

# A Multilingual View of Unsupervised Machine Translation

Xavier Garcia<sup>\*1</sup> Pierre Foret<sup>\*1</sup> Thibault Sellam<sup>1</sup> Ankur P. Parikh<sup>1</sup>

## Abstract

We present a probabilistic framework for multilingual neural machine translation that encompasses supervised and unsupervised setups, focusing on unsupervised translation. In addition to studying the vanilla case where there is only monolingual data available, we propose a novel setup where one language in the (source, target) pair is not associated with any parallel data, but there may exist auxiliary parallel data that contains the other. This auxiliary data can naturally be utilized in our probabilistic framework via a novel cross-translation loss term. Empirically, we show that our approach results in higher BLEU scores over state-of-the-art unsupervised models on the WMT’14 English-French, WMT’16 English-German, and WMT’16 English-Romanian datasets in most directions. In particular, we obtain a +1.65 BLEU advantage over the best-performing unsupervised model in the Romanian-English direction.

## 1. Introduction

The popularity of neural machine translation systems (Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016) has exploded in recent years. Those systems have obtained state-of-the-art results for a wide collection of language pairs, but they often require large amounts of parallel (source, target) sentence pairs to train (Koehn & Knowles, 2017), making them impractical for scenarios with resource-poor languages. As a result, there has been interest in *unsupervised machine translation* (Ravi & Knight, 2011), and more recently *unsupervised neural machine translation* (UNMT) (Lample et al., 2018; Artetxe et al., 2018), which uses only monolingual source and target corpora for learning. Unsupervised NMT systems have achieved rapid progress recently (Lam-

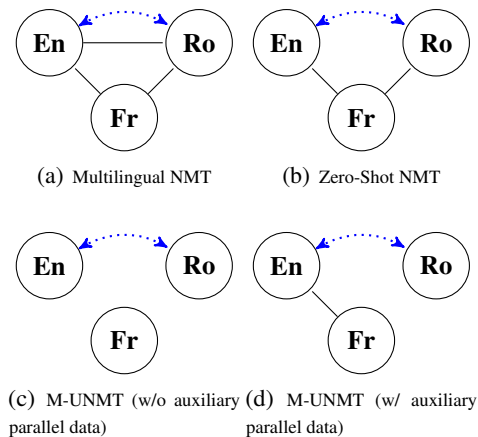


Figure 1. Different setups for English (En), French (Fr) and Romanian (Ro). The dashed edge indicates the target language pair. Full edges indicate the existence of parallel training data.

ple & Conneau, 2019; Artetxe et al., 2019; Ren et al., 2019; Li et al., 2020), largely thanks to two key ideas: one-the-fly back-translation (i.e., minimizing round-trip translation inconsistency) (Bannard & Callison-Burch, 2005; Sennrich et al., 2015; He et al., 2016; Artetxe et al., 2018) and pre-trained language models (Lample & Conneau, 2019; Song et al., 2019). Despite the difficulty of the problem, those systems have achieved surprisingly strong results.

In this work, we investigate *Multilingual UNMT* (M-UNMT), a generalization of the UNMT setup that involves more than two languages. Multilinguality has been explored in the supervised NMT literature, where it has been shown to enable information sharing among related languages. This allows higher resource language pairs (e.g. English-French) to improve performance among lower resource pairs (e.g., English-Romanian) (Johnson et al., 2017; Firat et al., 2016). Yet multilingual translation has only received little attention in the unsupervised literature, and the performance of preliminary works (Sen et al., 2019; Xu et al., 2019) is considerably below that of state-of-the-art bilingual unsupervised systems (Lample & Conneau, 2019; Song et al., 2019). Another line of work has studied zero-shot translation in the presence of a “pivot” language, e.g., using French-English and English-Romanian corpora to model French-Romanian (Johnson et al., 2017; Arivazhagan et al.,

<sup>\*</sup>Work done as a part of Google AI residency program. <sup>1</sup>Google Research, NY, USA. Correspondence to: Xavier Garcia <xgarcia@google.com>, Pierre Foret <pierreforet@google.com>, Thibault Sellam <tsellam@google.com>, Ankur P. Parikh <aparikh@google.com>.

2019; Gu et al., 2019; Al-Shedivat & Parikh, 2019). However, zero-shot translation is not unsupervised since one can perform two-step supervised translation through the pivot language.

We introduce a novel probabilistic formulation of multilingual translation, which encompasses not only existing supervised and zero-shot setups, but also two variants of Multilingual UNMT: (1) a strict M-UNMT setup in which there is no parallel data for any pair of language, and (2) a novel, looser setup where there exists parallel data that contains one language in the (source, target) pair but not the other. We illustrate those two variants and contrast them to existing work in Figure 1. As shown in Figures 1(c) and 1(d), the defining feature of M-UNMT is that the (source, target) pair of interest is not connected in the graph, precluding the possibility of any direct or multi-step supervised solution. Leveraging auxiliary parallel data for UNMT as shown in Figure 1(d) has not been well studied in the literature. However, this setup may be more realistic than the strictly unsupervised case since it enables the use of high resource languages (e.g. En) to aid translation into rare languages.

For the strict M-UNMT setup pictured in Figure 1(c), our probabilistic formulation yields a multi-way back-translation objective that is an intuitive generalization of existing work (Artetxe et al., 2018; Lample et al., 2018; He et al., 2020). We provide a rigorous derivation of this objective as an application of the Expectation Maximization algorithm (Dempster et al., 1977). Effectively utilizing the auxiliary parallel corpus pictured in Figure 1(d) is less straightforward since the common approaches for UNMT are explicitly designed for the bilingual case. For this setting, we propose two algorithmic contributions. First, we derive a novel *cross-translation* loss term from our probabilistic framework that enforces cross-language pair consistency. Second, we utilize the auxiliary parallel data for *pre-training*, which allows the model to build representations better suited to translation.

Empirically, we evaluate both setups, demonstrating that our approach of leveraging auxiliary parallel data offers quantifiable gains over existing state-of-the-art unsupervised models on 3 language pairs: En – Ro, En – Fr, and En – De. Additionally, we obtain a +1.65 BLEU advantage over the best unsupervised model on the WMT’16 Romanian-English machine translation task. Finally, we perform a series of ablation studies that highlight the impact of the additional data, our additional loss terms, as well as the choice of auxiliary language.

## 2. Background and Overview

**Notation:** Before discussing our approach, we introduce some notation. We denote random variables by capital letters  $X, Y, Z$ , and their realizations by their corresponding lowercase version  $x, y, z$ . We abuse this convention to compactly write objects like the conditional density  $p(Y = y|X = x)$  as  $p(y|x)$  or the marginalized distributions  $p(X = x)$  as  $p(x)$ , with the understanding that the lowercase variables are connected to their corresponding uppercase random variables. Given a random variable  $X$ , we write  $\mathbb{E}_{x \sim X}$  to mean the expectation with respect to  $x$ , where  $x$  follows the distribution of  $X$ . We use a similar convention for conditional distributions e.g. we write  $\mathbb{E}_{y \sim p(\cdot|x)}$  to denote the expectation of  $Y$  conditioned on  $X = x$ . Similarly, we write  $H(X)$  or  $H(p(x))$  to denote the entropy of the random variable  $X$  i.e.  $H(X) = \mathbb{E}_{x \sim X}[-\log p(x)]$ . We reserve the use of typewriter font for languages e.g.  $\mathbb{X}$ .

**Neural Machine Translation:** In bilingual supervised machine translation we are given a training dataset  $\mathcal{D}_{\mathbb{X}, \mathbb{Y}}$ . Each  $(x, y) \in \mathcal{D}_{\mathbb{X}, \mathbb{Y}}$  is a (source, target) pair consisting of a sentence  $x$  in language  $\mathbb{X}$  and a semantically equivalent sentence  $y$  in language  $\mathbb{Y}$ . We train a translation model using maximum likelihood:

$$L_{sup}(\theta) = \sum_{(x, y) \in \mathcal{D}_{\mathbb{X}, \mathbb{Y}}} \log p_{\theta}(y|x)$$

In neural machine translation,  $p_{\theta}(y|x)$  is modelled with the encoder-decoder paradigm where  $x$  is encoded into a set of vectors via a neural network  $\text{enc}_{\theta}$  and a decoder neural network defines  $p_{\theta}(y|\text{enc}_{\theta}(x))$ . In this work, we use a transformer (Vaswani et al., 2017) as the encoder and decoder network. At inference time, computing the most likely target sentence  $y$  is intractable since it requires enumerating over all possible sequences, and is thus approximated via beam search.

**Unsupervised Machine Translation:** The requirement of a training dataset  $\mathcal{D}_{\mathbb{X}, \mathbb{Y}}$  with source-target pairs can often be prohibitive for rare or low resource languages. Bilingual unsupervised translation attempts to learn  $p_{\theta}(y|x)$  using monolingual corpora  $\mathcal{D}_{\mathbb{X}}$  and  $\mathcal{D}_{\mathbb{Y}}$ . For each sentence  $x \in \mathcal{D}_{\mathbb{X}}$ ,  $\mathcal{D}_{\mathbb{Y}}$  may not contain an equivalent sentence in  $\mathbb{Y}$ , and vice versa.

State of the art unsupervised methods typically work as follows. They first perform pre-training and learn an initial set of parameters  $\theta$  based on a variety of language modeling or noisy reconstruction objectives (Lample & Conneau, 2019; Lewis et al., 2019; Song et al., 2019) over  $\mathcal{D}_{\mathbb{X}}$  and  $\mathcal{D}_{\mathbb{Y}}$ . A fine-tuning stage then follows which typically uses back-translation (Sennrich et al., 2016; Lample & Conneau, 2019; He et al., 2016) that involves translating  $x$  to the target

language  $Y$ , translating it back to a sentence  $x'$  in  $X$ , and penalizing the reconstruction error between  $x$  and  $x'$ .

**Overview of our Approach:** The following sections describe a probabilistic MT framework that justifies and generalizes the aforementioned approaches. We first model the case where we have access to several monolingual corpora, pictured in Figure 1(c). We introduce light independence assumptions to make the joint likelihood tractable and derive a lower bound, obtaining a generalization of the back-translation loss. We then extend our model to include the auxiliary parallel data pictured in Figure 1(d). We demonstrate the emergence of a *cross-translation* loss term, which binds distinct pairs of languages together. Finally, we present our complete training procedure, based on the EM algorithm. Building upon existing work (Song et al., 2019), we introduce a pre-training step that we run before maximizing the likelihood to obtain good representations.

### 3. Multilingual Unsupervised Machine Translation

In this section, we formulate our approach for M-UNMT. We restrict ourselves to three languages, but the arguments naturally extend to an arbitrary number of languages. Inspired by the recent style transfer literature (He et al., 2020), we introduce a generative model of which the available data can be seen as partially-observed samples. We first investigate the strict unsupervised case, where only monolingual data is available. Our framework naturally leads to an aggregate back-translation loss that generalizes previous work. We then incorporate the auxiliary corpus, introducing a novel cross-translation term. To optimize our loss, we leverage the EM algorithm, giving a rigorous justification for the stop-gradient operation that is usually applied in the UNMT and style transfer literature (Lample & Conneau, 2019; Artetxe et al., 2019; He et al., 2020).

#### 3.1. M-UNMT - Monolingual Data Only

We begin with the assumption that we have three sets of monolingual data,  $\mathcal{D}_x$ ,  $\mathcal{D}_y$ ,  $\mathcal{D}_z$  for languages  $X$ ,  $Y$  and  $Z$  respectively. We take the viewpoint that these datasets form the visible parts of a larger dataset  $\mathcal{D}_{x,y,z}$  of triplets  $(x, y, z)$  which are translations of each other. We think of these translations as samples of a triplet  $(X, Y, Z)$  of random variables and write the observed data log-likelihood as:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathcal{D}_x} + \mathcal{L}_{\mathcal{D}_y} + \mathcal{L}_{\mathcal{D}_z}$$

Our goal however is to learn a conditional translation model  $p_\theta$ . We thus rewrite the log likelihood as a marginalization over the unobserved variables for each dataset as shown

below:

$$\mathcal{L}(\theta) = \sum_{x \in \mathcal{D}_x} \log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} p_\theta(x|y, z) \quad (1)$$

$$+ \sum_{y \in \mathcal{D}_y} \log \mathbb{E}_{\substack{(x,z) \\ \sim (X,Z)}} p_\theta(y|x, z) \quad (2)$$

$$+ \sum_{z \in \mathcal{D}_z} \log \mathbb{E}_{\substack{(x,y) \\ \sim (X,Y)}} p_\theta(z|x, y) \quad (3)$$

Learning a model for  $p_\theta(x|y, z)$  is not practical since the translation task is to translate  $z \rightarrow x$  without access to  $y$ , or  $y \rightarrow x$  without access to  $z$ . Thus, we make the following structural assumption: given any variable in the triplet  $(X, Y, Z)$ , the remaining two are independent. We implicitly think of the conditioned variable as detailing the content and the two remaining variables as independent manifestations of this content in the respective languages. Using the fact that  $p_\theta(x|y, z) = p_\theta(x|y) = p_\theta(x|z)$  under this assumption, we rewrite the summand in (1) as follows:

$$\log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} p_\theta(x|y, z) = \log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} \sqrt{p_\theta(x|y)p_\theta(x|z)}.$$

Next, note that all these expectations in Eq. 1, 2, and 3 are intractable to compute due to the number of possible sequences in each language. We address this problem through the Expectation Maximization (EM) algorithm (Dempster et al., 1977). We first use Jensen's inequality<sup>1</sup>:

$$\begin{aligned} \log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} p_\theta(x|y, z) &= \log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} \frac{p_\theta(x|y, z)}{p_\theta(y, z|x)} p_\theta(y, z|x) \\ &= \log \mathbb{E}_{\substack{(y,z) \\ \sim p_\theta(y, z|x)}} \frac{p_\theta(x|y, z)}{p_\theta(y, z|x)} p(y, z) \\ &= \mathbb{E}_{\substack{(y,z) \\ \sim p_\theta(y, z|x)}} [\log p_\theta(x|y, z) + \log p(y, z)] \\ &\quad + H(p_\theta(y, z|x)) \end{aligned}$$

Since the entropy of a random variable is always non-negative, we can bound the quantity on the right from below as follows:

$$\begin{aligned} \log \mathbb{E}_{\substack{(y,z) \\ \sim (Y,Z)}} p_\theta(x|y, z) &\geq \mathbb{E}_{\substack{(y,z) \\ \sim p_\theta(y, z|x)}} [\log p_\theta(x|y, z) + \log p_\theta(y, z)] \\ &= \frac{1}{2} \mathbb{E}_{y \sim p_\theta(y|x)} \log p_\theta(x|y) \\ &\quad + \frac{1}{2} \mathbb{E}_{z \sim p_\theta(z|x)} \log p_\theta(x|z) \\ &\quad + \mathbb{E}_{\substack{(y,z) \\ \sim p_\theta(y, z|x)}} \log p(y, z) \end{aligned}$$

<sup>1</sup>This is actually an equality in this case since  $\frac{p_\theta(x|y, z)}{p_\theta(y, z|x)} p(y, z) = p(x)$  and hence the expectant does not actually depend on  $y$  or  $z$ .

Applying the above strategy to (2) and (3) and rearranging terms gives us:

$$\begin{aligned} \mathcal{L}(\theta) \geq & \frac{1}{2} \mathbb{E}_{y \sim p_\theta(\cdot|x)} \log p_\theta(x|y) + \frac{1}{2} \mathbb{E}_{z \sim p_\theta(\cdot|x)} \log p_\theta(x|z) \\ & + \frac{1}{2} \mathbb{E}_{x \sim p_\theta(\cdot|y)} \log p_\theta(y|x) + \frac{1}{2} \mathbb{E}_{z \sim p_\theta(\cdot|y)} \log p_\theta(y|z) \\ & + \frac{1}{2} \mathbb{E}_{y \sim p_\theta(\cdot|z)} \log p_\theta(z|y) + \frac{1}{2} \mathbb{E}_{x \sim p_\theta(\cdot|z)} \log p_\theta(z|x) \\ & + \mathbb{E}_{\substack{(y,z) \\ \sim p_\theta(\cdot, \cdot|x)}} \log p(y, z) + \mathbb{E}_{\substack{(x,z) \\ \sim p_\theta(\cdot, \cdot|y)}} \log p(x, z) \\ & + \mathbb{E}_{\substack{(x,y) \\ \sim p_\theta(\cdot, \cdot|z)}} \log p(x, y) \end{aligned} \quad (4)$$

This lower-bound contains two types of terms. The *back-translation* terms, e.g.,

$$\mathbb{E}_{y \sim p_\theta(\cdot|x)} \log p_\theta(x|y), \quad (5)$$

enforce that reciprocal translation models are consistent. The joint terms e.g.  $\mathbb{E}_{(x,y) \sim p_\theta(\cdot, \cdot|z)} \log p(x, y)$  will vanish in our optimization procedure, as explained next.

We use the EM algorithm to maximize Eq. 4. In our setup, the **E**-step at iteration  $t$  amounts to computing the expectations against the conditional distributions evaluated at the current set of parameters  $\theta = \theta^{(t)}$ . We approximate this by removing the expectations and replacing the random variable with the mode of its distribution i.e.  $\mathbb{E}_{y \sim p_{\theta^{(t)}}(\cdot|x)} \log p_{\theta^{(t)}}(x|y) \approx p_{\theta^{(t)}}(x|\hat{y})$  where

$\hat{y} = \arg \max_y p_{\theta^{(t)}}(y|x)$ . In practice, this amounts to running a greedy decoding procedure for the relevant translation models.

The **M**-step then corresponds to choosing the  $\theta$  which maximizes the resulting terms after we perform the **E**-step. Notice that for this step, the last three terms in Eq. 4 no longer possess a  $\theta$  dependence, as the expectation was computed in the **E**-step with a dependence on  $\theta^{(t)}$ . These terms can therefore be safely ignored, leaving us with only the back-translation terms. By our approximation to the **E**-step, these expressions become exactly the loss terms that appear in the current UNMT literature (Artetxe et al., 2019; Lample & Conneau, 2019; Song et al., 2019), see Figure 2(a) for a graphical depiction. Since computing the argmax is a difficult task, we perform a single gradient update for the **M**-step and define  $\theta^{(t+1)}$  inductively this way.

### 3.2. Auxiliary parallel data

We now extend our framework with an auxiliary parallel corpus (Figure 1(d)). We assume that we wish to translate from  $X$  to  $Z$ , and that we have access to a parallel corpus

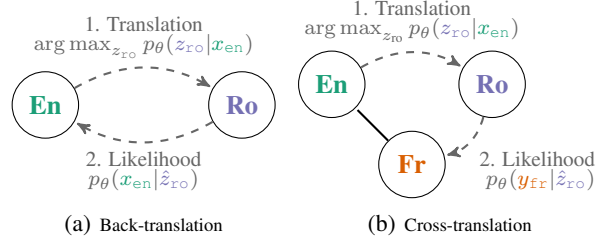


Figure 2. Illustration of the back-translation and cross-translation losses. Stop gradient is applied on step 1.

$\mathcal{D}_{x,y}$  that maps sentences from  $X$  to  $Y$ . To leverage this source of data, we augment the log-likelihood  $\mathcal{L}$  as follows:

$$\mathcal{L}_{\text{aug}}(\theta) = \mathcal{L}(\theta) + \sum_{\substack{(x,y) \\ \in \mathcal{D}_{x,y}}} \log \mathbb{E}_{z \sim Z} p_\theta(x, y|z) \quad (6)$$

Similar to how we handled the monolingual terms, we can utilize the EM algorithm to obtain an objective amenable to gradient optimization. By using the EM algorithm, we can substitute the distribution of  $Z$  in Eq. 6 with the one given by  $p_\theta(z|x, y)$ . The structural assumption we made in the case of monolingual data still holds: given any variable in the triplet  $(X, Y, Z)$ , the remaining two are independent. Using this assumption, we can rewrite the distribution  $p_\theta(z|x, y)$  as either  $p_\theta(z|x)$  or  $p_\theta(z|y)$ . Since we can decompose  $\log p_\theta(x, y|z) = \log p_\theta(x|z) + \log p_\theta(y|z)$ , we can leverage both formulations with an argument analogous to the one in §3.1:

$$\begin{aligned} \log \mathbb{E}_{z \sim Z} p_\theta(x, y|z) &= \log \mathbb{E}_{z \sim Z} p_\theta(x|z) p_\theta(y|z) \\ &\geq \mathbb{E}_{z \sim p_\theta(\cdot|y)} \log p_\theta(x|z) \\ &\quad + \mathbb{E}_{z \sim p_\theta(\cdot|x)} \log p_\theta(y|z) \\ &\quad + \mathbb{E}_{z \sim p_\theta(\cdot|y)} \log p(z) + \mathbb{E}_{z \sim p_\theta(\cdot|x)} \log p(z) \end{aligned} \quad (7)$$

A key feature of this lower bound is the emergence of the expressions:

$$\mathbb{E}_{z \sim p_\theta(\cdot|y)} \log p_\theta(x|z) \text{ and } \mathbb{E}_{z \sim p_\theta(\cdot|x)} \log p_\theta(y|z). \quad (8)$$

Intuitively, those terms ensure that the models can accurately translate from  $Y$  to  $Z$ , then  $Z$  to  $X$  (resp.  $X$  to  $Z$ , then  $Z$  to  $Y$ ). Because they enforce cross-language pair consistency, we will refer to them as *cross-translation* terms. In contrast, the back-translation terms, e.g., Eq. 5, only enforced monolingual consistency. We provide a graphical depiction of these terms in Figure 2(b).

As in the case of monolingual data, we optimize the full likelihood with EM. During the **E**-step, we approximate the



expectation with evaluation of the expectant at the mode of the distribution. As with §3.1, the last two terms in Eq. 7 disappear in the M-step.

### 3.3. Connections with supervised and zero shot methods

So far, we have only discussed multilingual unsupervised neural machine translation setups. We now derive the other configurations of Figure 1, that is, supervised and zero-shot translation, through our framework.

**Supervised translation:** Deriving supervised translation is straightforward. Given the parallel data dataset  $\mathcal{D}_{x,y}$ , we can rewrite the likelihood as:

$$\sum_{(x,y) \in \mathcal{D}_{x,y}} \log p_{\theta}(x, y) = \sum_{(x,y) \in \mathcal{D}_{x,y}} \log p_{\theta}(y|x) + \log p(x)$$

where the second term is a language model that does not depend on  $\theta$ .

**Zero-shot translation:** We can also connect the cross-translation term to the zero-shot MT approach from Al-Shedivat & Parikh (2019). Simplifying their setup, they consider three languages X, Y and Z with parallel data between X and Y as well as X and Z. In addition to the usual cross-entropy objective, they also add *agreement* terms i.e.  $\mathbb{E}_{z \sim p_{\theta}(\cdot|x)} \log p(z|y)$  and  $\mathbb{E}_{z \sim p_{\theta}(\cdot|y)} \log p(z|x)$ . We show that these agreement terms are operationally equivalent to the cross-translation terms i.e. Eq. 8. We first obtain the following equality by a simple application of Bayes' theorem:

$$\log p_{\theta}(y|z) = \log p_{\theta}(z|y) + \log p(y) - \log p(z).$$

We then apply the expectation operation  $\mathbb{E}_{z \sim p_{\theta}(\cdot|x)}$  to both sides of this equation. From an optimization perspective, we are only interested in terms involving the learnable parameters so we can dispose of the term involving  $\log p(y)$  on the right. Applying the same argument to  $\log p_{\theta}(x|z)$ , we obtain:

$$\begin{aligned} & \mathbb{E}_{z \sim p_{\theta}(\cdot|x)} \log p_{\theta}(y|z) + \mathbb{E}_{z \sim p_{\theta}(\cdot|y)} \log p_{\theta}(x|z) \\ &= \mathbb{E}_{z \sim p_{\theta}(\cdot|x)} \log p_{\theta}(z|y) + \mathbb{E}_{z \sim p_{\theta}(\cdot|y)} \log p_{\theta}(z|x) \\ & \quad - \mathbb{E}_{z \sim p_{\theta}(\cdot|x)} \log p(z) - \mathbb{E}_{z \sim p_{\theta}(\cdot|y)} \log p(z) \end{aligned}$$

By adding the quantity  $\mathbb{E}_{z \sim p_{\theta}(\cdot|x)} \log p(z) + \mathbb{E}_{z \sim p_{\theta}(\cdot|y)} \log p(z)$  to both sides of this inequality, the left-hand side becomes the lower bound introduced in the previous subsection, consisting of the cross-translations terms. The right-hand side consists of the agreement terms from Al-Shedivat & Parikh

### Algorithm 1 PRE-TRAINING

**Input:** Datasets  $\mathcal{D}$ , number of steps  $N$

```

1: Initialize  $\theta \leftarrow \theta_0$ 
2: for step in 1, 2, 3, ...,  $N$  do
3:   Choose dataset  $D$  at random from  $\mathcal{D}$ .
4:   if  $D$  consists of monolingual data then
5:     Sample batch  $x$  from  $D$ .
6:     Masked version of  $x$ :  $x_M \leftarrow \text{MASK}(x)$ 
7:     MASS Loss:  $ml \leftarrow \log p_{\theta}(x|x_M)$ 
8:     Update:  $\theta \leftarrow \text{optimizer\_update}(ml, \theta)$ 
9:   else if  $D$  consists of parallel data then
10:    Sample batch  $(x, y)$  from  $D$ .
11:    Translation Loss:  $tl \leftarrow \log p_{\theta}(y|x) + \log p_{\theta}(x|y)$ 
12:    Update:  $\theta \leftarrow \text{optimizer\_update}(tl, \theta)$ 
13:   end if
14: end for

```

(2019). We experimented using this term instead of our cross-translation terms, but we found it to be unstable. This could be attributed to the fact that we lack  $X \leftrightarrow Z$  parallel data, which is available in the setup of Al-Shedivat & Parikh (2019).

## 4. Training algorithms

We now discuss how to train the model end-to-end. We introduce a *pre-training phase* that we run before the EM procedure to initialize the model. Pre-training was already known to be crucial for UNMT (Lample & Conneau, 2019; Song et al., 2019). We make use of an existing method, MASS, and enrich it with the auxiliary parallel corpus if available. We refer to the EM algorithm described in §3 as *fine-tuning* for consistency with the literature.

### 4.1. Pre-training

The aim of the pre-training phase is to produce an intermediate translation model  $p_{\theta}$ , to be refined during the fine-tuning step. We pre-train the model differently based on the data available to us. For monolingual data, we use the MASS objective (Song et al., 2019). The MASS objective consists of masking randomly-chosen contiguous segments<sup>2</sup> of the input then reconstructing the masked portion. We refer to this operation as MASK. If we have auxiliary parallel data, we use the traditional cross-entropy translation objective. We describe the full procedure in Algorithm 1.

<sup>2</sup>We choose the starting index to be 0 or the total length of the input divided by two with 20% chance for either scenario otherwise we sample uniformly at random then take the segment starting from this index and replace all tokens with a [MASK] token.

**Algorithm 2** FINE-TUNING

**Input:** Datasets  $\mathcal{D}$ , languages  $\mathcal{L}$ 

```

1: Initialize  $\theta \leftarrow \theta_0$ 
2: while not converged do
3:   for  $D$  in  $\mathcal{D}$  do
4:     if  $D$  consists of monolingual data then
5:       Current language:  $l_D \leftarrow$  Language of  $D$ .
6:       Sample batch  $x$  from  $D$ .
7:       for  $l$  in  $\mathcal{L}, l \neq l_D$  do
8:         Translation to  $l$ :  $\hat{y}_l \leftarrow \text{Decode } p_\theta(\hat{y}_l|x)$ 
9:         Back-translation loss:  $\text{bt}_{l_D,l} \leftarrow \log p_\theta(x|\hat{y}_l)$ 
10:        Update:  $\theta \leftarrow \text{optimizer\_update}(\text{bt}_{l_D,l}, \theta)$ 
11:      end for
12:     else if  $D$  consists of parallel data then
13:       Sample batch  $(x, y)$  from  $D$ .
14:       Get source language:  $l_x \leftarrow$  Language of  $x$ .
15:       Get target language:  $l_y \leftarrow$  Language of  $y$ .
16:       for  $l$  in  $\mathcal{L}, l \neq l_x, l_y$  do
17:         Translation to  $l$ :  $\hat{z}_l \leftarrow \text{Decode } p_\theta(\hat{z}_l|x)$ 
18:         Cross-translation loss:  $\text{ct} \leftarrow \log p_\theta(y|\hat{z}_l)$ 
19:         Update:  $\theta \leftarrow \text{optimizer\_update}(\text{ct}, \theta)$ 
20:       end for
21:     end if
22:   end for
23: end while

```

## 4.2. Fine-tuning

During the fine-tuning phase, we utilize the objectives derived in Section 3. At each training step we choose a dataset (either monolingual or bilingual), sample a batch, compute the loss, and update the weights. If the corpus is monolingual, we use the back-translation loss i.e. Eq. 5. If the corpus is bilingual, we compute the cross-translation terms i.e. Eq. 8 in both directions and perform one update for each term. We detail the steps in Algorithm 2.

## 5. Experiments

In this section, we discuss the details of the experiments undertaken throughout this work. We conduct experiments on the language triplets English-French-Romanian with English-French parallel data, English-Czech-German with English-Czech parallel data and English-Spanish-French with English-Spanish parallel data, with the unsupervised directions chosen solely for the purposes of comparing with previous recent work (Lample & Conneau, 2019; Song et al., 2019; Ren et al., 2019; Artetxe et al., 2019).

### 5.1. Datasets and preprocessing

We use the News Crawl datasets from WMT as our sole source of monolingual data for all the languages considered.

We used the data from years 2007-2018 for all languages except for Romanian, for which we use years 2015-2018. We ensure the monolingual data is properly labeled by using the fastText language classification tool (Joulin et al., 2016) and keep only the lines of data with the appropriate language classification. For parallel data, we used the UN Corpus (Ziemiński et al., 2016) for English-Spanish, the 10<sup>9</sup> French-English Gigaword corpus<sup>3</sup> for the English-French and the CzEng 1.7 dataset (Bojar et al., 2016) for English-Czech. We preprocess all text by using the tools from Moses (Koehn et al., 2007), and apply the Moses tokenizer to separate the text inputs into tokens. We normalize punctuation, remove non-printing characters, and replace unicode symbols with their non-unicode equivalent. For Romanian, we also use the scripts from Sennrich<sup>4</sup> to normalize the scripts and remove diacritics. For a given language triplet, we select 10 million lines of monolingual data from each language and use SentencePiece (Kudo & Richardson, 2018) to create vocabularies containing 64,000 tokens of each. We then remove lines with more than 100 tokens from the training set. We compute the BLEU score (Papineni et al., 2002) using the multi-bleu.pl script from Moses to be consistent with the literature (Song et al., 2019; Lample & Conneau, 2019; Li et al., 2020).

### 5.2. Model architectures

We use Transformers (Vaswani et al., 2017) for our translation models  $p_\theta$  with a 6-layer encoder and decoder, a hidden size of 1024 and a 4096 feedforward filter size. We share the same encoder for all languages. Following XLM (Lample & Conneau, 2019), we use language embeddings to differentiate between the languages by adding these embeddings to each token’s embedding. Unlike XLM, we only use the language embeddings for the decoder side. We follow the same modification as done in Song et al. (2019) and modify the output transformation of each attention head in each transformer block in the decoder to be distinct for each language. Besides these modifications, we share the parameters of the decoder for every language.

### 5.3. Training configuration

For pre-training, we group the data into batches of 1024 examples each, where each batch consists of either monolingual data of a single language or parallel data, but not both at once. We pad sequences up to a maximum length of 100 SentencePiece tokens. During pre-training, we used the Adam optimizer (Kingma & Ba, 2015) with initial learning rate of 0.0002 and weight decay parameter of 0.01, as well as 4,000 warmup steps and a linear decay schedule for 1.2

<sup>3</sup><https://www.statmt.org/wmt10/training-giga-fren.tar>

<sup>4</sup><https://github.com/rsennrich/wmt16-scripts>

	En – Fr	Fr – En	En – De	De – En	En – Ro	Ro – En
<b>Models without auxiliary parallel data</b>						
XLM (Lample & Conneau, 2019)	33.4	33.3	27.0	34.3	33.3	31.8
MASS (Song et al., 2019)	37.50	34.90	28.30	35.20	35.20	33.10
D2GPo (Li et al., 2020)	37.92	34.94	28.42	35.62	36.31	33.41
Artetxe et al. (2019)	36.2	33.5	26.9	34.4	-	-
Ren et al. (2019)	35.4	34.9	27.7	35.6	34.9	34.1
mBART (Liu et al., 2020) <sup>5</sup>	-	-	<b>29.8</b>	34.0	35.0	30.5
<i>M-UNMT</i>	36.25	33.50	25.47	32.32	34.87	32.10
<b>Models with auxiliary parallel data</b>						
mBART (Liu et al., 2020)	-	-	-	-	-	33.9
<i>M-UNMT</i> (Only Pre-Train)	29.22	33.84	18.33	29.04	25.25	32.64
<i>M-UNMT</i> (Fine-Tuned)	<b>38.34</b>	<b>36.05</b>	28.73	<b>35.98</b>	<b>37.4</b>	<b>35.75</b>

Table 1. BLEU scores of various models for UNMT. *M-UNMT* refers to our approach. The En – Fr/Fr – En and En – De/De – En directions were on *newstest2014*, while the En – Ro/Ro – En directions were on *newstest2016*. For mBART, they presented various BLEU scores for the unsupervised setup with auxiliary parallel data so we took the largest one.

million steps. For fine-tuning, we used Adamax (Kingma & Ba, 2015) with the same learning rate and warmup steps, no weight decay, and trained the models until convergence. We used Google Cloud TPUs for pre-training and 8 NVIDIA V100 GPUs with a batch size of 3,000 tokens per GPU for fine-tuning.

#### 5.4. Results

We list our results in Table 1. We also include the results of six strong unsupervised baselines: (1) XLM (Lample & Conneau, 2019), a cross-lingual language model fine-tuned with back-translation; (2) MASS (Song et al., 2019), which uses the aforementioned pre-training task with back-translation during fine-tuning; (3) D2GPo (Li et al., 2020), which builds on MASS and leverages an additional regularizer by use of a data-dependent Gaussian prior; (4) The recent work of Artetxe et al. (2019) which leverages tools from statistical MT as well subword information to enrich their models; (5) the work of Ren et al. (2019) that explicitly attempts to pre-train for UNMT by building cross-lingual  $n$ -gram tables and building a new pre-training task based on them; (6) mBART (Liu et al., 2020), which is concurrent work that also examines a multilingual unsupervised setup, as well as one with auxiliary parallel data for the Romanian-English direction. We present scores for models trained with and without auxiliary data, corresponding to Figure 1(d) and Figure 1(c). We also include the results of our model after pre-training i.e. no back-translation or cross-translation objective, under the title *M-UNMT (Only Pre-Train)*.

Our models with auxiliary data obtain better scores for almost all translation directions. Pre-training with the auxiliary data by itself gives competitive results in two of the three  $X - En$  directions. Moreover, our approach outperforms (Liu et al., 2020) which also leverages auxiliary par-

allel data. This suggests that our improved performance comes from both our choice of objectives and the additional data.

## 6. Ablations

We perform a series of ablation studies to determine which aspects of our formulation explain the improved performance. The first study probes the effect of auxiliary parallel data on the pre-training and fine-tuning phases. The next study analyzes the impact of multilinguality on fine-tuning, by examining the contribution of each loss term. The last study investigates the impact of the choice of auxiliary language. We focus on the English-Romanian pair since Romanian is low-resource, which is the most-likely setting in which UNMT would be applied.

### 6.1. Impact of the auxiliary data

We first examine the value provided by the inclusion of the auxiliary data, focusing on the triplet English-French-Romanian. To that end, we study four types of training configurations: (1) Our implementation of MASS (Song et al., 2019), with only English and Romanian data. (2) No auxiliary parallel data during pre-training and fine-tuning with only the multi-way back-translation objective (3) No parallel data during the pre-training phase but available during the fine-tuning phase, allowing us to leverage the cross-translation terms. (4) Auxiliary parallel data available during both the pre-training and the fine-tuning phases of training. We also include the numbers reported in the original MASS paper (Song et al., 2019) as well as the best-performing model of the WMT’16 Romanian-English news translation task (Sennrich et al., 2016) and report them in Table 2.

The results show that leveraging the auxiliary data induces superior performance, even surpassing the supervised scores of Sennrich et al. (2016). These gains can manifest in either

<sup>5</sup>This is concurrent work, made public on Jan 22, 2020.

Configuration	En – Ro	Ro – En
<b>Bilingual configurations</b>		
MASS (Song et al., 2019)	35.20	33.10
MASS (Our implementation)	34.14	31.78
<b>M-UNMT configurations</b>		
No auxiliary data.	34.87	32.10
Auxiliary data in fine-tuning	36.57	34.32
Auxiliary data in both phases	37.4	35.75
<b>Supervised</b>		
(Sennrich et al., 2016)	28.2	33.9
mBART (Liu et al., 2020)	38.5	39.9

Table 2. En – Ro and Ro – En BLEU scores on *newstest2016* for different ways of leveraging multilinguality and the auxiliary parallel data. M-UNMT refers to our approach.

Languages	En – Ro	Ro – En
En, Fr, Ro	37.21	35.5
En, Es, Ro	37.38	35.21
En, Cs, Ro	36.37	34.15

Table 3. En – Ro and Ro – En BLEU scores for varying choices of auxiliary language on WMT *newstest2016*.

pre-training or fine-tuning, with superior performance when the auxiliary data is available to both phases of training.

## 6.2. Impact of the additional objectives

Given the strong performance of our model just after the pre-training phase, it would be plausible that the gains from multilinguality arise exclusively during the pre-training phase. To demonstrate that this is not the case, we investigate three types of fine-tuning configurations: (1) Disregard the auxiliary language and fine-tune using only back-translation with English and Romanian data as per Song et al. (2019). (2) Finetune with our multi-way back-translation objective. (3) Finetune with our multi-way back-translation objective and leverage the auxiliary parallel data through the cross-translation terms. We name these configurations **BT**, **M-BT**, and **Full** respectively. We plot the results of training for 100k steps in Figure 3, reporting the numbers on a modified version of the dev set from the WMT’16 Romanian-English competition where all samples with more than 100 tokens were removed.

In the En – Ro direction, the BLEU score of the **Full** setup dominates the score of the other approaches. Furthermore, the performance of **BT** decays after a few training steps. In the En – Ro direction, the BLEU score for the **BT** and **M-BT** reach a plateau about 1 point under **Full**. Those charts illustrate the positive effect of the cross-translation terms. We contrast the BLEU curves with the back-translation loss curves in Figure 3(c) and 3(d). We see that even that though the **BT** configuration achieves the lowest back-translation loss, it does not attain the largest BLEU score. This demonstrates that using back-translation for the desired (source, target) pair alone is not the best task for the fine-tuning

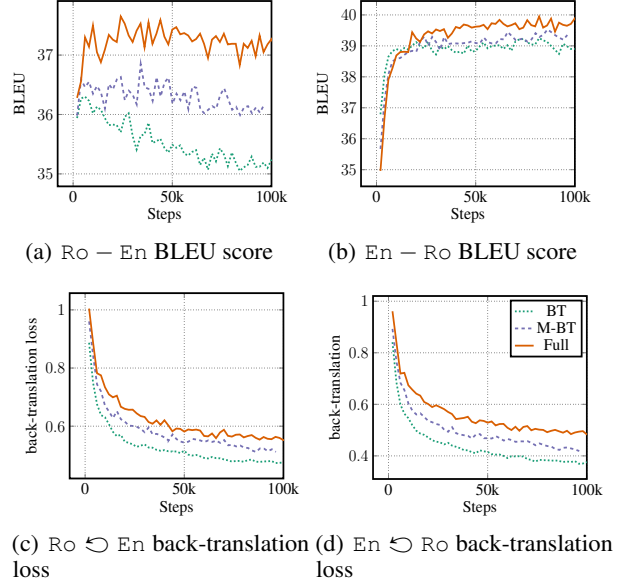


Figure 3. Back-translation losses and BLEU scores for the three configurations on our modified version of the WMT’16 dev set. We see that the multilinguality helps, as adding more back-translation terms with other languages involved improves the BLEU score at the cost of higher back-translation errors. From this viewpoint, the multilinguality acts as a regularizer, as it does for traditional supervised machine translation.

## 6.3. Impact of the choice of auxiliary language

In this study, we examine the impact of the choice of auxiliary language. We perform the same pre-training and fine-tuning procedure using either French, Spanish or Czech as the auxiliary language for the English-Romanian pair, with relevant parallel data of this auxiliary language into English. To isolate the effect of the language choice, we fixed the amount of monolingual data of the auxiliary language to roughly 40 million examples, as well as roughly 12.5 million lines of parallel data in the X-English direction.

We present the results in Table 3, indicating that using French or Spanish yields similar BLEU scores. Using Czech induces inferior performance, demonstrating that choosing a suitable auxiliary language plays an important role for optimal performance. Nevertheless, the configuration using Czech still outperforms the baselines, demonstrating the value of having any auxiliary parallel data at all.

## 7. Conclusion and Future Work

In this work, we explored a simple multilingual approach to UNMT and demonstrated that multilinguality and auxiliary parallel data offer quantifiable gains over strong baselines. We hope to explore massively multilingual unsupervised machine translation in the future.



## References

- Al-Shedivat, M. and Parikh, A. P. Consistency by agreement in zero-shot neural machine translation. In *NAACL*, 2019.
- Arivazhagan, N., Bapna, A., Firat, O., Lepikhin, D., Johnson, M., Krikun, M., Chen, M. X., Cao, Y., Foster, G., Cherry, C., et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019.
- Artetxe, M., Labaka, G., Agirre, E., and Cho, K. Unsupervised neural machine translation. In *ICLR*, 2018.
- Artetxe, M., Labaka, G., and Agirre, E. An effective approach to unsupervised machine translation. In *ACL*, 2019.
- Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- Bannard, C. and Callison-Burch, C. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 597–604. Association for Computational Linguistics, 2005.
- Bojar, O., Dušek, O., Kocmi, T., Libovický, J., Novák, M., Popel, M., Sudarikov, R., and Variš, D. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In Sojka, P., Horák, A., Kopeček, I., and Pala, K. (eds.), *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pp. 231–238, Cham / Heidelberg / New York / Dordrecht / London, 2016. Masaryk University, Springer International Publishing. ISBN 978-3-319-45509-9.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
- Firat, O., Cho, K., and Bengio, Y. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL*, 2016.
- Gu, J., Wang, Y., Cho, K., and Li, V. O. Improved zero-shot neural machine translation via ignoring spurious correlations. In *ACL*, 2019.
- He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., and Ma, W.-Y. Dual learning for machine translation. In *Advances in neural information processing systems*, pp. 820–828, 2016.
- He, J., Wang, X., Neubig, G., and Berg-Kirkpatrick, T. A probabilistic formulation of unsupervised text style transfer. In *ICLR*, 2020.
- Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.
- Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification. In *EACL*, 2016.
- Kalchbrenner, N. and Blunsom, P. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, 2013.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- Koehn, P. and Knowles, R. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*, 2017.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pp. 177–180, 2007.
- Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.
- Lample, G. and Conneau, A. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.
- Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.
- Li, Z., Wang, R., Chen, K., Utiyama, M., Sumita, E., Zhang, Z., and Zhao, H. Data-dependent gaussian prior objective for language generation. In *ICLR*, 2020.
- Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*, 2020.

- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.
- Ravi, S. and Knight, K. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 12–21, 2011.
- Ren, S., Wu, Y., Liu, S., Zhou, M., and Ma, S. Explicit cross-lingual pre-training for unsupervised machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 770–779, 2019.
- Sen, S., Gupta, K. K., Ekbali, A., and Bhattacharyya, P. Multilingual unsupervised nmt using shared encoder and language-specific decoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3083–3089, 2019.
- Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.
- Sennrich, R., Haddow, B., and Birch, A. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*, 2016.
- Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pp. 5926–5936, 2019.
- Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- Xu, C., Qin, T., Wang, G., and Liu, T.-Y. Polygon-net: a general framework for jointly boosting multiple unsupervised neural machine translation models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 5320–5326. AAAI Press, 2019.
- Ziemska, M., Junczys-Dowmunt, M., and Pouliquen, B. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 3530–3534, 2016.