMT, and our methods are word-embedding-based methods[10] somewhat. Specifically, the regularization is trivial over the early training because bilingual word embeddings are randomly initialized, which results in random regularizing and aligning. To further understand this aspect, we anneal λ to weigh the new loss of regularization in Eq.6, linearly increasing λ from 0 to 1 over the first $200k$ iterations of training. Therefore, UNMT pays a little attention to the new loss over the early training when bilingual word embeddings have not been trained, and UNMT pays more attention to the new loss over the late training when bilingual word embeddings have been trained. In Table 3, the performance in the last 4 rows is better than the corresponding performance in the row $2 \sim 5$, which explicitly indicates this aspect. *Meanwhile, we are aware our method is only a complementary method of pre-training because pre-training can generally achieve better performance. However, our method and pre-training are perfectly compatible.*

Impact of Tokenization Method

Our methods are word-embedding-based methods, which is discussed in the **Random Initialization** experiment. We are

[10]Regardless of word-wise regularization and sentence-wise regularization, the input is word embeddings. See §5 Framework.

Configuration	Model	$De \rightarrow En$	$En \rightarrow De$
1)	BPE-ENDE baseline	33.81	26.32
	+ GAN	34.94	27.12
	+ BOW	34.73	26.90
	Word-ENDE baseline	33.04	25.67
	+ GAN	34.21	26.48
	+ BOW	33.83	26.21
2)	BPE-UBWE baseline	21.34	17.89
	+ GAN	24.12	19.87
	+ BOW	23.87	19.22
	Word-UBWE baseline	21.12	17.65
	+ GAN	23.94	19.62
	+ BOW	23.71	19.43

Table 4: Performance of UNMT. 1): 6-layer transformer UNMT with encoder-decoder pre-training. 2): 4-layer transformer UNMT with UBWE pre-training.

interested in how the tokenization method affects the performance of our method because there are potential problems when dealing with non-standard-word BPE tokens, e.g., non-standard-word tokens may not be aligned properly.

UNMT Configuration We use two configurations: *1)* the UNMT configuration is identical to the configuration of XLM in the **Encoder-decoder Pre-training** experiment; *2)* the UNMT configuration is identical to the configuration in the **UBWE Pre-training** experiment.

Pre-training Configuration Also, we have two configurations: *1) ENDE:* the encoder-decoder pre-training configuration is identical to the configuration of XLM in the **Encoder-decoder Pre-training** experiment; *2) UBWE:* the UBWE pre-training configuration is identical to the configuration in the **UBWE Pre-training** experiment. We use both *BPE-ENDE* and *BPE-UBWE* to denote the models on BPE vocabularies and both *Word-ENDE* and *Word-UBWE* to denote the models on word vocabularies. Meanwhile, to be comparable, the size of the word vocabulary is the same as the size of the BPE vocabulary.

Result Table 4 shows that our method is robust to different tokenization methods. Regardless of the marginal difference between the two baseline models in the same configuration, our method can generally improve the performance.

Effect of λ

We have conducted a λ-related experiment in the **Random Initialization** experiment. We further study the effect of λ in this experiment. Although we empirically set $\lambda = 1$ (Eq.6) to weigh the new loss in training, we further study how λ affects the UNMT performance.

UNMT Configuration & Pre-training Configuration All the configurations are identical to the configurations in the **UBWE Pre-training** experiment.

Model	λ	$De \rightarrow En$	$En \rightarrow De$
baseline	0	21.34	17.89
+ GAN	anneal from 0 to 1	23.89	19.71
+ GAN	1	24.12	19.87
+ GAN	0.01	22.12	18.05
+ GAN	0.1	22.95	18.31
+ GAN	0.5	23.87	19.66
+ GAN	2	23.51	19.85
+ GAN	5	22.30	18.87

Table 5: Effect of λ for UNMT with UBWE pre-training and 4-layer transformer.

Model	Token feeding/s	Degradation
baseline	1 ×	
+ Naive	0.99 ×	-1%
+ Auto-encoder	0.95 ×	-8%
+ GAN	0.92×	-8%
+ BOW	0.94 ×	-6%

Table 6: Training efficiency.

Result In Table 5, λ influences the performance of the regularization over the course of training. A large λ forces training to pay more attention to the regularization objective than to the UNMT objective. A small λ degrades the significance of the regularization. Although all the choices of λ generally improve the UNMT performance, a balance value of $\lambda = 1$ gains the best performance.

Training Efficiency

UNMT Configuration & Pre-training Configuration All the configurations are identical to the configurations in the **Random Initialization** experiment. We measure the performance of token feeding per second based on vanilla UNMT without any regularization.

Result To interact with UNMT training, some parameters are added to UNMT. However, we do not observe any significant degradation in the training efficiency. Within our settings, the training efficiency is only degraded by $1\% \sim 8\%$, presented in Table 6, that the additional cost is acceptable.

Conclusion

To further improve the performance of UNMT, we empirically study the core training procedure of UNMT that generates the synthetic sentence pairs. We assume that regularizing synthetic sentence pairs can improve the performance without any additional data or cross-lingual signal. Based on our assumption, we present four simple but effective regularization methods, and we observe significant improvement from our experiments, regardless of pre-training methods and tokenization methods. Meanwhile, our methods do not hurt the training efficiency severely. However, in the scenario of UNMT, compared to similar pairs, dissimilar pairs are still a challenge, which needs future work, and the regularization is only a complementary method of pre-training.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; and Zheng, X. 2016. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Artetxe, M.; Labaka, G.; and Agirre, E. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2289–2294. ISBN 9781945626258. doi:10.18653/v1/d16-1250.

Artetxe, M.; Labaka, G.; and Agirre, E. 2017. Learning bilingual word embeddings with (almost) no bilingual data. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1: 451–462. doi: 10.18653/v1/P17-1042.

Artetxe, M.; Labaka, G.; and Agirre, E. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, 789–798. ISBN 9781948087322. doi:10.18653/v1/p18-1073.

Artetxe, M.; Labaka, G.; and Agirre, E. 2019. An Effective Approach to Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 194–203. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1019.

Artetxe, M.; Labaka, G.; Agirre, E.; and Cho, K. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Bahdanau, D.; Cho, K. H.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5: 135–146. ISSN 2307-387X. doi:10.1162/tacl_a_00051.

Bojar, O. r.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huck, M.; Jimeno Yepes, A.; Koehn, P.; Logacheva, V.; Monz, C.; Negri, M.; Neveol, A.; Neves, M.; Popel, M.; Post, M.; Rubino, R.; Scarton, C.; Specia, L.; Turchi, M.; Verspoor, K.; and Zampieri, M. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, 131–198. Berlin, Germany: Association for Computational Linguistics.

Bojar, O. r.; Federmann, C.; Fishel, M.; Graham, Y.; Haddow, B.; Huck, M.; Koehn, P.; and Monz, C. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, 272–307. Belgium, Brussels: Association for Computational Linguistics.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Jozefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, 10–21. ISBN 9781945626197. doi:10.18653/v1/k16-1002.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners.

Dai, A. M.; and Le, Q. V. 2015. Semi-supervised Sequence Learning. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*, 3079–3087. Curran Associates, Inc.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Journal of Machine Learning Research*, volume 9, 249–256. ISSN 15324435. URL http://www.iro.umontreal.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*, 2672–2680. Curran Associates, Inc.

Grover, J.; and Mitra, P. 2017. Bilingual word embeddings with bucketed CNN for parallel sentence extraction. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Student Research Workshop*, 11–16. ISBN 9781945626562. doi:10.18653/v1/P17-3003.

Hangya, V.; Braune, F.; Kalasouskaya, Y.; and Fraser, A. 2018. Unsupervised Parallel Sentence Extraction from Comparable Corpora. In *Proceedings of the 15th International Workshop on Spoken Language Translation (IWSLT)*, 7–13.

Hangya, V.; and Fraser, A. 2020. Unsupervised parallel sentence extraction with parallel segment detection helps machine translation. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1224–1234. ISBN 9781950737482. doi:10.18653/v1/p19-1118.

Hill, F.; Cho, K.; and Korhonen, A. 2016. Learning distributed representations of sentences from unlabelled data. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 1367–1377. ISBN 9781941643914. doi:10.18653/v1/n16-1162.

Johnson, M.; Schuster, M.; Le, Q. V.; Krikun, M.; Wu, Y.; Chen, Z.; Thorat, N.; Viégas, F.; Wattenberg, M.; Corrado, G.; Hughes, M.; and Dean, J. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics* 5: 339–351. doi: 10.1162/tacl_a_00065.

Karthikeyan, K.; Wang, Z.; Mayhew, S.; and Roth, D. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Kim, Y.; Gao, Y.; and Ney, H. 2019. Effective Cross-lingual Transfer of Neural Machine Translation Models without Shared Vocabularies. 1246–1257. doi:10.18653/v1/p19-1120.

Kingma, D. P.; and Ba, J. L. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 127–133.

Lample, G.; and Conneau, A. 2019. Cross-lingual Language Model Pretraining. In *Advances in neural information processing systems*.

Lample, G.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Lample, G.; Conneau, A.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2018b. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018c. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 5039–5049. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-1549.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.

Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; and Zettlemoyer, L. 2020. Multilingual Denoising Pre-training for Neural Machine Translation .

Mezzini, M. 2018. Empirical study on label smoothing in neural networks. In *WSCG 2018 - Short papers proceedings*. doi:10.24132/csrn.2018.2802.25.

Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting Similarities among Languages for Machine Translation. *ArXiv* abs/1309.4168.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Munteanu, D. S.; Fraser, A.; and Marcu, D. 2004. Improved Machine Translation Performance via Parallel Sentence Extraction from Comparable Corpora. In *HLT-NAACL 2004: Main Proceedings*, 265–272.

Pires, T.; Schlinger, E.; and Garrette, D. 2019. How multilingual is Multilingual BERT? *arXiv preprint arXiv:1906.01502* .

Ren, S.; Zhang, Z.; Liu, S.; Zhou, M.; and Ma, S. 2019. Unsupervised Neural Machine Translation with SMT as Posterior Regularization. *Proceedings of the AAAI Conference on Artificial Intelligence* 33: 241–248. ISSN 2159-5399. doi:10.1609/aaai.v33i01.3301241.

Sennrich, R.; Haddow, B.; and Birch, A. 2016a. Improving neural machine translation models with monolingual data. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 1, 86–96. ISBN 9781510827585. doi:10.18653/v1/p16-1009.

Sennrich, R.; Haddow, B.; and Birch, A. 2016b. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3, 1715–1725. ISBN 9781510827585. doi: 10.18653/v1/p16-1162.

Shah, H.; and Barber, D. 2018. Generative neural machine translation. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, 1346–1355. ISSN 10495258.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T. Y. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *36th International Conference on Machine Learning, ICML 2019*, volume 2019-June, 10384–10394. ISBN 9781510886988.

Sun, H.; Wang, R.; Chen, K.; Utiyama, M.; Sumita, E.; and Zhao, T. 2019. Unsupervised Bilingual Word Embedding Agreement for Unsupervised Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1235–1245. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1119.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 4, 3104–3112. ISSN 10495258.

Vaswani, A.; Brain, G.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention Is All You Need. *Advances in neural information processing systems* (Nips): 5998–6008.

Vincent, P. 2010. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* 11: 3371–3408.

Wu, S.; and Dredze, M. 2020. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of Bert. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 833–844. ISBN 9781950737901. doi:10.18653/v1/d19-1077.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, L.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G. S.; Hughes, M.; and Dean, J. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv* abs/1609.08144.

Yang, Z.; Chen, W.; Wang, F.; and Xu, B. 2018. Unsupervised neural machine translation with weight sharing. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, 46–55. ISBN 9781948087322. doi:10.18653/v1/p18-1005.

Zhang, M.; Liu, Y.; Luan, H.; and Sun, M. 2017. Adversarial training for unsupervised bilingual lexicon induction. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1: 1959–1970. doi: 10.18653/v1/P17-1179.