Machine Translation for Low Resource Languages

Siddhartha. Saxena, (150719)
Department of Computer Science and Engg.
IIT Kanpur
Kanpur, India

Jaivardhan Kapoor, (150300)
Department of Electrical Engineering
IIT Kanpur
Kanpur, India

Amrit Singhal, (150092)
Department of Computer Science and Engg.
IIT Kanpur
Kanpur, India

Harshvardhan (150283)
Department of Electrical Engineering
IIT Kanpur
Kanpur, India

Harsh Narang, (150267)
Department of Computer Science and Engg.
IIT Kanpur
Kanpur, India

Shanu Kumar, (150659)
Department of Electrical Engineering
IIT Kanpur
Kanpur, India

Abstract

Much work has been done on Neural Machine Translation, but most of it requires the availability of a large parallel corpora to train on, which is not always readily available. Unsupervised methods also exist that work by encoding both the languages in consideration to a common latent space, and then decoding them to the target language as required. This work aims at an accurate implementation of the work presented in Lample et al. (2017), and train it on a non-parallel corpora of English and Catalan languages. Baseline implementation is evaluated using BLEU score metric. We also hypothesize that adding the GCN layers would optimize the existing representations.

1 Introduction

Machine Translation is a classical problem in Natural Language Processing with vast applicability. Recent surges in the area have been dominated by Deep Neural Networks, like Sutskever et al. (2014) Kalchbrenner & Blunsom (2013). Most of these models are a variation of an encoder-decoder setting, where they encode a source language to some latent representation and then try to generate a translation in the targeted language, through the latent representations. This has become a key paradigm in *Neural Machine Translation*.

After being introduced by Sutskever et al. (2014), the Seq2Seq model has been at the heart of various state of the art models in various NL tasks other than MT, such as question-answering Xiong et al. (2016)and abstractive text summarization Nallapati et al. (2016).

The performance of these neural translation methods have become exceptional these days even surpassing that of a human translator in some cases (Table 9 Wu et al. (2016)) when provided with a large enough parallel corpora and translating isolated sentences is tough. But in cases when we do not have such large corpora i.e. in low-resource languages, neural machine translation models have suffered to produce results.

This is the case for a wide variety of languages such as Basque or even language pairs of popular languages such as German-Russian. This has led to work in unsupervised machine translation, especially with non-parallel corpora such as Lample et al. (2017), Artetxe et al. (2017). These model represent interesting directions of work as they try to learn a representation that is agnostic to the involved languages, hence one can learn translation for low-resource languages by using high-resource languages like French and English. This is akin to the work done in style-transfer in vision Gatys et al. (2015) as there, they try to learn a common representation for images drawn by various artists and then transfer style of one artist to another artist's image.

In this work, we study the work introduced in Lample et al. (2017), and attempt a faithful implementation of the model and algorithm. The subsequent sections deal with the model thus used, delving into some related works that inspire the model. The various components of the model are then explained using appropriate motivations. Section 4 explains the data set used, the training procedure employed and the results thus obtained. Lastly, we conclude the report with a discussion on the results and further augmentations to the model, including a GCN-based autoencoder Bastings et al. (2017) impose structure into the model.

2 Related Works

2.1 NMT with non-parallel corpora

There have been very few previous attempts at developing a general machine translation system with only monolingual corpora and the absence of any other kind of supervision. Monolingual data has been widely used to improve translation quality in semi-supervised scenarios(Irvine & Callison-Burch (2016)). Work on zero-resource machine translation has relied on some kind of external supervision either from language pairs different from the ones being translated(Chen et al. (2017)) or from different modalities(Nakayama & Nishida (2016)). The work of Lee et al. (2016) has focused on developing a common latent representation but on a character-level limiting it to languages having same alphabets.

Work by Conneau et al. (2017a) has achieved state-of-the-art results in word translation without any parallel data. They seek to learn a mapping from the latent space of L_1 to L_2. This is done in 2 steps - first, using adversarial training, in which the discriminator is trained to distinguish between the target embeddings and mapped source embeddings , while the mapping is trained to fool the discriminator, and second, the mapping obtained from adversarial training is fine-tuned to fit the target embedding using the closed-form Procrustes solution with more frequent words as anchor points. Translation is performed by finding nearest neighbors using an unsupervised selection metric. The training, validation and dictionary generation are completely unsupervised. We will use this bilingual dictionary to generate word-by-word sentence translations to initialize our model.

2.2 Style Transfer

Style transfer is a particular technique that has come from Computer Vision. It deals with generating images that are of a given style from a given test image. Hence it also deals with a similar problem on a broad level, of translating from source to a target image.

The technique used is based upon two neural models that are able to effectively extract the style and the spacial features from the image separately. The style image is passed from the style capturing neural net and the test image is passed through the spacial neural net to generate their representations respecitvely. Finally to get style transfered image, a noise image is cycled through both these networks minimizing the gap between the style representations (between noise and style image) and the spatial image (noise and spacial image) hence over multiple iterations the noise image is learnt to be the style transfer image of it.

2.3 Unsupervised Machine Translation

In the next section we shall describe the model and its components, and provide appropriate motivations for each part. The final objective function is

$$\begin{aligned}\mathcal{L}(\theta_{enc}, \theta_{dec}, \mathcal{Z}) =& \lambda_{cd}[\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt, src) + \mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src, tgt)] \\ &+ \lambda_{auto}[\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt)] \\ &+ \lambda_{adv}\mathcal{L}_{adv}(\theta_{enc}, \mathcal{Z}|\theta_D)\end{aligned}$$

The losses are weighed using the hyperparameters, and are computed via cross validation. Cross validation is measured in terms of BLEU scores between the input sentences and 'back-translations' of the translation of input sentences. The model is initialized by the word-by-word translations of sentences generated by Conneau et al. (2017a). Our model tries to incorporate the same principles as this model and hence contains auto encoder , cross domain and adversarial losses as well. The difference between our proposed model and this model lies in the encoder architecture where we use a GCN layer after the LSTM. Further details about are model have been described in the later sections. We will be comparing our results with the word-by-word translations produced by unsupervised dictionaries generated by MUSE and the original model without GCN to observe the changes in BLEU scores which adding structure to the representation provides.

3 The Model

The model proposed in Lample et al. (2017) tries to align the latent representations of a sentence and its translation in the latent space. This core principle advocates a language-independent representation with language-specific encoders and decoders. To incorporate this principle, their architecture tries to ensure two things –the first is to reconstruct a sentence in one language from a noisy version of it and the second is to obtain the correct translation of a sentence given a noisy version of the source as input. The auto-encoder loss ($\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src)$, $\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt)$, one for each language) accounts for the reconstruction in same language. For defining cross domain loss, source sentence is translated into target language, corrupted and then translated back to the source language. This cross-domain loss($\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt, src)$, $\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src, tgt)$, one for each direction) accounts for the error in translation from a corrupted source. Finally a discriminator is trained to identify the language from which the given embedding is generated. The encoder is trained to fool the discriminator which gives rise to the adversarial loss($\mathcal{L}_{adv}(\theta_{enc}, \mathcal{Z}|\theta_D)$), θ_D is the encoder parameter). The encoder used is a bidirectional LSTM where $\mathcal{Z}$ denotes the embeddings for the monolingual data. The encoder and decoder parameters are shared between the languages.

3.1 Problem Definition

We denote the dataset of sentences in the source domain by D_{src} and another dataset in the target domain by D_{tgt}.

Let us denote by $\mathrm{W_S}$ the set of words in the source domain associated with the (learned) words embeddings $\mathrm{Z^S} = (z_1^s, \dots, z_{|\mathrm{W_S}|}^s)$, and by $\mathrm{W_T}$ the set of words in the target domain associated with the words embeddings $\mathrm{Z^T} = (z_1^t, \dots, z_{|\mathrm{W_T}|}^t)$, Z being the set of all the embeddings.

3.2 Model

The model consists of the following parts:

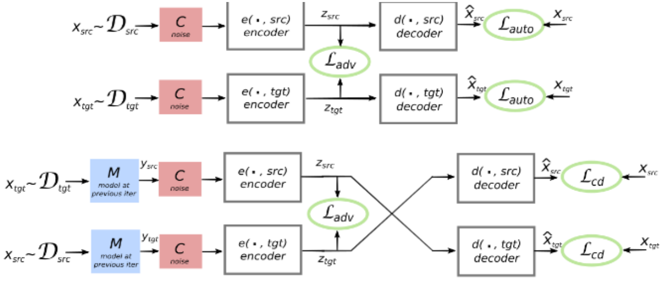
- Encoder for encoding the source sentence to latent space
- Decoder for decoding the target sentence from the latent space
- Discriminator to classify between the encoding of source and target sentences.
- Model starts with an unsupervised translation model obtained by making word-by-word translation of sentences.

Given an input sentence of m words **x** = $(x_1, x_2, \dots, x_m)$ in a particular language l, $l \in \{src, tgt\}$, the encoder $e_{\theta_{enc},\mathrm{Z}}(x, l)$ computes a sequence of m hidden states $\mathbf{z} = (z_1, z_2, \dots, z_m)$ by using the corresponding word embeddings, i.e. $\mathrm{Z_S}$ if $l = src$ and $\mathrm{Z_T}$ if $l = tgt$; the other parameters θ_{enc} and θ_{dec} are instead shared between the source and target languages.

The decoder $d_{\theta_{dec}, \mathrm{Z}}(g, l)$ generates output sequence $\mathbf{y} = (y_1, y_2, \ldots, y_k)$, , where each word y_i is in the corresponding vocabulary W^{l}.

The following figure demonstrates the losses and model structure.

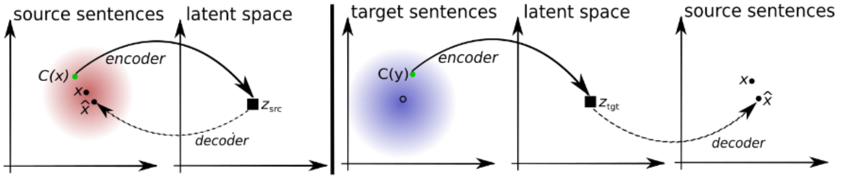
Figure 1: Illustration of proposed architecture and training losses



3.2.1 Denoising Auto-encoding

The input sentence is reconstructed from its noisy version. The noisy output is obtained by dropping and swapping words. Without this component the encoder/decoder quickly learns to merely copy words. The figure below shows the process.

Figure 2: Illustration of proposed architecture and training losses



3.2.2 Cross Domain Training

The input sentence is reconstructed from the noisy version of the translated sentence. This is done to improve the mapping of the input sentence from the source language to target language.

3.2.3 Adversarial Training

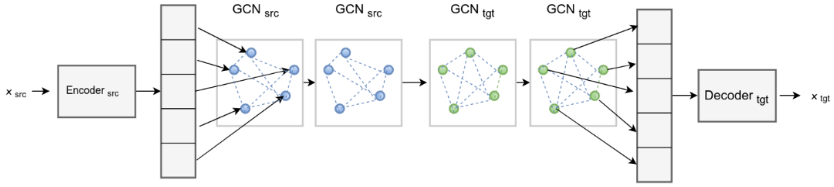
We constrain the model to encode both source language and target language sentences in the same latent space. The discriminator has to classify between the encoding of source sentences and the encoding of target sentences, whereas the encoder is trained to fool the discriminator.

3.3 Graph autoencoder: possible extension

To incorporate structure into the latent space in the form of a graph with connections between various features, we use a graph autoencoder(figure below), which is a special case of message passing networks. The graph autoencoder is inserted between the current encoder and decoder, and as a result the latent space then consists of a learnt graph. The encoder followed by the graph layer encodes source and target sentences to a structured latent space, and then decoder's graph layer followed by the decoder decodes them from the structured latent space to the source or the target domain.

The encoder's graph layer takes the output of encoder as input nodes in its graph, denoted by $H^{(0)}_{enc} = \mathbf{z}$ and outputs a structured vector $H^{(n)}_{enc}$ in the source language latent space, where n is the number of layers of the GCN. The decoder's graph layer takes $H^{(n)}_{enc}$ as input nodes in its graph, denoted by $H^{(0)}_{dec} = H^{(n)}_{enc}$ and outputs a structured vector $H^{(n)}_{dec} = g$ in the target language latent

Figure 3: Model Architecture with structured latent space using GCN



space. Our focus is to make the latent representations of sentences in both languages similar in the latent space for the output of the GCN encoder.

The total loss function obtained will share stark similarity to the loss described for model in Lample et al. (2017), with a difference that the losses of the GCN layers will be incorporated in the losses associated with the encoder/decoder layers. As the latent representations is now the output of the GCN, the loss functions will now try to obtain an identical representation for a input sentence and its translation in that latent space of GCN output. It is redundant to write fully in terms of the additional GCN layers, as they are automatically computed using automatic differentiation routines in computational frameworks such as TensorFlow and PyTorch.

4 Experiments

4.1 Dataset

We have used the Newstest dataset (New) which contains 20 million monolingual sentences in English and 26 million monolingual sentences in Catalan. We use a fraction of these datasets.

4.2 Initialization

Further, we need to initialize our model with a translation model to obtain cross-domain losses. We use a pair of dictionaries constructed from aligned pre-trained word embeddings trained using the MUSEConneau et al. (2017b) library. We use the embeddings from the MUSE library to keep the training completely unsupervised. The unsupervised dictionaries contain 30k pairs which were obtained from a nearest neighbor metric defined in the MUSE paper.

4.3 Training

In the paper of Lample et al. (2017), around 38 million sentences were present in each monolingual corpora. In comparison, we have trained 3 models with 32k(¡1%), 0.96 million (2.5%) and 1.28 million(6%) sentences with training times of 8 hours, 54 hours and 164 hours respectively on a single GPU. We have trained the model taking the same set of hyperparameters specified in the paperLample et al. (2017).

4.4 Parameters

We used three layers of bidirectional LSTM with attention as encoder with RNN size of 300, three layers of bidirectional LSTM as decoder and a discriminator with hidden layer size 1024. We used Adam optimizer for training the model.

4.5 Results

We report the BLEU scores for the model trained on 1.28 million sentences. We have implemented only the word-by-word translation baseline from the paper. Even though the increase in BLEU scores was not far off from those mentioned in the paper, upon manual inspection of the Catalan translations to English using Google Translate, we observed that the model was far away from its optima. The model had learned the structure of the language and could frame grammatically correct sentences but semantically the translations were very different from the input sentences. We attribute this problem to the tiny training corpora, small number of epochs and poor quality of the bilingual dictionary synthesized by MUSE. There were certain methods for improving the translation quality, like using ground-truth dictionaries or adding parallel sentences during training, but these deviate from the baseline of ’completely’ unsupervised machine translation which we wish to demonstrate.

Table 1: BLEU score on the News 2014 datasets consisting of 3000 sentences

BLEU Scores	**English→Catalan→English**	**English→Catalan**
word-by-word	41.6	9.6
Model after 5k batches processed	15.2	10.8
Model after 33k batches processed	16.1	14.4
Model after 136k batches processed	16.5	18.1
Model after 148k batches processed	16.7	17.7
Model after 174k batches processed	17.9	19.0

5 Discussion

The model proposed differs from the model described in Lample et al. (2017) in terms of the encoder and decoder structure. The loss functions change accordingly to include the form of the GCN loss, and we hypothesize that using a GCN layer will greatly optimize our learned representation in the latent space. It is, however, worth noting that the added complexity may render the training more unstable and chaotic, and the model may fail to converge to a well-enough parameter configuration that correctly utilizes the advantageous properties that GCNs offer. As shown in Bastings et al. (2017), syntactic structure in the form of relations between different words in a sentence greatly increases the modeling power and unsupervised learning capacity of the network.

Intuitively, adding structure in unsupervised setting should increase accuracy than that of supervised settings since any additional information is more valuable when existing information is scarce. Our model may slightly differ from the former in that it automatically provides the GCN layer the desired graph structure through transformations by previous layers in the encoder, instead of imposing a pre-defined linguistic structure-like syntax. Whether or not this is a viable approach to utilize GCNs, and whether the GCN input must be constrained to some structure, remains to be tested.

A recurring theme in neural machine translation is the large amount of resources in terms of time and computing power it takes for the model to converge meaningfully. The dataset we are planning to train and test the model on may be too small for sufficient convergence of the model, as the paper we derive much of our model from clearly states that they used a dataset an order of magnitude larger than ours. It is thus imperative that we attempt to train the model such that all losses are sufficiently decreasing in tandem. Due to the use of multiple interconnected losses, which also include adversarial losses, this is a daunting task, and a challenging engineering problem.

As we were unable to find a substantial parallel English-Catalan sentence corpora, we have assumed the Google Translate translations as the ground truth. This is a very big assumption since Google Translate is a translation model.

Due to the long training times we cannot present the results of training the model with GCN layer.Reducing training time is imperative if we want to perform realistic evaluations on this model. The addition of the GCN layer should add more structure to the latent space representation but there remain many critical design decisions for GCN layer like the number of nodes, adjacency matrix and transformation of LSTM output to GCN nodes, asymmetry in the GCN structure for source and target languages, which would require months to tune if we go by the current training times. Also, the improvements due to addition of the GCN layer over the baseline might not be obvious if the models haven’t converged. Future work should first focus on speeding up the training time. Effect of addition of parallel data in form of both sentences and dictionaries needs to be investigated. Effect of GCN layer addition may well be observed if the training times are reasonable for this semi-supervised setting. The one deficit which this paper has is the lack of structure in language modeling. GCN is one such method to add structure. Exploring other architectures which can add a specific structure to the model might improve the translation quality.

6 Contributions

Sidharth - 21% Shanu - 21% Amrit - 21% Harshvardhan - 21% Jaivardhan -11% Harsh Narang - 5 %

References

Newstest2013. `http://www.statmt.org/wmt14/translation-task.html`.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*, 2017.

Joost Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. *CoRR*, abs/1704.04675, 2017. URL `http://arxiv.org/abs/1704.04675`.

Yun Chen, Yang Liu, Yong Cheng, and Victor O. K. Li. A teacher-student framework for zero-resource neural machine translation. *CoRR*, abs/1705.00753, 2017. URL `http://arxiv.org/abs/1705.00753`.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017a. URL `http://arxiv.org/abs/1710.04087`.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017b.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL `http://arxiv.org/abs/1508.06576`.

Ann Irvine and Chris Callison-Burch. End-to-end statistical machine translation with zero or small parallel texts. *Natural Language Engineering*, 22:517–548, 2016.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1700–1709, 2013.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043, 2017. URL `http://arxiv.org/abs/1711.00043`.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. Fully character-level neural machine translation without explicit segmentation. *CoRR*, abs/1610.03017, 2016. URL `http://arxiv.org/abs/1610.03017`.

Hideki Nakayama and Noriki Nishida. Zero-resource machine translation by multimodal encoder-decoder network with multimedia pivot. *CoRR*, abs/1611.04503, 2016. URL `http://arxiv.org/abs/1611.04503`.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL `http://arxiv.org/abs/1609.08144`.

Kun Xiong, Anqi Cui, Zefeng Zhang, and Ming Li. Neural contextual conversation learning with labeled question-answering pairs. *CoRR*, abs/1607.05809, 2016. URL `http://arxiv.org/abs/1607.05809`.