Unsupervised Neural Machine Translation

Noah Golowich   Andrew Jin   Jonah Philion   Jesse Zhang

Abstract

In the task of unsupervised neural machine translation (NMT), one is given only monolingual data and uses it to infer a translation model between two languages. We describe our implementation of an unsupervised NMT model following (Artetxe et al., ICLR 2018). In addition we consider two new approaches for the task of unsupervised neural machine translation, one using an adversarially regularized auto-encoder, and the other using variational attention. For the latter, we describe some promising results and indicate some clear directions for further work.

1. Introduction

Rapid progress in the area of neural machine translation (NMT) has enabled deep learning-based approaches to reach remarkable, near human-level accuracy on multiple major language pairs (Sutskever et al., 2014; Bahdanau et al., 2014). However, current state-of-the-art approaches rely on enormous parallel corpora, requiring millions of parallel sentences for training before surpassing the performance of traditional phrase-based statistical machine translation systems. Unfortunately, sufficiently large parallel corpora are costly and difficult to obtain, particularly for low-resource languages, or for domain adaptation to highly specialized topic areas. Even with major languages where parallel sentences may be more readily accessible, developing general NMT systems to handle all the different combinations of language pairs very quickly becomes intractable. Thus, reducing the reliance of NMT on parallel data remains a significant open challenge.

Due to the generally greater availability of large monolingual corpora (including for low-resource language pairs), much attention has been paid to semi-supervised methods. These approaches augment unsupervised training on monolingual data with limited supervision through small sets of parallel sentences, large parallel corpora in related languages, or bilingual dictionaries.

More recently, three novel (though fairly similar) methods have been proposed to train unsupervised NMT models on monolingual data only, eliminating the need for any cross-lingual information whatsoever (Artetxe et al., 2017b; Lample et al., 2018a;b). Understandably, these approaches cannot compete with fully supervised NMT systems trained on millions of parallel sentences, but they nevertheless reach remarkable and exciting levels of performance. For the en-fr language pair, Artetxe et al., 2018 and Lample et al., 2018a achieve BLEU scores of around 14 to 15 on the WMT dataset. Lample et al., 2018b combines elements of the previous two approaches to achieve impressive BLEU scores of around 24 to 25 with NMT methods (and 26 to 27 if phrase-based methods are used as well).

In light of the results of these three recent publications, the contributions and goals of this final project were twofold: (i) (Lample et al., 2018b) shares many important methodological elements with (Artetxe et al., 2017b) yet performs significantly better, so we aimed to understand the factors underlying its success through exploring the impact of the former's embedding and denoising approaches; (ii) all three approaches heavily utilize denoising autoencoders. In our opinion, this emphasis on denoising seemed suboptimal, and their explanations for it were not convincing enough, so we attempted to improve performance by replacing the denoising autoencoders with variational, attention-based autoencoders.

Since (Artetxe et al., 2017b) released their code, we chose to re-implement it as our baseline model and basis for extensions. Just the reimplementation itself (mostly from scratch) proved to be a challenging (but highly educational) task. We remark that the released code ran extremely slowly and was not actually functional due to out-of-memory errors. In the end, our implementation ran approximately 10-15 times faster. We achieved BLEU scores of 24-29 on the Multi30k dataset for the en-fr language pair, which are comparable to the results of Lample et al., 2018a, and we made a successful initial foray into replacing the denoising autoencoder of (Artetxe et al., 2017b) with a variational, attention-based one.

The remainder of this paper is organized as follows. Section 2 provides background on unsupervised NMT with monolingual corpora only. Section 3 reviews related work, as well as some of the cross-lingual word embedding methods that laid the foundations for this work. Sections 4 to 6 describe the proposed model, training methods, and experimental settings. We then present experimental results in Section 7. Finally, we discuss the obtained results in Section 8 and summarize our findings in Section 9.

Notational conventions. We use the following notational conventions. We let $x_1, \ldots, x_S$ denote the tokens of a source sentence, and $y_1, \ldots, y_T$ denote the tokens of a target sentnece. If necessary, we will use superscripts $x^{(1)}, \ldots, x^{(n)}$ to denote different sentences in a dataset. If necessary to avoid ambiguity, we will use boldface $\boldsymbol{x}_i, \boldsymbol{y}_i$ to denote the corresponding word embeddings (which are vectors in some $\mathbb{R}^d$). We use $\boldsymbol{h}_s^{(src)}$, $1 \leq s \leq S$ to denote the hidden states corresponding to an RNN over the source sentences, so that $\boldsymbol{h}_s^{(src)} = RNN(\boldsymbol{h}_{s-1}^{(src)}, \boldsymbol{x}_s)$ for $1 \leq s \leq S$. To denote the hidden states corresponding to an RNN over the target sentence, we use the superscript (trg), as in $\boldsymbol{h}_t^{(trg)}$. (Note that, as described below, in the unsupervised NMT framework, we build translation models in both directions; however, there is still always a concept of "source" and "target" sentence depending on which reconstruction or back-translation error we are minimizing; see Section 4 for more details.) The vector $\boldsymbol{z}$ is used to denote the latent state in a latent variable model.

2. Background

Unsupervised NMT can be boiled down into three primary principles: initialization, language modeling, and iterative back-translation (Lample et al., 2018b). This background section aims to provide a high-level overview, introducing each principle in a general, abstract manner. The following section on related work will then cover how Artetxe et al., 2018, Lample et al., 2018a, and Lample et al., 2018b architecture unique solutions that address these primary principles of unsupervised NMT.

2.1. Initialization

Typical unsupervised NMT systems begin by initializing a model that aligns individual words, short phrases, or even sub-word units (e.g. through byte-pair encoding strategies) from the two languages in the translation pair. This alignment can be accomplished in numerous ways, but a common approach involves inferring a bilingual dictionary by mapping between the monolingual word embedding spaces in an unsupervised manner (Conneau et al., 2018; Artetxe et al., 2017). These cross-lingual word embeddings, discussed in greater detail in Section 3, prove to be quite effective and achieve high levels of accuracy (i.e. 70 to 80 percent) on word translation tasks. Consequently, they are used to obtain an initial "word-by-word" translation that captures correct semantic meaning (but not correct language structure).

2.2. Language Modeling

The large monolingual corpora at the unsupervised NMT system's disposal can be leveraged to train effective language models on the source and target languages. The aforementioned initialization component can preserve semantic meaning through "word-by-word" translation, but language models are required to improve translation quality by substituting and reordering words to capture the structure of the target language. Most current approaches employ denoising autoencoders as the language model of choice, corrupting input sentences with random word drops, adjacent swaps, and constrained permutations. It is claimed that such noise models produce noisy permutations analogous to what can be observed in word-by-word translation.

2.3. Iterative Back-translation

In unsupervised NMT, the back-translation component involves simultaneously training a forward model (that translates from source to target language) and a backward model (that translates from target to source language). The backward model can be applied to produce noisy parallel source sentences for the target sentences of a monolingual corpus, essentially transforming the challenging unsupervised learning task into a supervised one. The initial back-translation model (e.g. naïve word-by-word replacement with an inferred bilingual dictionary) can be iteratively refined, as the translation model continuously improves. Adversarial learning may also be incorporated into this iterative back-translation process (Lample et al., 2018a).

3. Related Work

3.1. Cross-Lingual Word Embeddings

Unsupervised NMT with only monolingual corpora is highly challenging because without parallel data, there are countless possible ways to associate source and target sentences. Thus, progress in learning unsupervised cross-lingual word embeddings set the foundation for success in unsupervised NMT by providing ways to roughly align monolingual corpora (i.e. the initialization component). Artetxe et al., 2017 and Conneau et al., 2018 proposed methods to perform linear mappings from source to target embedding space through self-learning (beginning from a small 25-word dictionary) or adversarial training. Using fastText vectors trained on Wikipedia, they achieve impressive accuracy rates of around 80 percent on en-fr word translation tasks (but only around 40 percent with word2vec on WaCky), and serve as the bases for the unsupervised NMT systems of Artetxe et al., 2018 and Lample et al., 2018a respectively.

3.2. Artetxe et al., 2018, Lample et al., 2018a, and Lample et al., 2018b

In this section, we compare the three recent unsupervised NMT publications in the context of the three principles from Section 2. With respect to initialization, Artetxe et al., 2018 and Lample et al., 2018a employ inferred bilingual dictionaries from the cross-lingual word embeddings developed by Artetxe et al., 2017 and Conneau et al., 2018 respectively. Lample et al., 2018b eliminates the need for bilingual dictionaries by instead aligning sub-word units of byte-pair encodings (BPE). They learn embeddings of BPE tokens, defined by jointly processing the source and target monolingual corpora.

For language modeling, all three approaches utilize denoising autoencoders. Artetxe et al., 2018 corrupts sentences by randomly swapping adjacent words, while Lample et al., 2018a and Lample et al., 2018b's noise model randomly drops words and applies constrained permutations.

Finally, for the back-translation component, Lample et al., 2018a is unique as it iteratively improves on a naïve, initial word-by-word translation model. Moreover, Artetxe et al., 2018 and Lample et al., 2018b both leverage shared encoder representations with fixed embeddings to act like an interlingua, while Lample et al., 2018a instead achieves a similar effect through adversarial training.

4. Model

In this section we describe the three models we implemented; we refer to them as the ARAE, the unsupervised NMT model, and the unsupervised variational attention model. We have implemented a fourth model, a local unsupervised variational attention model, but due to resource constraints did not have time to test it; we believe that it, as well as more careful parameter tuning, will correct many of the problems with the variational attention model, and is a promising direction for future work, as we discuss in Section 8.

4.1. Adversarially Regularized Autoencoder Translator (ARAE)

Aversarially Regularized Autoencoders compress discrete, sequential data into a continuous code space. Translation with the ARAE is achieved by regularizing the code space to be independent of language.

For monolingual corpora, the ARAE model consists of encoder enc_ϕ, a generator g_θ, a critic f_w, and a decoder p_ψ. The encoder and decoder are trained to recreate inputs x^i. Simultaneously, the distribution of the code space $\text{enc}_\phi(x^i)$ is constrained to be indistinguishable from output of the generator $g_\theta(x)$ for $x \sim N(0, I)$ using a Wasserstein GAN (Arjovsky et al., 2017).

$$L(\phi, \psi, \theta) = L_{rec}(\phi, \psi) + W(\mathbb{P}_r, \mathbb{P}_g)$$

Where W is the Wasserstein distance between the distribution of outputs of the encoder and distribution of outputs of the generator as measured by f_w.

To adapt the model for translation, we use one decoder for each language and place an adversarial loss from classifier p_u on the distributions of $\text{enc}_\phi(x)$ for $x \sim D_l$ for all languages l.

$$L(\phi, \psi, \theta) = L_{rec}(\phi, \psi) + W(\mathbb{P}_r, \mathbb{P}_g) + L_{class}(\phi, u)$$

The encoder and decoder are parameterized as LSTMs and the classifier, critic, and generator are parameterized as MLPs. During translation from language l to l', a sentence from language l is fed to the universal encoder and decoded by the decoder associated with l'. Since the model is non-attentional, unsupervised translation can be attempted without pretrained cross lingual word embeddings.

4.2. Unsupervised Neural Machine Translator

In this section we briefly describe the components of the unsupervised neural machine translation model used in (Artetxe et al., 2017b) (and which we reimplement). Throughout this paper we suppose that we are given as training data two monolingual corpora in different languages, which we refer to as ℓ_1, ℓ_2 at times.

- **Word embeddings.** We use pre-trained cross-lingual word embeddings that are aligned using a technique such as that in (Artetxe et al., 2017a; Conneau et al., 2018) (we use the former to align fastText (Bojanowski et al., 2017) embeddings).
- **Shared encoder.** The fixed and aligned embeddings are used with a shared encoder RNN to encode sentences from either language into a shared latent space, consisting of the hidden states of the RNN. This produces a language-independent reprsentation of the input text.
- **Decoder.** We use separate RNN decoders with attention for each language, which, given a latent state (i.e. sequence of hidden states for each RNN encoder), computes word-level probabilities for the underlying sentence corresponding to that latent state. We use the "general" attention mechanism from (Luong et al., 2015).
- **Denoising.** We have a training term enforcing that the composition of the encoder and the decoder for langauge ℓ_1, when passed a sentence $x = (x_1, \ldots, x_S)$ from language ℓ_1, is the identity. This task itself is trivial, but to encourage the encoder and the decoder to learn actual language models, we first swap $S/2$ random pairs of consecutive words of x to produce a noisy version $N(x)$ which is then fed to the encoder.
- **Back-translation.** To enforce the encoder and decoder to produce similar latent states given translations of the same sentence, we use on-the-fly back-translation (Artetxe et al., 2017b) as an additional term in the loss; this specific term is described further in Section 5.2.

4.3. Unsupervised Variational Attention

In this section we describe our adaptation of the variational attention approach introduced in (Bahuleyan et al., 2017) to the unsupervised NMT setting. Our goal is ultimately to replace the denoising autoencoder in existing models with our variational attention approach. Roughly speaking, both approaches force the encoder and decoder to produce and decode, respectively, representations of the data in a way that is robust to noise. The denoising autoencoder accomplishes this by adding noise to the input tokens, whereas the variational autoencoder does so by adding noise to the attention distribution. The variational autoencoder has the added benefit of being able to draw latent states from the prior distribution and generate artifical data; we leave this appliation for future work, however.

Recall that for standard (deterministic) attention models, the probability of an output word is predicted as

$$p(y_t | y_{<t}, x) = \text{softmax}(W_{gen} \boldsymbol{c}_t), \tag{1}$$

where $\tilde{\boldsymbol{h}}_t$, $1 \leq t \leq T$, is computed as $\tilde{\boldsymbol{h}}_t = \tanh(W_c[\boldsymbol{c}_t; \boldsymbol{h}_t^{(trg)}]$. Here $\boldsymbol{c}_t$ is a context vector computed from the hidden states $\boldsymbol{h}_s$ of the source-side encoder using an attention, and $\boldsymbol{h}_t^{(trg)}$ is the hidden state at timestep t of the target-side decoder. In particular, we have $\boldsymbol{c}_t = \sum_{s=1}^S \alpha_{ts} \boldsymbol{h}_s^{(src)}$ for some positive real scalars α_{ts}, which sum to 1.

Now, we describe variational encoder-decoders and variational attention in the unsupervised NMT setting. Given a collection of sentences $Y = (y^{(1)}, \ldots, y^{(n)})$ from one dataset, we suppose that there is an invertible translation function $\mathcal{T}$ which maps $Y \to \mathcal{T}(Y) = (\mathcal{T}(y^{(1)}), \ldots, \mathcal{T}(y^{(n)})) = (x^{(1)}, \ldots, x^{(n)}) =: X$ into another language. In the case of reconstruction loss, then $\mathcal{T}$ is the identity, and in the case of back-translation loss, $\mathcal{T}$ is the back-translation model (i.e. the model at the previous iteration of the algorithm). We may therefore also view Y as a function of X since we make the simplifying assumption that $\mathcal{T}$ is invertible, i.e. $Y = \mathcal{T}^{-1}(X)$.

Now let ϕ denote the parameters of the shared encoder, and θ denote the parameters of the decoders, as in the previous section. Given a latent variable Z (which we will describe in more detail below; for now we only assume we have local latent variables, so that we can write $Z = (\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)})$), we consider the generative model parametrized by the parameters θ of the encoder, $p_\theta(Z, Y) = p(Z) p_\theta(Y|Z)$. We suppose that ϕ parametrize a probabilistic decoder, $q_\phi(Z|Y)$. The marginal likelihood of a given data point $y^{(i)} \in Y$ (i.e. $y^{(i)}$ is a sentence) is then lower-bounded by the standard evidence lower bound:

$$\begin{aligned} \log p_\theta(y^{(i)}) &\geq \mathbb{E}_{Z \sim q_\phi(z^{(i)}|y^{(i)}} \left[\log \left(\frac{p_\theta(y^{(i)}, \boldsymbol{z}^{(i)})}{q_\phi(\boldsymbol{z}^{(i)}|\boldsymbol{y}^{(i)})} \right) \right] \\ &= \mathbb{E}_{Z \sim q_\phi(z^{(i)}|y^{(i)})} [\log p_\theta(y^{(i)}|z^{(i)}] \\ &\quad - KL(q_\phi(z^{(i)}|y^{(i)})||p(z^{(i)})) := \mathcal{L}^{(i)}(\theta, \phi). \end{aligned} \tag{2}$$

By our assumption that Y is a function $Y = \mathcal{T}^{-1}(X)$

of X, the above becomes

$$\begin{aligned} & \mathcal{L}^{(i)}(\theta, \phi) \\ = \; & \mathbb{E}_{z^{(i)} \sim q_\phi(z^{(i)}|x^{(i)})}[\log p_\theta(y^{(i)}|\boldsymbol{z}^{(i)})] \qquad (3) \\ & -KL(q_\phi(\boldsymbol{z}^{(i)}|x^{(i)})||p(\boldsymbol{z}^{(i)})). \end{aligned}$$

In (Bahuleyan et al., 2017), the latent state $\boldsymbol{z}^{(i)}$ had the form $\boldsymbol{z}^{(i)} = (\boldsymbol{w}^{(i)}, \boldsymbol{c}_1^{(i)}, \ldots, \boldsymbol{c}_T^{i)})$, where $\boldsymbol{c}_t^{(i)}$, $1 \le t \le T$, are the context vectors computed using attention at each step of translation of the target sentence, as described above. It is clear how the probabilities of the output words $y^{(i)}$ can be computed from the vectors $(\boldsymbol{w}^{(i)}, \boldsymbol{c}_1^{(i)}, \ldots, \boldsymbol{c}_T^{(i)})$. The prior on the latent variables $\boldsymbol{c}_1^{(i)}, \ldots, \boldsymbol{c}_T^{(i)})$ was chosen either to be an independent standard Gaussian $\boldsymbol{c}_t^{(i)} \sim \mathcal{N}(0, I_d)$ for $1 \le t \le T$, or to be a Gaussian with unit covariance and mean equal to the average of $\boldsymbol{h}_s^{(src)}$; that is, $\boldsymbol{c}_t^{(i)} \sim \mathcal{N}(\frac{1}{S}\sum_{s=1}^S \boldsymbol{h}_s^{(i)}, I_d)$.

With this latter prior, the prior depends on the source data $x^{(i)}$ (through $\boldsymbol{h}_s^{(i)}, 1 \le s \le S$), so does not exactly follow the theoretical setup in (3). This will also be an issue in our model, as we describe below, and resolving this difference is an interesting area for future work. Note that this is similar to the setting of (Zhang et al., 2016), whereby all variables are further conditioned on the source sentence $x^{(i)}$. In that work, the prior has the form $p_\theta(z^{(i)}|x^{(i)})$ and the posterior has the form $q_\phi(z^{(i)}|x^{(i)}, y^{(i)})$. However, using this model leads to the problem that at inference time, the target sentences $y^{(i)}$ are not available, which is why (Bahuleyan et al., 2017) decide to go with the model (2) where Y is modeled as a deterministic function of Y.

Note that the individual words $y_t^{(i)}$ of the target sentence are assumed to be conditionally independent given the latent state $\boldsymbol{z}^{(i)}$. Moreover, assuming that the distribution of each context vector $\boldsymbol{c}_{t+1}^{(i)}$ is computed as a deterministic function of the distributions of the previous context vectors $\boldsymbol{w}^{(i)}, \boldsymbol{c}_1^{(i)}, \ldots, \boldsymbol{c}_t^{(i)}$ (not of the context vectors themselves, which are random variables), then the random variables $\boldsymbol{w}^{(i)}, \boldsymbol{c}_t^{(i)}, 1 \le t \le T$, are all independent under $q_\phi(\cdot|x^{(i)})$. This latter assumption holds during inference if we use the mean of the distribution of each $\boldsymbol{c}_t^{(i)}$ to compute each next word (i.e. using greedy search), and it holds during training with teacher forcing if we do not use input feeding (which is the case in our experiments). In this case, we may therefore decompose the loss in (3) into one term for each target word, i.e.

$$\begin{aligned} \mathcal{L}_t^{(i)}(\theta, \phi) \;\; = \;\; & \mathbb{E}_{z^{(i)} \sim q_\phi(z^{(i)}|x^{(i)})}[\log p_\theta(y_t^{(i)}|\boldsymbol{z}^{(i)})] \\ & -KL(q_\phi(\boldsymbol{c}_t^{(i)}|x^{(i)})||p(\boldsymbol{z}^{(i)})), \end{aligned}$$

where we are slightly abusing the notation $q_\phi(\cdot|x^{(i)})$.

There are two main departures for our variational model from the one in (Bahuleyan et al., 2017). When describing them, we drop the superscripts indexing the sentences of the dataset for simplicity.

4.3.1. Elimination of $\boldsymbol{w}$

First, we eliminate the latent variable $\boldsymbol{w}$, which is used in (Bahuleyan et al., 2017) to initialize the hidden state for the decoder RNN. Instead, we deterministically initialize the hidden state of the decoder RNN from the final hidden state of the encoder RNN. This is equivalent to setting the prior of $\boldsymbol{w}$ to be a delta function with its point mass on the final state of the encoder RNN. Note that this prior then is no longer a "true" prior since it depends on the data x; we discuss this issue further in a different context below.

4.3.2. Dirichlet posteriors

The second departure is that we model the posterior and prior over the context variables $\boldsymbol{c}_t$ differently. In particular, rather than modeling $\boldsymbol{c}_t$ directly as a latent variable, we recall that it is written as $\boldsymbol{c}_t = \sum_{s=1}^S \alpha_{ts}\boldsymbol{h}_s^{(src)}$, and then view the variables α_{ts} as the latent state. This is exactly the approach proposed in the Conclusion of (Bahuleyan et al., 2017) but not implemented there. Since for each t, the parameters $\alpha_{ts}, 1 \le s \le S$, parametrize a categorical distribution, we should take the family that parametrizes the α_{ts} to be the conjugate prior of the categorical, namely the Dirichlet distribution. Therefore, the variational distribution $q_\phi(\boldsymbol{c}_t|x)$ is computed as follows: given the source tokens $x_1, \ldots, x_S$, and the uniquely determined target tokens $y_1, \ldots, y_T$ (by our assumption of $\mathcal{T}$ above), scores are first computed via:

$$\sigma_{st} := score(\boldsymbol{h}_t^{(trg)}, \boldsymbol{h}_s^{(src)}) := \boldsymbol{v}_a^T \cdot \tanh(\boldsymbol{W}_a[\boldsymbol{h}_t^{(trg)}; \boldsymbol{h}_s^{(src)}]),$$

which corresponds to the "concat" method of attention in (Luong et al., 2015). Next, for each t, the random variables $\alpha_{t1}, \ldots, \alpha_{tS}$ are distributed according to the Dirichlet distribution,

$$Dirichlet(\exp(\sigma_{t1}), \ldots, \exp(\sigma_{tS})).$$

Then the random variables $\boldsymbol{c}_t$ are given by $\boldsymbol{c}_t = \sum_{s=1}^s \alpha_{ts}\boldsymbol{h}_s^{(src)}$. Note that we exponentiate the σ_{ts} since the parameter space of the Dirichlet family requires that all parameters be positive; the magnitude of these parameters, roughly speaking, corresponds to the variance of the random variables α_{ts}. In particular, if all of the $\exp(\sigma_{ts})$ are roughly equal, then the variance of α_{ts} is approximately $\frac{1}{S\cdot\sum_{s=1}^{S}\exp(\sigma_{ts})}$.

As the prior $p((\boldsymbol{c}_1,\ldots,\boldsymbol{c}_T))$ over the context variables, we choose the distribution induced by taking $(\alpha_{t1},\ldots,\alpha_{tS})$ to be $Dirichlet(1/S,\ldots,1/S)$. The variance of each α_{ts} then grows roughly as $1/S$, which keeps the sum of the variances of the individual α_{ts} constant. We emphasize that it is an interesting direction of future work to consider different priors over the latent variables α_{ts} and to find some theoretical explanation for which one works best.

We also note that, similar to the situation with (Bahuleyan et al., 2017) mentioned above, the prior $p(\boldsymbol{z})$ here depends on the data (x, y) through the hidden states $\boldsymbol{h}^{(src)}, \boldsymbol{h}^{(trg)}$ (though the random varaibles α_{ts} do not), so we do not quite follow the standard setup of a variational auto-encoder. Understanding the theoretical implications for this heuristic is an important direction for future work.

5. Training

5.1. ARAE

The ARAE Translator is trained verbatim as in (Junbo et al., 2017).

5.2. Unsupervised NMT Model

We re-implemented (from scratch) the unsupervised model described in Section 4.2 from (Artetxe et al., 2017b). The original implementation appears to have a bug that causes the process to run out of memory when allowing the matrix W_{gen} in (1) to be trainable (instead, those weights must be fixed to the fixed embeddings for that language). Our implementation fixed this bug and runs several times faster due to significant changes in the overall structure of our code.

To train, we suppose we are given as input two monolingual corpora, $X = (x^{(1)},\ldots,x^{(n)}), \tilde{X} = (\tilde{x}^{(1)},\ldots,\tilde{x}^{(\tilde{n})}$. We minimize the sum of word-level cross-entropy losses for both reconstruction and back-translation losses. In particular, the loss we minimize is the following. Letting d_θ denote the decoder for the language corresponding to X, $\tilde{d}_\theta$ denote the decoder for the language corresponding to $\tilde{X}$, and e_ϕ denote the (shared) encoder, then the back translation loss for the $X \to \tilde{X}$ direction is given by

$$\mathcal{L}_{back}(\phi,\theta,X) = \mathbb{E}_{x\sim X, y=\tilde{d}_\theta(e_\phi(x))}[\Delta(x, d_\theta(e_\phi(N(y))))],$$

where $N(\cdot)$ denotes the noise model discussed in Section 4.2 and Δ denotes word-level cross-entropy loss. $x \sim X$ means that x is drawn uniformly at random from the set of sentences in dataset X. Similarly, the reconstruction loss is given by

$$\mathcal{L}_{rec}(\phi,\theta,X) = \mathbb{E}_{x\sim X}[\Delta(x, d_\theta(e_\phi(N(x))))].$$

The overall loss minimized is then

$$\mathcal{L}(\phi,\theta) = \mathcal{L}_{back}(\phi,\theta,X) + \mathcal{L}_{back}(\phi,\theta,\tilde{X}) + \mathcal{L}_{rec}(\phi,\theta,X) + \mathcal{L}_{rec}(\phi,\theta,\tilde{X}). \tag{4}$$

This is exactly the same loss minimized as in ((Artetxe et al., 2017b).

5.3. Unsupervised variational attention

The setup for training the unsupervised variational model is quite similar to that in the previous section: in particular, it has the same components $\mathcal{L}_{rec}$ and $\mathcal{L}_{rec}$ of the overall loss $\mathcal{L}$, but these components are now given by the appropriate evidence lower bound. For instance, the variational back-translation loss is given from (3) by:

$$\begin{aligned}\mathcal{L}_{var-back}(\phi,\theta,X) \quad = \quad & \mathbb{E}_{x\sim X, y=\tilde{d}_\theta(e_\phi(x))}[\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(y|z)] \\ & -KL(q_\phi(z|x)||p(z))].\end{aligned}$$

The variational reconstruction loss $\mathcal{L}_{var-rec}$ is defined similarly, and the total variational loss $\mathcal{L}_{var}$ is defined as in (4) with respect to the variational back-translation and reconstruction losses.

6. Methods

6.1. ARAE

The ARAE Translator is used in two experiments. The first experiment is on the low resource IWSLT'15 dataset of 133K English and Vietnamese parallel sentences analyzed in (Luong & Manning, 2015). The second experiment is on the medium resource WMT'14 dataset of 4.5M English-German data. The default training, validating, and testing splits are used in both cases. Input and target sentences are lowercased as suggested by (Junbo et al., 2017). Only sentences of length at most 30 are used during training. The vocabulary consists of the 11000 most used words from either input language. The model is trained for 20 epochs on the Vietnamese dataset and for 4 epochs on the German dataset. All other architectural and training hyperparameters are borrowed from (Junbo et al., 2017).

The ability of the classifier p_u to distinguish between the latent spaces of either language is a measure of alignment in the code spaces. We therefore track the classification accuracy at the end of each epoch and display the results in Figure 1. The perplexity of the validation set as measured by the autoencoder is also recorded. The final perplexity is 4.58 for vietnamese and 6.73 for english.

6.2. Unsupervised NMT and Variational Model

Both the unsupervised NMT and unsupervised variational attention model are trained on the Multi30k-Task 1 dataset (Elliott et al., 2016) of captions describing images (the images themselves are ignored). All training datasets contain 29000 sentences. We use the fastText word embeddings computed from Wikipedia (Bojanowski et al., 2017), which we then align using Vecmap (Artetxe et al., 2017a). We report results on the English, French, and German languages (in particular, we train models between English/French and English/German).

Both models were trained using the Adam optimizer with a learning rate of 0.0002. The encoder and decoders in all models were 2-layer bidirectional gated recurrent units, with dropout with probability 0.3 and hidden size of 600. Parameters were initialized uniformly in $[-0.1, 0.1]$.

For the variational models, we disabled the denoising aspect of the encoder so as to determine the effect of the variational setup itself (as a possible substitute for denoising). We moreover disabled input feeding (Luong et al., 2015) for performance reasons and also because of the subtle independence issue discussed in Section 4.3. We also experimented with disabling input feeding for the (plain) unsupervised NMT model.

Depending on perplexity as measured on a validation set, models were trained for approximately 3000-5000 training steps, which amounts to between 10 and 16 epochs. The models took between 2 and 5 hours to train on a GPU, which is 2-4 times faster than the code released by (Artetxe et al., 2017b) (where both implementations are run on the GPU).

Apart from using a validation set for early stopping, we do not perform any parameter tuning due to limited resources, instead relying on the parameters of (Artetxe et al., 2017b). Translations are evaluated using the BLEU metric. [1]

Example ARAE Translations (VI→EN, DE→EN)
Input: sau đây là mt ví d .
Target: so here ’s another example .
Output: so this is a great camera .
Input: zwischen 2006 und 2010 steigen die verkäufe um 26 % und etablieren sich mit einem absatz von UNK millionen stück .
Target: between 2006 and 2010 , sales rose by 25 % to reach 12.7 million units .
Output: between 2006 and 2010 aircraft will be extended opening after 2002 and as quickly as a number of around ten countries .

Table 1. The ARAE for the most part learns to generate sentences of length roughly the length of the input sentence. There is occasional overlap in phrases or words common to both languages such as numbers and punctuation. BLEU scores for the translator are too low to be significant.

7. Results

7.1. ARAE

Example sentences translated by the ARAE translator are shown in Table 1. In addition, we sample twice from a normal distribution and decode the interpolations with both decoders. These decodings are shown in Table 4. If the latent spaces were aligned semantically, this procedure would result in a sequence of similar parallel sentence pairs. Instead, we find that the model generates sentences in both languages of similar length but lacking shared meaning.

7.2. Unsupervised NMT and Unsupervised Variational Attention

In this section we report results of our unsupervised NMT and variational attention models on the validation set of the Multi30k dataset.[2] Table 2 shows the results of the unsupervised NMT model in (Lample et al., 2018a) on the Multi30k dataset, as well as the performance of their model in a supervised manner. (Artetxe et al., 2017b) did not report results on Multi30k.

In Table 3 we report results of our unsupervised NMT and variational attention models. As variational encoder-decoder models often experience the problem of the gradients from the KL divergence

[1] As is standard, we use the Perl script at https://raw.githubusercontent.com/moses-smt/mosesdecoder/master/scripts/generic/multi-bleu.perl.

[2] It will be ideal to use the test dataset, which we plan to do in the future.

Model	en-fr	fr-en	en-de	de-en
SUPERVISED	56.83	50.77	38.38	35.16
UNSUPERVISED	32.76	32.07	26.26	22.74

Table 2. BLEU score of (Lample et al., 2018a) models on Multi30k-Task1 test set.

term dominating it is often necessary to decrease the effect of the KL component of the loss. This is certainly problematic for us, as performing gradient descent on the loss $\mathcal{L}_{var-back}$ quickly forces the KL divergences to approach 0, corresponding to "attention" which simply averages evenly over the source hidden states. VAR-ATTN-1 implements KL cost annealing, whereby all KL terms in the loss $\mathcal{L}_{var-back}$ are muliplied by $\tanh(r/5000)$, where r denotes the training iteration (we run for a total of 5000 iterations). VAR-ATTN-2 does not have a KL cost annealing term, but rather multiples all KL terms in the loss by 0.01 throughout training. We note that our application of variational encoders-decoders is slightly different from the standard approach (Bahuleyan et al., 2017; Bowman et al., 2015), in that our reconstruction task on a single language is trivial, but rather we hope to be able to force the KL divergence to be small enough so that our encoder-decoder is robust to noise, but not too small, so that the model can still use nontrivial (i.e. non-uniform) attention.

Model	en-fr	fr-en	en-de	de-en
INPUT-FEED	10.08	12.4	17.75	21.95
NO-INPUT-FEED	24.84	29.3	–	–
VAR-ATTN-1	11.72	12.35	–	–
VAR-ATTN-2	20.01	23.61	–	–

Table 3. BLEU score of our models on Multi30k-Task 1 validation set. Models trained with fixed fastText word embeddings.

We also experimented with fixing the embeddings to both the fastText vectors mentioned in the previous section, as well as the WacKy (Baroni et al., 2009) vectors used in (Artetxe et al., 2017b). We hypothesized that one potential reason for the greatly improved neural machine translation BLEU scores in (Lample et al., 2018b) over the unsupervised NMT model in (Artetxe et al., 2017b) (which is very similar) is that the fixed embeddings in (Lample et al., 2018b) are trained on the Wikipedia/fastText dataset, whereas (Artetxe et al., 2017b) uses WacKy embeddings. It was shown in (Conneau et al., 2018) that the former word embeddings lead to much superior word-level translations. However, both settings of these embeddings led to similar performances on the Multi30k dataset, as is shown in Figure 4 (in the appendix).

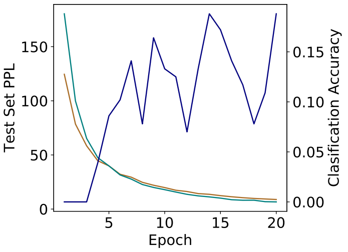

Figure 1. ARAE validation loss during training. The perplexity as measured by the autoencoder for the vietnamese and english validation sets are plotted in green and brown respectively on the left y-axis. The accuracy of the classifier p_u on the hidden states of the encoder for sentences from the validation sets is shown on the right y-axis. The PPL is comparable to the literature experiment on the SNLI (Junbo et al., 2017).

8. Discussion and Conclusion

8.1. ARAE

The failure of the ARAE Translator shows that even if two languages share a latent space, the semantics of the latent space do not necessarily align.

8.2. Unsupervised NMT

Encouragingly, we are able to achieve BLEU scores on Multi30k-Task 1 that are comparable (only a few BLEU points lower, for translations to English) to the results presented in (Lample et al., 2018a). We hypothesize that our BLEU scores are lower due to two factors: (1) insufficient hyperparameter tuning (we did none, using the parameters of (Artetxe et al., 2017b), which were supposedly tuned for the much larger WMT dataset), and (2) insufficient vocabulary sizes. Regarding the latter point, we lower-cased all words and replaced any word not in the embeddings dictionary with an OOV symbol. The resulting vocabulary sizes were: 9179 (en), 10613 (de), 10093 (fr). For instance, in the case of German, we had to drop nearly 9000 words from the Multi30k training set, which, combined with the importance of capitalization in German, likely led to the lower BLEU scores reported in Table 3.

The significantly lower en-fr BLEU scores for the model with input feeding is puzzling. These were trained using exactly the same code as the (relatively successful) en-de model with input feeding. To try to determine the reason for this, we have visualized the attention distribution for some sample data points in the appendix; the model is clearly not effectively using attention, but it is unclear why. Our best guess is a bad initialization or local minimum.

8.3. Variational attention

The models VAR-ATTN-1 and VAR-ATTN-2 do manage to learn to some degree, but a closer analysis of the attention distributions produced in either model reveals two distinct modes of failure. For the model VAR-ATTN-1, in which the coefficient on all KL losses is increased from 0 to $\tanh(1) \approx 0.8$ throughout training, the KL divergences are all driven to 0, and the attention distributions produced are all approximately uniform, as shown in Figure 2. On the other hand, in, the model VAR-ATTN-2, the multiplier 0.01 on all KL divergences is evidently too small, as shown by the very sharp attention distributions in Figure 2. Such sharp distributions correspond to little to no noise being added to the latent varaible $z^{(i)}$ when calculating $\log p_\theta(y_t^{(i)}|z^{(i)})$ in (3), so this model seems to be roughly equivalent to simply disabling noise in the denoising auto-encoder of (Artetxe et al., 2017b) (recall that for the variational attention models we do not add noise to the inputs to the encoder).

Future work should experiment further with parameter tuning in the VAR-ATTN-1 and VAR-ATTN-2 models above. Moreover, since we generally only want the encoder and decoder to be robust to *local* noise, enforcing a *global* uniform prior seems to be too strict. Therefore, we have implemented a *local variational attention encoder-decoder*, which uses the local-m attention of (Luong et al., 2015) (with a window-width of 5) for the posterior, and forces the prior to be uniform over each local window. Unfortunately, we did not have time to test this model, but it is a promising approach for future work in this area.

Moreover, it would be informative to compare the performance of the VAR-ATTN-2 model with simply turning off denoising in the (Artetxe et al., 2017b) model.

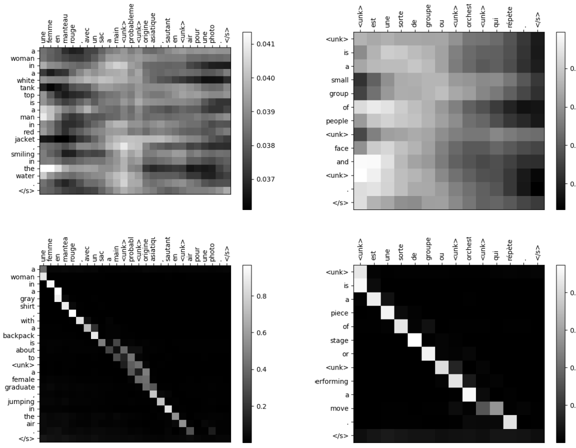

Figure 2. Attention distributions, given by the expected values of α_{ts} for the corresponding posterior distributions. Top: VAR-ATTN-1; bottom: VAR-ATTN-2.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *ArXiv e-prints*, January 2017.

Artetxe, Mikel, Labaka, Gorka, and Agirre, Eneko. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 451–462, Vancouver, Canada, July 2017a. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1042`.

Artetxe, Mikel, Labaka, Gorka, Agirre, Eneko, and Cho, Kyunghyun. Unsupervised Neural Machine Translation. *arXiv:1710.11041 [cs]*, October 2017b. URL `http://arxiv.org/abs/1710.11041`. arXiv: 1710.11041.

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, September 2014. URL `http://arxiv.org/abs/1409.0473`. arXiv: 1409.0473.

Bahuleyan, Hareesh, Mou, Lili, Vechtomova, Olga, and Poupart, Pascal. Variational Attention for Sequence-to-Sequence Models. pp. 8, 2017.

Baroni, Marco, Bernardini, Silvia, Ferraresi, Adriano, and Zanchetta, Eros. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, September 2009. ISSN 1574-0218. doi: 10.1007/s10579-009-9081-4. URL `https://doi.org/10.1007/s10579-009-9081-4`.

Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, and Mikolov, Tomas. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M., Jozefowicz, Rafal, and Bengio, Samy. Generating Sentences from a Continuous Space. *arXiv:1511.06349 [cs]*, November 2015. URL `http://arxiv.org/abs/1511.06349`. arXiv: 1511.06349.

Conneau, Alexis, Lample, Guillaume, Ranzato, Marc'Aurelio, Denoyer, Ludovic, and Jegou, Herve. Word translation without parallel data. pp. 14, 2018.

Elliott, Desmond, Frank, Stella, Sima'an, Khalil, and Specia, Lucia. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pp. 70–74. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-3210. URL `http://www.aclweb.org/anthology/W16-3210`.

Junbo, Zhao, Kim, Y., Zhang, K., Rush, A. M., and LeCun, Y. Adversarially Regularized Autoencoders for Generating Discrete Structures. *ArXiv e-prints*, June 2017.

Lample, Guillaume, Denoyer, Ludovic, and Ranzato, Marc'Aurelio. Unsupervised Machine Translation Using Monolingual Corpora Only. pp. 12, 2018a.

Lample, Guillaume, Ott, Myle, Conneau, Alexis, Denoyer, Ludovic, and Ranzato, Marc'Aurelio. Phrase-Based & Neural Unsupervised Machine Translation. *arXiv:1804.07755 [cs]*, April 2018b. URL `http://arxiv.org/abs/1804.07755`. arXiv: 1804.07755.

Luong, Minh-Thang and Manning, Christopher D. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, 2015.

Luong, Minh-Thang, Pham, Hieu, and Manning, Christopher D. Effective Approaches to Attention-based Neural Machine Translation. *arXiv:1508.04025 [cs]*, August 2015. URL `http://arxiv.org/abs/1508.04025`. arXiv: 1508.04025.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Zhang, Biao, Xiong, Deyi, su, jinsong, Duan, Hong, and Zhang, Min. Variational Neural Machine Translation. pp. 521–530. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1050. URL `http://aclweb.org/anthology/D16-1050`.

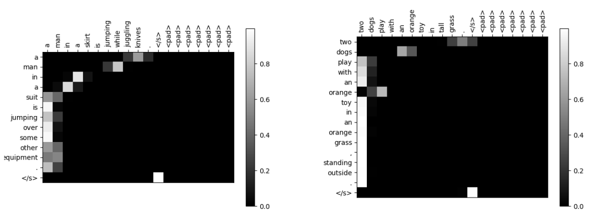

Figure 3. Attention visualizations for fr-en model with input feeding (and BLEU score of 12.4). The reason for the model's failure to use attention successfully, whereas the same model for other language pairs was able to do so, remains a mystery.

A. ARAE Parallel Interpolation table

See Table 4.

B. Attention visualization for fr-en model with input feeding

See Figure 3

C. Learning curves for word embedding comparison

See Figure 4.

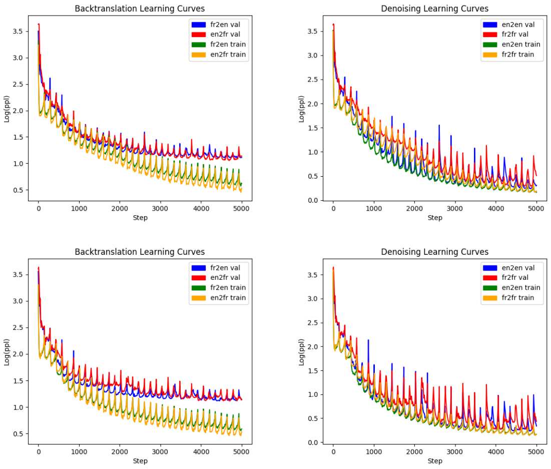

Figure 4. Training and validation average negative log-likelihoods (i.e. average cross-entropy, or log-perplexity) on English-French datasets. Training curves represent cross-entropies for the back-translation loss. Validation curves represent cross-entropies evaluated on a parallel dataset (this is technically not allowed in the setting of unsupervised NMT, but we did not use these data for parameter tuning). Note the cyclical nature of the learning curves is due to the fact that for each epoch, batches are presented in increasing order of length. Top: word embeddings fixed to fastText. Bottom: WacKy word embeddings. Logarithms are computed base 10.

ARAE Parallel Interpolation
EN Decoded: and this thing happens from all . VI Decoded: nhng nu tôi hi bn là mt câu tr li có nghĩa . G(VI Decoded): but if I ask you a meaningful answer.
EN Decoded: and this thing I came out . VI Decoded: nu bn xem xét nó có mt chút nào v nhà . G(VI Decoded): if you consider it a little home.
EN Decoded: and I actually see a lot of what happened when I came . VI Decoded: khi bn b b qua mt chút na . G(VI Decoded): when you're a little skeptical.
EN Decoded: and there 's a whole picture of the same time I got over . VI Decoded: các nhà này b mt nhiu hn so vi mi ngày . G(VI Decoded): these homes lost more than every day.
EN Decoded: and there 's a first picture of the last year – it was a few times . VI Decoded: các nhà này khp ni ni các nc nghèo đói nghèo . G(VI Decoded): these homes are everywhere in poorer countries.

Table 4. We sample $x_1, x_2 \sim \mathcal{N}(0, I)$ and apply the universal encoder to get h_1, h_2 =enc$_\phi(x_1)$, enc$_\phi(x_1)$. We then apply the decoder p^l_ψ for each language l to $h_1 + t(h_2 - h_1)$ for each $t \in \{0, 1/5, ..., 1\}$. For understanding, the translation of the vietnamese sentence is included in table above by $G(\cdot)$. We find that while the latent spaces of both languages appear to be smooth, they do not overlap semantically. For instance, in the bottom two examples, both english and vietnamese decodings preserve the first two words (”these homes“ and ”các nhà“) of the decoding and alter what follows.