E1 246: Natural Language Understanding (2019)
Assignment-2: Seq-2-Seq Model for Machine Translation

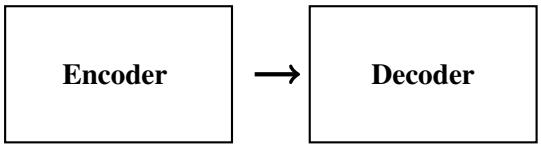
Abhishek Kumar
abhishekkumar@iisc.ac.in

Abstract

This documents contains the final report of 2^{nd} Assignment of Course: E1 $-$ 246 . In this assignment I have build a seq2seq model with attention for the task of machine translation. This document contains the various details,charts and results of seq2seq model with attention.

1 Introduction

Seq2Seq is a method of encoder-decoder based machine translation that maps an input sequence to an output sequence.The core idea is to use RNN based model to implement encoder and decoder.



Seq2seq with various attention mechanisms were trained and evaluated using BLEU score (intrinsic evaluation):
• Model 1 : seq2seq with scaled dot product
• Model 2 : seq2seq with multiplicative
• Model 3 : seq2seq with additive
• Model 4 : seq2seq with key-value attention
• Model 5 : seq2seq with self-attention
The report contains details about the experiments and their evaluation comparison.

2 Datasets

I had used the data from the WMT14 machine translation task : http://www.statmt.org/wmt14.
For training: D1: English-German parallel corpus available at the website (Europarl v7, Common Crawl corpus, News Commentary), and
D2: English-Hindi parallel corpus available at http://ufal.mff.cuni.cz/hindencorp .

2.1 Training and Testing Data Details:

For English to German Task:
• Vocab : Contains Most frequent 50000 Words from the corpus for both English and German words and <UNK> token is used for words not in the vocab but in the corpus.
• Training files contains $\approx 2,20,000$ Sentences
• Validation files contains ≈ 850 Sentences
• Testing files contains $\approx 8,000$ Sentences
For English to Hindi Task:
I have included sentences with no. of words less than 20 only in my training corpus because it was giving out of CUDA storage error for bigger sentences (Maybe because of lack of computational power) • Vocab : Contains Most frequent 50000 Words from the corpus for both English and Hindi words and <UNK> token is used for words not in the vocab but in the corpus.
• Training files contains $\approx 30,000$ Sentences only
• Validation files contains ≈ 520 Sentences
• Testing files contains $\approx 2,600$ Sentences

3 Model

I have used a starter kit for this project which is taken from Stanford CS224n. (**?**)
Each Model is using a single RNN layer only. It uses a bidirectional LSTM with bias for encoder and a unidirectional LSTM with bias for decoder.
Due to computational restriction, Each model is trained with same parameters i.e., each model is compared only for the effects of different attention mechanism on a seq2seq model.
Below table contains the details of the hyperparameter I have used for training each model.
• I have used early stoppage,once the gradient and validation error for a starts saturating.Maximun no. of epoch used to train a model is 15 and min. epoch used is 8 (in case of early stoppage)

Parameter	Value
Batch Size	32
Hidden Size	256
Embedding Dimensions	256
Hyper-parameter	**Value**
Max Epoch	15
Dropout rate	0.3

Table 1: Model Parameters

4 Task: En-De and En-Hn Translation

4.1 Scaled dot-product attention

• In scaled dot attention : The attention score is calculated by performing inner product between encoder-hidden states and decoder-hidden state at timestamp t.
$\Rightarrow$ Attention score = Σ (enc_hidden_state$_\mathbf{i}$ $\cdot$ dec_hidden_state$_t$)
The model with Scaled dot-product attention was trained for 15 epocs.
Training loss during the process initially drops rapidly to the level of 45-30 as seen in Figure 1. After that the loss further keeps dropping at a slow rate and after 14 epochs($\approx 60,000$ iteration) it has further decreased to $\approx$ 10 and it starts saturating.

Similar behaviour is seen for Perplexity, it

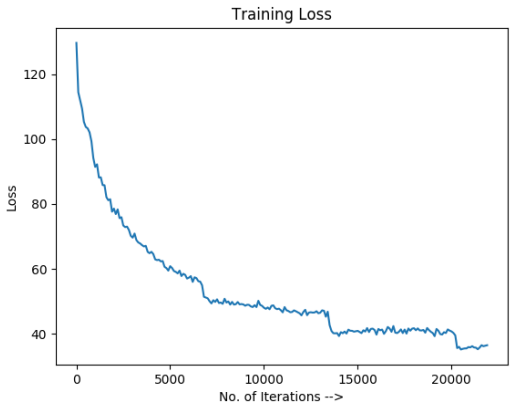


Figure 1: Avg. Training Loss: (Scaled dot-product)

rapidly drops to the level of 600 - 800 in first $\approx$ 200 iteration and after 2000 iteration it drpped to $\approx$ 100 as seen in Figure **??** and further keeps dropping at slow rate.
• BLEU Score : (Eng - de Task : 9.72001 and Eng-hn task $\approx$ 1.079)
$\rightarrow$ Possible reason for low BLEU score for English to hindi task is due to data used for training this model. Training corpus for eng-hn model contains less sentences and also I have removed larger sentences.

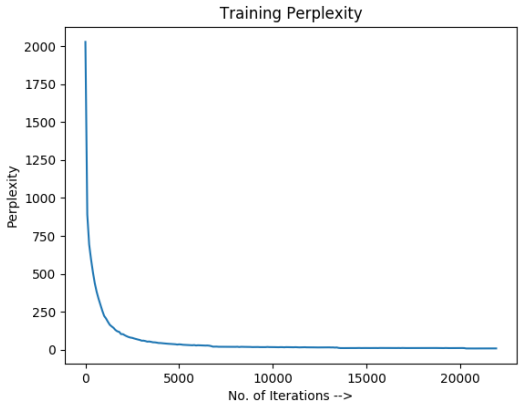


Figure 2: Avg. Perplexity: (Scaled dot-product)

4.2 Multiplicative Attention

In this model I have used the same parameter as above Seq2seq model but for calculation the attention I am using multiplicative attention method.
• Attention Score $e_t = dec_{h_t} \cdot W \cdot enc_h$
• No. of Epochs : 8

Figure 3: Avg. Training Loss (Multiplicative)

From the above avg. training error graph it can be clearly seen that it is following the same trend as of the scale-dot product attention model. After 8 epoch the change in training error as well as validation error became very less i.e., I early stopped the model.
• BLEU Score : (eng - de Task : 19.1048 and eng-hn task $\approx$ 2.13)
Same reason as above for low score of eng-hin model.

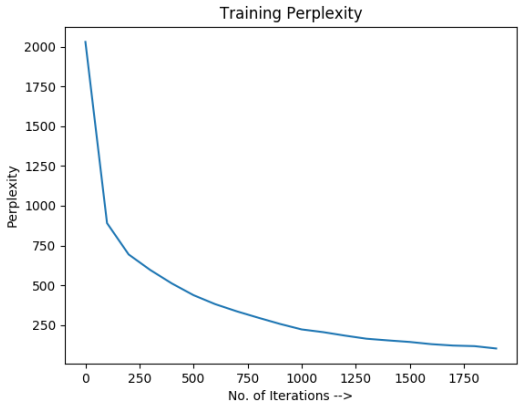


Figure 4: Avg. perplexity (Multiplicative)

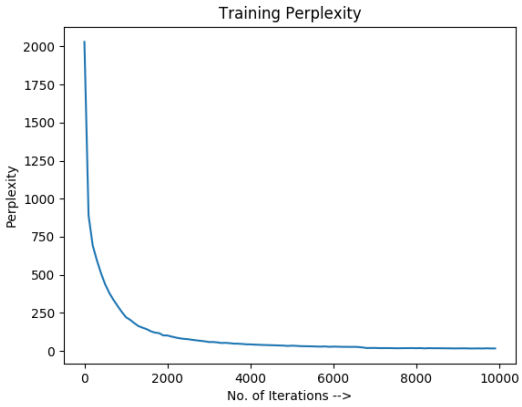


Figure 6: Avg. Perplexity (Additive)

4.3 Additive Attention

In this model I have used the same parameter as above Seq2seq model but for calculation the attention I am using additive attention method.
Attention Score : $e_t = V^T \tanh(W_1 \cdot enc_{hidden} + W_2 \cdot dec_{hidden_t})$
Where V^T, W_1 and W_2 are learned attention parameter.
• No. of Epochs : 7

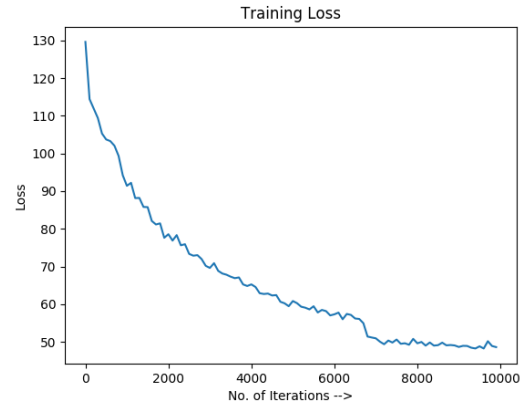


Figure 5: Avg.Training Loss (Additive)

Somewhat similar trend for avg. training loss and perplexity when compared to prev. model.
•BLEU Score : (eng - de Task : 19.3048 and eng-hn task $\approx$ 2.11)

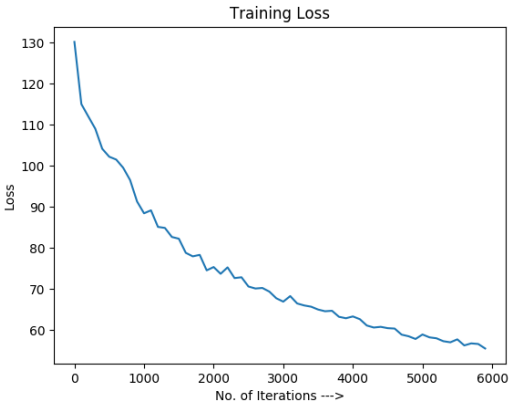


Figure 7: Avg.Training Loss (Key-Value)

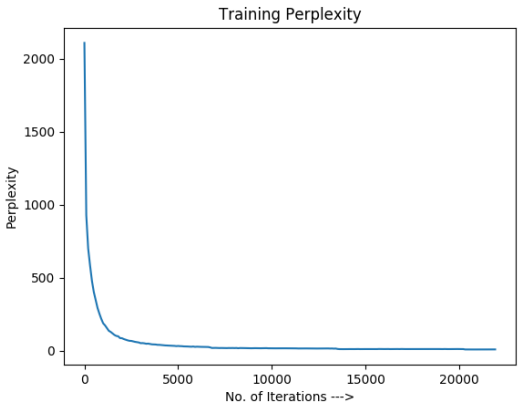


Figure 8: Avg. Perplexity (Key-Value)

4.4 Key Value Attention

In this model I have used the same parameter as above Seq2seq model but for calculation the attention I am using Key value attention method.
• No. of Epochs : 11

•BLEU Score : (eng - de Task : 18.1535 and eng-hn task $\approx$ 1.97)

4.5 Self-Attention

Self-attention was implemented for encoder and decoder for the best performing model among the previous model i.e., additive attention model. The avg. training loss and avg. perplexity follows the similar expected behaviour as earlier models where it initially drops rapidly and then slow down towards end of training.
In this model I have used the same parameter as above Seq2seq model but for calculation the attention I am using additive attention with self attention at encoder side.
To calculate attention score I am using the same method as 2^{nd} model i.e., additive attention model but I am also using self attention at encoder side.
In general we can use self attention at all three sides i.e., encoder, decoder and to calculate final attention score.
• Attention Score:
$e_t(enc_{h_i})$ = V^T tanh $(W \cdot enc_{h_i})$
a = Softmax (enc_{h_i}) = V^T tanh $(W \cdot H^T)$ where H = $[h_0, h_1, ..., h_n]$
$c = H \cdot a^T$
• No. of Epochs : 8

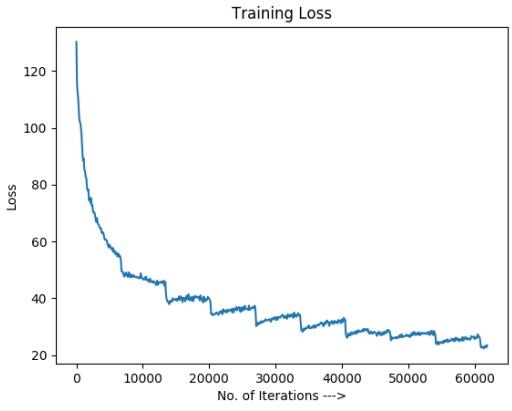


Figure 9: Avg.Training Loss (Self attention)

BLEU Score: (eng - de Task : 20.173 and eng-hn task ≈ 2.21)

5 Model comparision

I have compared each model on the basis of their respective BLEU Score. Model performance can be ranked as follows (en-de translation): Self attention (with additive) > Additive attention > Multiplicative attention > Key - value attention > Scaled-dot product attention.

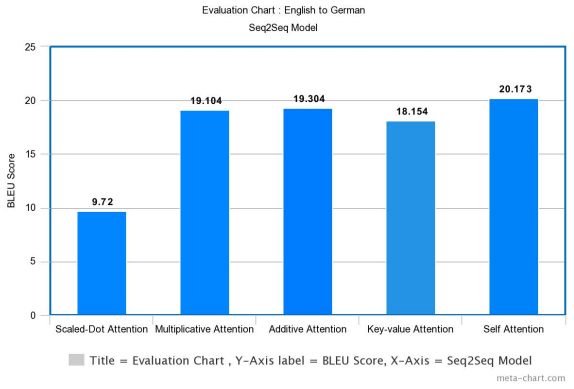


Figure 10: Avg.Training Loss (Self attention)

6 Conclusion

Seq2seq model with attention is a very good model to build translation machine. Neural Model can easily outperform traditional SMT model.
Self attention is a very impressive technique to improve the accuracy and performance of the NMT model.
English to Hindi NMT model are performing poor because of the training file that I am using for training.

7 Refrences

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[2] Luong, Thang, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015.
[3] Vaswani, Ashish, et al. "Attention is all you need." Advances in Neural Information Processing Systems. 2017.
[4] Liu, Yang, and Mirella Lapata. "Learning structured text representations." Transactions of the Association of Computational Linguistics 6 (2018): 63-75.
[5] Stanford CS224N: NLP with deep learning.