Neural Machine Translation in Low Resource Setting Using Pre-Trained Contextual Embeddings

Sheikh, Shakeel Ahmad
Laboratoire dé Informatique de Grenoble
Grenoble, France
shakeel.sheikh@grenoble-inp.org
Supervised by: Laurent Besacier

Abstract

Deep Learning (DL) is a very powerful technique through which many Natural Language Processing (NLP) tasks solved, have achieved the excellent performance. Neural Machine Translation (NMT) is a new approach to machine translation task. Most of the current-state-of-the-art machine translation systems employ an encoder-decoder approach in which the input sequence is first encoded and based on this encoding, output is generated. The econder and decoder are based on attention mechanism that recombines a fixed encoding of source tokens based on the decoder state. The problem with these encodings is that they use unidirectional embeddings as an input to the encoder which makes it bit hard to generalize on the unseen data. We propose an alteration to this encoder-decoder approach in which rather than feeding input directly to the encoder we first feed it to the Bidirectional Encoder Representations from Transformers (BERT) model to generate the new representations of the text which will be then fed to the encoder to make it easier for translation model and we also investigated the effectiveness of the use of byte pair encodings (BPE) in NMT tasks.

1 Introduction

Artificial Intelligence (AI) machines are evaluated for their capacity to emulate human behaviour and according to a classic criterion proposed by Alan Turing, they are said to exhibit intelligent behaviour if they pass the Turing test [1]. If the judge confronted with the output given by a machine and a human cannot tell the machine from the human, the test is said to have passed. A machine translation system is an example of such a machine that emulates the intelligent behaviour of humans. Machine Translation (MT) investigates the use of software instead of humans to translate text or speech from one language to another.

In Natural Language Processing (NLP) tasks a huge impact is made by deep neural networks recently, in particular machine translation [2]. MT aims to find the most probable target language sentence that shares the most similar meaning for the source language sentence [3]. MT is a sequence-to-sequence prediction problem of variable and different source and target sequence length. Most of the contemporary state-of-the-art machine translation models belong to a family of encoders-decoders [4]. A variable-length input source sequence is being fed to the encoder neural network and is encoded into a fixed-length vector representation. This vector representation is taken by the decoder neural network and then outputs a translation from this encoded vector. The basic encoder-decoder model is generally equipped with an attention model which repetitively re-accesses the source sequence during the decoding process [5]. One of the main issues with the encoder-decoder approach is that all the necessary information of a source sequence needs to be compressed by the neural network into a fixed-length vector. The encoder compresses the input sequence into one single vector representation. This makes it difficult for the model to deal with the longer sentences [4].

With the help of Deep Neural Networks (DNNs), the translation quality in MT is improving but is far from the perfect. One of the major challenges in MT is how to efficiently cover most of the vocabulary and how to make use of large-scale monolingual data. The probability distribution computed over the elements in the source sequence is used to aggregate the features into a single "context vector" which is later used for the decoding purposes. The attention mechanism allows the decoder to "look back" into the input sequence and focus on the main positions [5]. The contemporary attention mechanism are generally a simple weighted sum of the input/source representations having limited modeling abilities [4].

We investigate NMT models that operate on the level of pre-trained bert- embeddings and on the level of BPE subword units.

2 Related Work

The encoder-decoder networks are the cardinal deep neural architectures for machine translation, in which the encoder is a recurrent neural network (RNN) based on gated recurrent units which converts the source sequence into its vector representation [5] [6]. Bi-directional(consisting of two RNNs) RNN is also used, in which the final states of these two opposite RNNs are concatenated to form the single input encoding. The biRNNs process the source sequence in opposite di-

rections anhttps://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ecd the result of which is concatenated at the final state. This input encoding is being fed to decoding unit RNN to generate the output sequence in a timely manner. The input and output sequences are processed individually as a one dimensional sequence by the encoder and decoder respectively in the recurrent models [5]. The attention mechanism acts as an interface between the encoder and the decoder and were introduced by [4]. From the input sequence, the attention models find which hidden states are the most important for generating the next target word by evaluating a context vector(a weighted average of input features) [4].

The positional embeddings with the input sequence together with the self-attention replaces the recurrent layers [5]. [7] proposes neural architecture which is relying entirely on attention. The transformer model architecture is purely based on an attention mechanism to draw global dependencies between source and the target. The advantage of attention can be described in three desiderata- complexity per layer, amount of computation that can be parallelized, as measured by the minimum number of sequential operations required and the path length between long-range dependencies in the network. An attention layer combines all positions with a constant number of sequentially executed operations, whereas O(n) sequential operations are required by recurrent layer. The training in transformer model is significantly faster than recurrent based architectures [7]. The attention models have an inductive bias which accesses the source sequence repetitively while decoding process and allows the decoder to focus on salient positions by looking back into source sequence.

The two dimensional convolutional neural networks (2D-CNN) neural MT architecture is a simple model and relies on a single 2D convolutional neural network in the input as well as in the ouptut sequence. Each layer in this 2D-CNN model re-encodes source tokens on the basis of the output sequence produced so far. The properties like attention are thus pervasive right through the network. This model is having fewer parameters and training can be done in parallelized manner . The main drawback of the CNNs is that they have limited history and a fixed size encoding which likely limits the amount of information a model can contain [5].

3 Methodology

Understanding the deixis, semantics of a word, phrase and in addition to the extension of bigger units like sentences, paragraphs, documents is the fundamental pursuit of NLP. In order to let the machines learn to obtain the profound understanding of words or sentences, we require to describe a characterization or representation of the words and documents which the machines can exercise or operate on and which is called "$word\ embeddings$" based on the linguist Firth's approach *You shall know a word by the company it keeps*. How to represent words or documents in a way so as to encode as much information as possible in the context is one of the fundamental challenges in the NLP communihttps://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ecty. The popularization of word embeddings can be ascribed to [8] with the famous Word2Vec embeddiing model. The ongoing research into word representations has been over-topped by these new word embeddings.

Text embeddings is one of the breakthroughs for solving NLP tasks, however it doesn't give the word its dynamic meaning when used in different contexts .e.g "Kashmir" can be the location or name like "University of Kashmir". Contextual string embeddings by [9] captures the semantics of the word based on its context. Pre-trained word representations [10] are a key element in many neural language understanding models. Some of the contextual pre-trained embedding models include ELMO [11] and BERT [12]. BERT a bidirectional encoder representation from transformers is essentially a new way of training language models. Since the pre-trained word representations can be context free like Word2Vec [10] or contextual [11] [12] which means that the same word which appears in different contexts can have different representations.

BERT is a deep pre-trained bidirectional representation's of the context jointly conditioned on both the left and right context [12]. The key technical element in the BERT is applying the bidirectional training of the popular attention language model- transformer [7]. BPE is a simple data compression scheme that repetitively uses a single unused byte and replaces the frequent pairs of bhttps://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ecytes with this single unused in a sentence. Consider an example $aaabdaaabac$. since the byte pair aa is more frequent and thus BPE will replace it with a byte that is not used in the data. Suppose BPE uses Z as replacement. So the new data after application of BPE will look like $ZabdZabac$ with $Z = aa$ [13].

Our approach is not specific to any NMT ahttps://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ecrchitecture, but in this experiment, we follow the NMT architecture by [4], which we briefly explain here. The NMT system is implemented in the form of encoder-decoder neural network with RNNs. The encoder consists of biRNNs with gated recurrent units [14], which reads the input sequence $x = (x_1, x_2, ...x_k)$ and computes respectively the forward and backward sequence of hidden states- $(\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_k})$ and $(\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_k})$,which are later concatenated into one single annotation vector h_i.

The decoder is also a RNN that predicts the target sequence $z = (z_1, z_2, ...z_k)$. Each and every target word z_i is predicted based on the previously predicted word z_{i-1}, context vector c_i (which is weighted sum of annotation vector h_j),and hidden state h_i. The weights of the annotation vector h_j are computed with the help of an alignment model α_{ij}, which is a single layer feedforward neural network, learned jointly with the whole network through backpropagation [4]. The details of the model can be found in the paper [4]. We use parallel English-Romanian corpus to train the model with adam optimizer.

In this work, we demonstrate the effectiveness of byte pair encoding (BPE) and pre-trained contextual embeddings

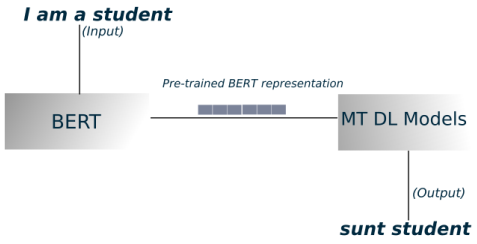

Figure 1: Proposed BERT based NMT architecture

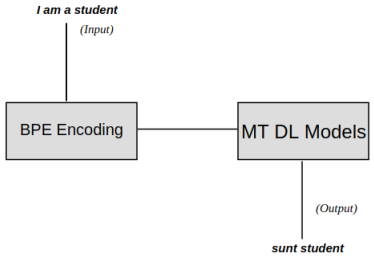

Figure 2: Proposed BPE based NMT architecture

(BERT) on neural machine translation tasks as shown in Fig. 1 and 2.

4 Experimental Results

4.1 Dataset and Preprocessing

MT systems are usually constrained by the size of the vocabulary during training phase. In NMT models, the size of the final softmax layer depends on this vocabulary size which means the bigger the vocabulary size, the more memory and computational power is required during the training phase. Thus, the selection of building this vocabulary in NMT systems have a considerable effect on the system performance. This implies we need to remove the unnecessary words called stop words and this can be done with the help of tokenization. In our word-based baseline and BPE-based NMT, we use Moses tokenizer to get tokens which will be fed into the NMT model, however in case of bert-based NMT, it uses its own tokenizer to tokenize the sentences and applies some BPE-ecodings for the source language (English) and then are fed into the biLSTM NMT model.

We train our model on the standard IWSLT17 English-Romanian (English is source and Romanian is target in our case study) dataset, comprised of 220538 training sentence pairs, 914 validation sentence pairs and 1678 test sentence pairs as shown in Table.1.In this experiment we try to analyze the effect of the the BPE and pre-trained bert embeddings in the NMT. In our baseline standard, each pair of sentences are converted to tokens with Moses tokenizer and based on the one hot encoding, the embedding corresponding to the token are fed into the NMT model from the randomly initialized embedding matrix. In our proposed BPE-based model, we used subword units of $40K$ together on both source and the target language pairs before feeding th data to the NMT encoder. In our proposed bert-based model, we use the pre-trained embeddings of the final layer among the 12 embedding layers of the BERT model for the source language only and feed those directly into the biLSTM encoder. In this experimental setup, we use the *bert-base-uncased* version of the BERT model with 12-layers, 768-hidden, 12-heads, 110M parameters.

Dataset	Sentences	Word Count
Language:	English	English
Training Set	220538	3778360
Dev. Set	914	17444
Test Set	1678	26827
Language:	Romanian	Romanian
Training Set	220538	3778360
Dev. Set	914	17444
Test Set	1678	26827

Table 1: English-Romanian Dataset

Hparam	Value
Encoder Layers	1
Decoder Layers	1
Batch Size	16
Optim	Adam
β(learning rate)	0.001
global attentioan	MLP
epochs	8

Table 2: Hyperparameters

4.2 Parameters

The experimental framework we used in this project is the pytorch version of OpenNMT [15] with some additional parameters as shown in Table. 2

To evaluate the quality of our NMT models, we compute the BLEU score using multi-bleu.pl [1]. For each training run, we use a batch size of 16 for our models.

4.3 Comparison of NMT Models

Perplexity in NMT is the evaluation of how well a model predicts a target word. A low perplexity indicates the NMT model is good at predicting the target word. We first compare wordbased biLSTM with the current state-of-the-art NMT models[2] and BPE-based biLSTM NMT models in terms of training perplexity and BLEU score. Due to the less training

[1] https://github.com/OpenNMT/OpenNMT-py/blob/4fbfbf4fb0f35b4cf056f10631838448d6b65ecd/tools/multi-bleu.perl

[2] https://paperswithcode.com/paper/a-convolutional-encoder-model-for-neural

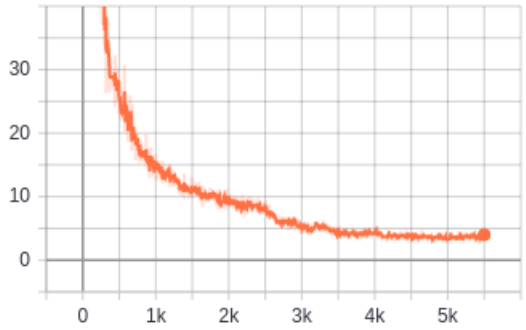

Figure 3: Perplexity curve of wordbased NMT model

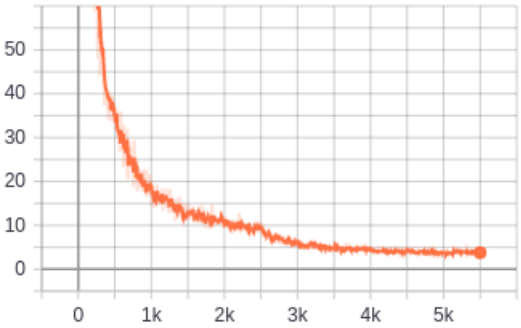

Figure 4: Perplexity curve of BPE-based NMT model

than the current state-of-the-art NMT models, we achieved slightly lower BLEU score as can be shown in the Table.3. The BPE-based NMT model can outperform baseline word-based NMT model as can be carefully analyzed from the Fig.4 and Fig. 3, the learning curve of BPE-based biLSTM NMT model is better than that of baseline wordbased NMT model. This can also be verified from the Table. 3 that the BLEU score of BPE-based NMT model is higher than that of baseline wordbased NMT model. With bert-based NMT, we achieved poor results because we got lot of unknowns in the output translation. The reason being the learning curve as shown in Fig. 5 is much higher in bert-based NMT model than other two NMT architectures. Since we are using only 8 epochs for our case study and the bert is a very large embedding model, so we can hypothesize that the bert-based NMT model requires more number of epochs and some fine tuning during the training phase. Due to some technical problems [3], we didn't get enough time to perform the required experiments and we didn't achieve the expected results as was supposed at the beginning of the project.

5 Conclusion

The application of DNNs to MT is a hot research topic. MT systems using a single approach fails to achieve the desired performance due to its inconsistency and inflexibility for large scale applications. In the contemporary, MT has an important role in various applications such as document translation, communications, customer management etc. In this work, we show how bilingual models perform in terms of overall translation quality. Our analysis compares word-base biLSTM with the pre-trained bert based embedding and BPE-based biLSTM NMT model. In future, the BERT model will be fine tuned for the BPE-based input sequence in order to generate the new representation which will be fed to the NMT model. In the future, We also want to explore how our modified models can be used to translate across multiple language pairs. We plan to apply pervasive based attention models to other tasks in near future. We plan to extend and modify these models to problems involving large inputs and outputs such as audios and videos. In this case study, I got an opportunity to learn and explore neural network models applied in NMT tasks and the two main frame works *PyTorch* and *OpenNMT-py* (PyTorch Version)[15; 17]. We have also added a pre-trained bert embeddings feature to the *OpenNMT-py* library and the code can be pulled from our $github$ repo **??**

[3]The systems we were allotted were working poorly and there were technical problems at regular intervals of time during our internship time period due to which, a very good amount of time had been wasted and we were not able to perform the expected experiments

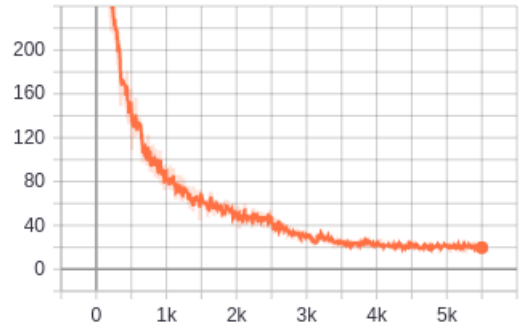

Figure 5: Perplexity curve of BERT-based NMT model

NMT Models	Performance (BLEU Score)
Base System (best SOTA[16])	27.5
word-based biLSTM	23.10
BPE-based biLSTM	23.20
Bert-based biLSTM	3.39*

where [16] is the reference of the paper that report this result

Table 3: Comparison with the state-of-the-art NMT Models

References

[1] A. M. TURING. I.Computing Machinery And Intelligence. *Mind*, LIX(236):433–460, 10 1950.

[2] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[3] Jiajun Zhang, Chengqing Zong, et al. Deep neural networks in machine translation: An overview. 2015.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction. *arXiv preprint arXiv:1808.03867*, 2018.

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

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[9] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[11] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Wikipedia contributors. Byte pair encoding — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Byte_pair_encoding&oldid=901142789`, 2019. [Online; accessed 12-June-2019].

[14] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[15] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017.

[16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.