NLU Assignment 2: Neural Machine Translation

Aditay Tripathi
IISc / Bengaluru
aditayt@iisc.ac.in

1 Neural Machine Translation

Machine translation is the task of translating a sentence x in one language to a sentence y in another language. It can modelled as calculating the $argmax_y P(y|x)$. In neural machine translator the probability $P(y|x)$ is modelled as a deep neural network. A special type of deep neural network called Seq2Seq (Sutskever et al., 2014) model is used for machine translation. Seq2seq model has 2 RNNs which work in encoder and decoder setup. Seq2seq model is a general neural network model and is suitable for machine translation task. It is a conditional Language Model(Mikolov et al., 2010) and it models the language translation as follows:

$$P(y|x) = P(y_1|x)P(y_2|y_1, x)...P(y_n|y_1, ..y_{n-1}, x), \quad (1)$$

in this case, the language model context is all the previous words and the input sentence. The whole input sentence is encoded into a vector and that is passed to the decoder RNN as shown in the fig. 1. In statistical machine translation(Brown et al., 1993), alignment between source and target sentence is required which is hard to obtain. However in seq2seq models, this requirement is lifted because we are encoding the whole input sentence to a vector and that is passed to a conditional LM to get the target sentence. But it requires a large corpus to train the Seq2Seq models. The vanilla Seq2seq model for NMT has been modified to allow to get the alignment between the source and target sentences as a result of training. The modifications has also improved the translation performance considerably. Vanilla seq2seq has a bottleneck in the sense that a vector is used to represent the whole input sequence which may limit the information transfer to the decoder network. Attention network provides decoder, links to the encoder at each time step which alleviates the information bottleneck caused by the vanilla encoder. One such attention network is shown in fig. 2.

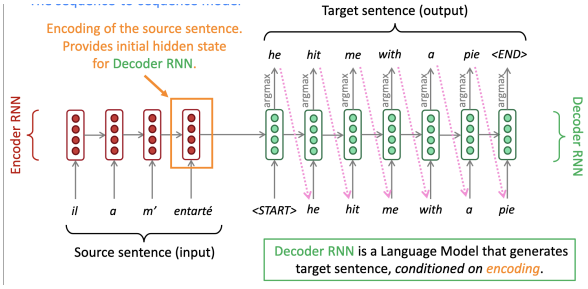

Figure 1: Source RNN encodes the source sentence. Decoder RNN acts as RNN that produces the target sentence. This image is taken from CS224n "Machine Translation, Seq2Seq and Attention" slide.

Given a query vector(decoder state), attention can be thought of as a process to get the weighted sum of the encoder states based on the query vector. There are various types of attention mechanisms. Some of them are listed below:

1.1 Attention Mechanisms

Consider the encoder hidden states at each time step $h_1, h_2, ..., h_N \epsilon R^{d_1}$ and a decoder state(query) at a time step $s \epsilon R^{d_2}$ attention involves the following three steps:

- Computing the attention scores $e \, \epsilon \, R^N$
- Get the attention distribution $\alpha = Softmax(e)$
- Using α to get the weighted sum of encoder states:

$$a = \sum_{i=1}^{N} \alpha_i h_i \quad (2)$$

1.1.1 Dot product attention

$$e_i = s^T h_i \quad (3)$$

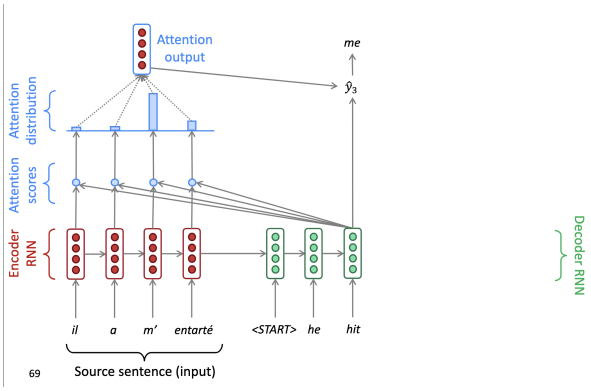

Figure 2: At each time step decoder can read the all encoder states and learns which is important. This image is taken from CS224n "Machine Translation, Seq2Seq and Attention" slide.

In this variant, d_1 is assumed to be equal to d_2. This attention is shown in fig. 2. This attention mechanism can be modified to include a scaling factor and can be written as (Vaswani et al., 2017):

$$\alpha = Softmax(\frac{s^T h_i}{\sqrt{d_1}}) \tag{4}$$

1.1.2 Multiplicative Attention(Luong et al., 2015)

$$e_i = s^T W h_i \ \epsilon R \tag{5}$$

W is a $d_1 x d_2$ matrix.

1.1.3 Additive attention(Bahdanau et al., 2014)

$$e_i = v^T tanh(W_1 h_i + W_2 s), \tag{6}$$

where, $W_1 \epsilon R^{d_3 x d_1}$ and $W_2 \epsilon R^{d_3 x d_2}$ and $v \epsilon R^{d_3}$

2 Dataset and preprocessing

For this assignment, the NMT is trained on German-English(WMT14) and English-Hindi(Bojar et al., 2014) datasets. The German-English corpus is combination of Europarl v7, Common Crawl and News Commentary corpuses. It has a total of $4.5M$ parallel sentences. NLTK tokenizer (Loper and Bird, 2002) and moses scripts are used to clean and tokenize the data. Even after all this processing, the vocabulary size for the German corpus is above $1M$. To reduce the vocabulary size a sub-word tokenizer(Sennrich et al., 2015) is used to tokenize the data. It reduces the German vocabulary size to $32K$. Training on whole $4.5M$ sentences is not feasible. Some of the sentences are well above 500 word length which makes the model harder to fit in the limited GPU memory. Few of the paper have trained NMT translater on $4.5M$ corpus and it takes them few days to train on more than 1 GPUs(Sennrich et al., 2015). I trained the model on $180K$ parallel corpus with maximum sentence length of 50. The dataset is divided into 80% train, 10% validation and 10% test sets.

For English-Hindi corpus, the total data has around $200K$ parallel sentences. For the ease of training, maximum sentence length is fixed at 20 which reduces the data size to $74k$ sentences. The vocabulary size for hindi is around $25k$.

Corpus	Attention	BLEU1	BLEU
Eng-Ger	Add	0.256	0.019
Eng-Ger	Mul	0.262	0.021
Eng-Ger	dot	0.306	0.020
Eng-Hin	Add	0.397	0.023
Eng-Hin	Mul	0.44	0.027
Eng-Hin	dot	0.475	0.022

Table 1

3 Model description

It is a BiLSTM seq2seq model with 1 layer per ecoder and decoder with 256 hidden units. I have implemented scaled dot product attention, multiplicative attention and additive attention for both the corpus.

4 Results and Discussion

The results of the experiments are in the table 1. The BLEU score calculates the average of 1-gram, 2-gram, 3-gram and 4-gram overlaps. The results are not promising as shown in the table. BlUE score is around 2% for all the models. But there seems to be good 1-gram overlap. The datasets are not curated datasets and the vocabulary size is very large. The datasets need to be trained on very large corpus on largeer model. However that requires significantly large compute resources and time.

References

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

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.