Attention Mechanisms in Neural Machine Translation

Raghu Chittersu
Department of Computation and Data Sciences
Indian Institute of Sciences, Bengaluru
chittersuv@iisc.ac.in

1 Introduction

The problem of machine translation is defined as the generation of a translated sentence given input sentence in another language. Neural Machine Translation is using deep neural models to solve machine translation

Through Attention deep learning models can perform even better. It was shown that attending input sequences to generate context gives more accurate results. The attention mechanism in sequence to sequence models was implemented in multiple flavors starting from Additive attention by Bahdanau et al. (2014), multiplicative and dot product by Luong et al. (2015) and finally Vaswani et al. (2017) used key value and self-attention.

In this assignment, Neural Machine Translated is implemented using Bi-LSTM encoder and decoder model. The effect of various encoder-decoder attention mechanism is studied. Finally, the effect of the addition of self-attention at encoder and decoder is studied. The evaluation is done by BLEU scores.

The core of this model is built using PyTorch and code is available at https://github.com/rv-chittersu/machine-translation.

2 Data

For this assignment, the translation is done from English to German and trained on parallel data from Europarl V7 dataset which is available at WMT14. WMT standard dev and test sets are used for validation and testing.

Due to the resource constraints only sentences with length less than twenty is used. The training data contains 447778 samples and a part of Europarl dataset is kept aside with 49610 pairs is also used for testing apart from the mentioned test set.

3 Model

The model mainly contains following components
(1) Encoder & Decoder
(2) Attention
(3) Self Attention
(4) Prediction

3.1 Encoder & Decoder

The encoder is implemented using standard multi-layer Bi-LSTM Trainable variables are used for initial cell and hidden state for Encoder. The forward and backward states are concatenated and projected to the dimension of hidden state.

Multi-layer forward LSTM is used as Decoder. The projection final cell and hidden states of Encoder is used as input to the decoder.

3.2 Attention

Attention layer generates context vector c_i for current decoder hidden state h_i by based on previous hidden states $s_1, s_2, s_2....s_m$. The current state and each of the previous state is passed through the attention function f_{att} to generate attention vector a_i. The context vector c_i is generated by averaging previous hidden states $\{s_j\}_1^m$ weighted by attention score a_i

$$\mathbf{a_i} = softmax(f_{att}(\mathbf{h_i}, \mathbf{s_i}))$$

$$\mathbf{c_i} = \sum_j \mathbf{a_{ij}} \mathbf{s_j}$$

In this assignment f_{att} is implemented using additive, multiplicative and scaled dot product attention functions.

The additive attention for machine translation is used by Bahdanau et al. (2014). The attention function between current hidden state(h_i) and previous hidden state as following

$$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v}_a{}^\top tanh(\mathbf{W}_a[\mathbf{h}_i; \mathbf{s}_j])$$

In the above formula $\mathbf{W_a}$ and $\mathbf{V_a}$ are parameters.

Multiplicative Attention used in (Luong et al., 2015) uses following approach to calculate attention function

$$f_{att}(h_i, s_j) = h_i^\top \mathbf{W}_a s_j$$

In the above problem W_a is parameter and can be learned during training.

Scaled dot product attention used in (Luong et al., 2015) and (Vaswani et al., 2017) is of following form where d is dimension of hidden units

$$f_{att}(h_i, s_j) = h_i^\top s_j / \sqrt{d}$$

The key value attention used in (Liu and Lapata, 2018) and (Daniluk et al., 2017) uses keys and values extracted from previous hidden states. The keys are used to compute attention distribution a_i where as values are used to compute context vector c_i. To above statement can be represented as.

$$[\mathbf{k_i}; \mathbf{v_i}] = \mathbf{T_a h_i}$$

Where $\mathbf{T_a}$ is hidden layer which can be used to generate keys and values from a hidden state.

then the attention distribution and context vector can be calculated as

$$\mathbf{a_i} = softmax(f_{att}(\mathbf{h_i}, \mathbf{k_i}))$$

$$\mathbf{c_i} = \sum_j \mathbf{a_{ij} v_j}$$

here f_{att} can be any of the previously mentioned attention functions.

3.3 Self Attention

Self-attention can be used to generate richer representation from LSTM hidden states. In this setup, multihead Self Attention is applied on encoder hidden states and previous decoder hidden states separately similar to that of (Vaswani et al., 2017). The encoder-decoder attention previously discussed is applied on top of the self-attention layers output.

Self attention on set of hidden states $h_1, h_2...$ is formulated as below.

$$Q_i = W_q h_i$$

$$K_i = W_k h_i$$

$$V_i = W_v h_i$$

then for each hidden state h_i, the scaled dot product is used to get a context vector c_i from a set of hidden states. This can be done multiple times to get multiple representations for h_i, hence multihead attention. In the end, all the representations for an LSTM output are concatenated

3.4 Prediction

The prediction step takes current decoder output and available context vectors if available and pass it through a single hidden layer to generate distribution on vocabulary. The weights of the hidden layer are updates as part of training.

4 Experiment Setup

4.1 Pre Processing

The first step is to remove the sentences which have more than a configured length(for the experiments fifteen is used as the limit). Then the contractions in the English language is replaced with their expanded counterparts. The sentences are tokenized by the tokenizer specific to language.

Once the sentences are tokenized they have divided to train, dev and validation datasets. The vocabulary is built on the training dataset. Then sentences in training, dev and test set are converted to an encoded format which can be directly fed to models.

4.2 Training

During training for each batch, the encoded source language sentences are given as input to Encoder and encoded destination languages are given to decoder.

The decoder is trained by teacher forcing method. The cross entropy loss is calculated at the prediction layer during each step of the Decoder. The loss is optimized by using Adam optimizer.

In case of validation mode, things remain the same except loss won't be optimized.

4.3 Inference

During inference, the functionality of encoder remains the same. For decoder the initial state is used as input for the first step and later on the previous step prediction is used as the next step input.

All the results generated in the inference step is stored in a file along with actual translations available from test data. Then the BLEU score is calculated using NLTK's corpus_bleu method

Table 1: BLEU Scores for English to German

Model	**BLEU Score** WMT Test Set	**BLEU Score** Europarl Test Set
seq2seq $(baseline)$	4.60	7.31
seq2seq + additive attn on encoder	10.77	17.83
seq2seq + additive attn on encoder and decoder	9.88	17.71
seq2seq + additive attn on encoder and decoder + self-attention	8.39	15.55
seq2seq + multiplicative attn on encoder	10.80	17.68
seq2seq + multiplicative attn on encoder and decoder	10.74	17.70
seq2seq + multiplicative attn on encoder and decoder + self-attention	8.60	15.37
seq2seq + scaled dot product attn on encoder	9.64	17.32
seq2seq + scaled dot product attn on encoder and decoder	9.93	17.16
seq2seq + scaled dot product attn on encoder and decoder + self-attention	8.91	15.72
seq2seq + key value attn on encoder	9.64	16.46
seq2seq + key value attn on encoder and decoder	10.59	17.07
seq2seq + key value attn on encoder and decoder + self-attention	8.22	13.98

(i) seq2seq is Bi-LSTM encoder and forward LSTM decoder and each of them are of two layers (hidden units are of dimension 64).
(ii) Keys and values are of dimensions 48 and 16 respectively.
(iii) In case of self attention, 2 attention heads are used

5 Experiments & Results

Experiments have been done with various kinds of possible attention configurations.

First, a regular seq2seq is used with 2 layered Bi-LSTM encoder and 2 layered forward LSTM as a decoder without attention.

The for each type of attention, following experiment are done (i) attention on encoder outputs, (ii) attention on encoder and decoder outputs and (iii) attention on top of the self-attention layer.

The scores are reported for standard WMT Test set and held out test set from Europarl data.

Results can be found at [1]

6 Analysis

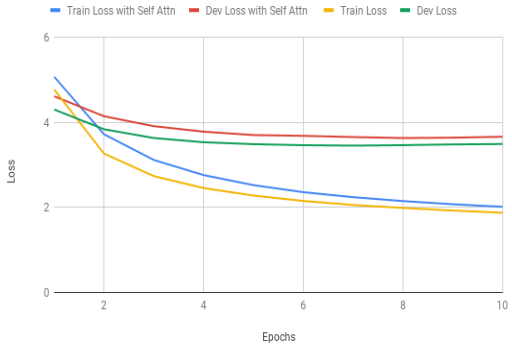
From [1] it can be seen that attention improves the performance of sequence to sequence models many folds.

Addition of Attention over previous decoder hidden states doesn't always result in better performance. It can be seen much clearly on Europarl Test Set as it is closer to the original training set.

Attention on top of representation from Self Attention Layer on top hidden units gave poor results compared to regular attention which is directly applied on hidden states. But it is not far off and performed way better than the baseline model.

The possible reason for the underperformance of self-attention model could be because of their high number of parameters to be trained. Quantitatively for the model mentioned in the results with two Attention Heads has 24,576 more trainable parameters than the attention on encoder and decoder.

So possibly with better hyperparameters and more epochs, it might give as good results as other Attention models.



The above plot further supports the claim as self-attention models reach a plateau early hence

need better hyperparameter turning.

7 conclusion

In this assignment, Neural machine translation is implemented using basic seq2seq to model. Then it was shown that attention improves the performance of sequence to sequence model. Results of various types of attention mechanism are presented. Finally, the effect of self-attention is studied.

In the appendix, a few examples of Machine Translations generated by the model and various attention mechanisms are visualized.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

Micha Daniluk, Tim Rocktschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:6375.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

A Appendices

This appendix provides qualitative results for Neural Machine Translation.

A.1 Sample Translations

For few sentences from test data set translations were generated and presented in this section.

A.2 Attention Visualization

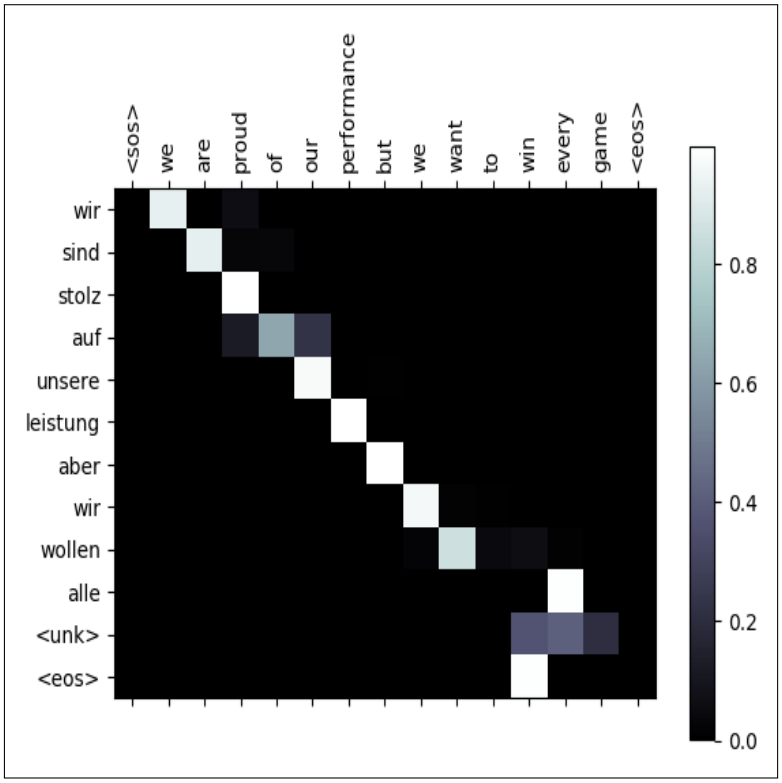
The attention distribution is visualized on the inputs with respect to corresponding hidden states on which attention distribution is computed.

In case of attention on encoder states, the attention distribution is visualized on source input for each output.

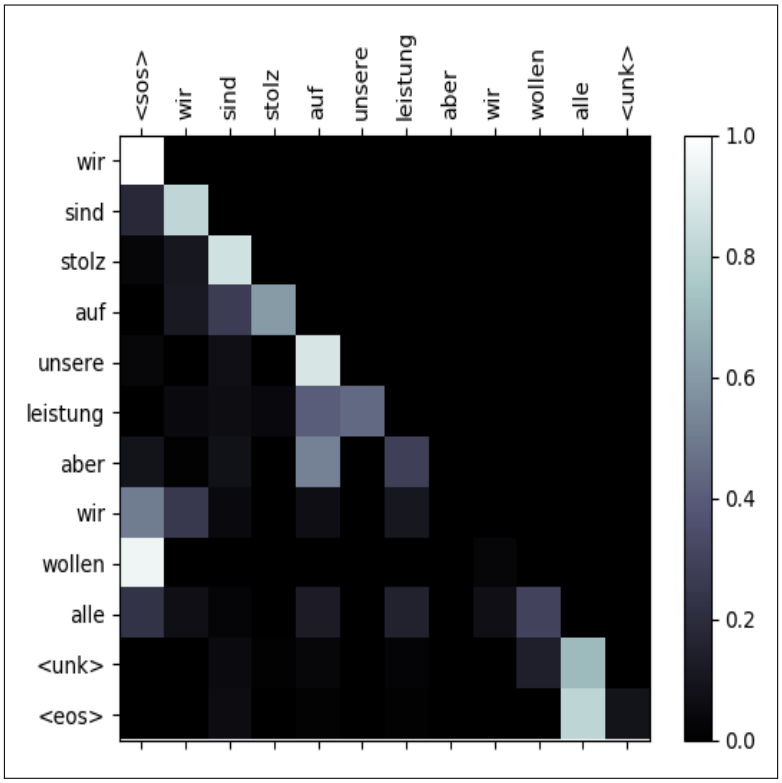
In case of attention on decoder, the attention distribution is visualized on previously generated output for each output.

Source	we are proud of our performance but we want to win every game
Reference	wir sind stolz auf unsere leistung aber wir wollen jedes spiel gewinnen
Hypothesis	wir sind stolz auf unsere leistung aber wir wollen alle UNK
Source	what are the conditions required for the formation of UNK and life
Reference	was sind die bedingungen fr die entstehung von planeten und leben
Hypothesis	was sind die bedingungen fr die bildung von UNK und leben
Source	a departure of UNK from russia is virtually impossible
Reference	eine UNK UNK aus russland ist praktisch ausgeschlossen
Hypothesis	ein UNK UNK aus russland ist praktisch unmglich
Source	i was unhappy but it was really difficult to ask questions
Reference	ich war unglcklich aber es war wirklich schwer fragen zu stellen
Hypothesis	ich war zwar nicht glcklich aber es war wirklich schwer fragen
Source	there will then only be one joint parish council for all six UNK
Reference	es gibt dann nur noch einen gemeinsamen UNK fr alle sechs UNK
Hypothesis	dann wird es dann nur einen gemeinsamen UNK fr alle sechs UNK geben
Source	what are the basic physical laws of the UNK
Reference	was sind die grundlegenden UNK gesetze des UNK
Hypothesis	was sind die grundlegenden physische UNK der UNK
Source	UNK has made transparent the intensive collaboration between us intelligence services and companies
Reference	UNK hat die intensive zusammenarbeit zwischen UNK und unternehmen transparent gemacht
Hypothesis	UNK hat die intensive zusammenarbeit zwischen den usa und unternehmen der UNK erwiesen
Source	the usa have not changed their position regarding the former us secret service employee UNK UNK
Reference	die usa haben ihre haltung zum frheren UNK UNK UNK nicht gendert
Hypothesis	die usa haben ihre haltung nicht gendert die UNK UNK UNK UNK UNK
Source	among other requirements the towns and municipalities applying for the subsidies must levy certain municipal rates
Reference	unter anderem mssen die stdte und gemeinden die zuschsse beantragen bestimmte UNK vorweisen
Hypothesis	unter anderem mssen die stdte und die UNK die UNK und bestimmte UNK UNK

Sample Translations - Source and References are sentences from test set. Hypothesis is sentence generated by model



Visualization of attention distribution on encoder outputs at each decoder step.



Visualization of attention distribution on previous decoder outputs at each decoder step