Machine Translation using Transformer

Bharghav Lad
Student ID 1117021
bamratbh@lakeheadu.ca

Manik Dhingra
Student ID 1116823
mdhingra@lakeheadu.ca

Shreya Bandyopadhyay
Student ID 1105134
sbandyo1@lakeheadu.ca

Pathikkumar Patel
Student ID 1117477
ppatel73@lakeheadu.ca

Department of Computer Science
Lakehead University

***Abstract*—Machine Translation is a fundamental part of Natural Language Processing research today. This is justified by the fact that at present there are more than 5,000 languages spoken by people throughout the world. Therefore, there needs to be an efficient system that can understand these languages and translate them to a specific language. In this article, we review a specific machine translation model built to overcome the complexities faced by the most commonly used Recurrent Neural Networks and Long Short-Term Memory networks – the Transformer model. We develop the baseline implementation of the Transformer model and study the network to further enhance the performance on the task of machine translation.**

***Index Terms*—Neural Machine Translation, Recurrent Neural Networks, Long Short-Term Memory networks, Transformer, Bilingual Evaluation Understudy (BLEU)**

I. Introduction

The task of machine translation involves conversion of a word, phrase or sentence from one language (source) to another (target). Neural Machine Translation (NMT) has been the most promising approach to solving this problem and addressing the drawbacks of traditional statistical machine translation systems. The reason behind such advancements in sentence-translation from input sequence to target sequence is the use of memory network models, such as the Recurrent Neural Networks (RNN) and the Long Short-Term Memory (LSTM) networks. This improvement is based on the fact that RNNs and LSTMs take into account the context of the sentences, that is they can handle temporal information well, for which a Convolutional Neural Network (CNN) or any feed-forward neural network is inadequate.

For our experiment, we undertake the development of a basic Transformer model and work on analyzing and improving the performance of the model for the task of machine translation. The Transformer model was originally proposed by Vaswani et al., (2017) [1] – the idea behind using this was to eliminate the use of computationally expensive RNNs and LSTMs by employing multi-head attention modules. The proposed Transformer proved to be a viable solution for machine translation problems by achieving state-of-the-art results in terms of BLEU scores and faster training times (mentioned later in the report) for the domain of machine translation. Sections II and III respectively give a literature review and a description of the data-set that we are using for our implementation of the transformer model. Sections IV and V give a detailed explanation about the network structure and the experimental analysis and results obtained respectively.

II. Literature Review

Research around this subject began in the early 1950s with Hutchins and Lovtskii [2]. At the time, a considerable amount of effort was focused on translations between the English and Russian languages due to the cold war. Early implementations of language translation models used bilingual dictionaries which mapped words from a source language X to a target language Y. Nearly three decades of work focused on the development of statistical models by using probabilistic approaches for language translations (Brown et al., (1990) [3].

Due to extensive human component in the feature engineering process in such complex systems, the direction of exploration turned towards the use of neural networks for machine translation after 2014. Development of neural network architectures to improve the state of statistical models was implemented with the help of RNNs by implementing encoders and decoders for better interpretation of sentences (Cho et al., (2014) [4]). Their proposal consists of two RNNs – one to encode the variable-size input in the source language to a fixed-length vector representation, and the other to decode this vector representation back to text sequence, but in the desired target language. The encoder and decoder models are trained in such a way that they tend to improve the conditional probability for the target sequence, given the source text.

Not many years ago, RNNs, LSTMs and Gated Recurrent Units (GRU) based network architectures were the state-of-the-art approach for a sequence-to-sequence (seq2seq) modelling task such as machine translation. The most frequently used type of network, the encoder-decoder, being initially developed for the task of machine translation also proved to be working well for other problems, such as question answering.

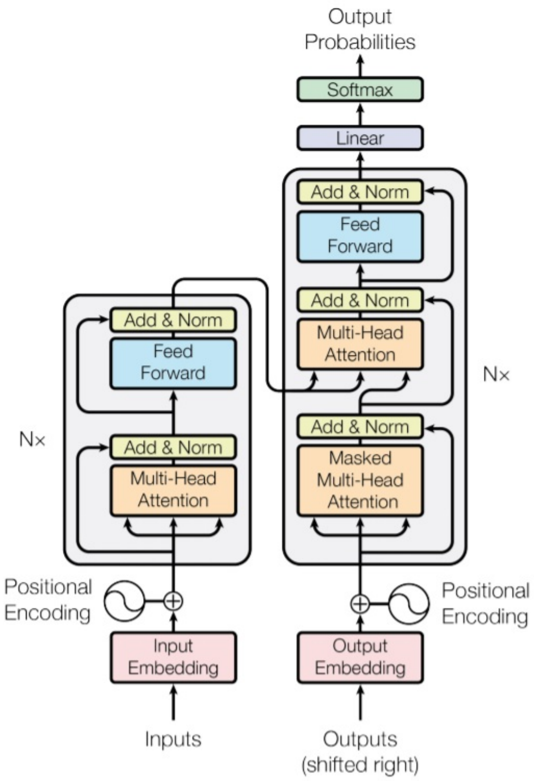


Fig. 1: Transformer Model (referred from Vaswani et al., (2017) [1])

Machine translation systems for a single language-pair have been explored so far. Ideally, a much more effective approach for solving this problem will be developing a single model that can handle several language pair translations. This was attempted by Johnson et al. (2017) [5]. Here, the approach involved the addition of an artificial token for defining the target language at the beginning of each input sequence, rather than designing and training separate architectures for each language-pair. This enabled translations by a single model between more than one language-pairs.

III. Dataset Specification

The Multi30K dataset is ideal for instant work on a broad variety of tasks, namely, but not restricted to, automated image classification, image-sentence ranking, spatial and multilingual semantics, and machine translation. The task of machine translation is usually performed on textual data, such as news reports, the EuroParl corpora, or web-harvested companies (CommonCrawl, Wikipedia, etc.).

The Multi30K dataset enables machine translation to be further established in an environment where multimodal data, such as images or video, are encountered alongside text. The possible benefits of using multimodal machine translation information include the ability to properly answer unclear source text and prevent (untranslated) out-of-vocabulary target language words (Calixto et al., (2012) [6]). Hitschler and Riezler (2016) [7] illustrated the ability of multimodal features in a target-side re-ranking model for the translation. Their methodology is initially trained on massive text-only corpora translation and then finely tuned with a limited volume of data in the domain.

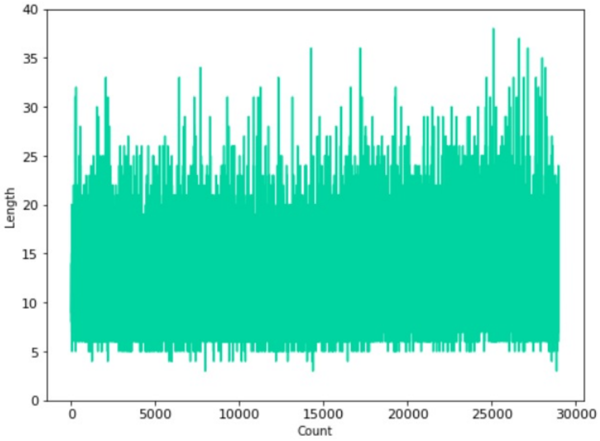


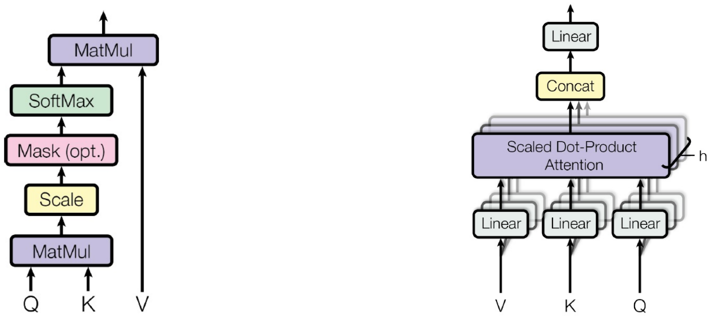
Fig. 2: English Sentences Word Count Distribution

Compared with the Workshop on Machine Translation (WMT) data-set, the Multi30K is relatively small. The corpus has 30K sentences, with an average sentence length of 10-12 words, and maximum and minimum sentence-lengths of 38 and 3 words, respectively. Therefore, models should be able to achieve decent Bilingual Evaluation Understudy (BLEU) scores fast (in several hours). For the scope of this project, the sentences used are in German and English languages – each sentence of the data-set is present in both the given languages. Therefore, the model can either use English sentences to German translations, or vice versa, to predict English translations using German sentences. We use German sentences and train the model to translate them to their English counterparts. The word-count distribution for English sentences can be visualized in Fig. (2).

BLEU score is a ubiquitous metric for evaluation of language translation task performances. In technical terms, it compares a candidate translation of text to one or more reference translations. It is quick and cheap to calculate, language independent, and easy to understand. However, one drawback of using BLEU as a metric is that it favours short translations to produce very high precision scores. In addition to this, it doesn't take into account the meaning of the sentences, i.e. the context.

IV. Network Architecture

A relatively new architecture was proposed by Vaswani et al., (2017) [1] with no recurrent layers, but which mainly consisted of the multi-head attention mechanism (first introduced in 2014 by Dzmitry et al. [8]) and feed-forward layers.



(a) Scaled Dot-Product Attention

(b) Multi-Head Attention

Fig. 3: (left) Scaled Dot-Product Attention, (right) Multi-Head Attention Layer consists of several attention layers running in parallel

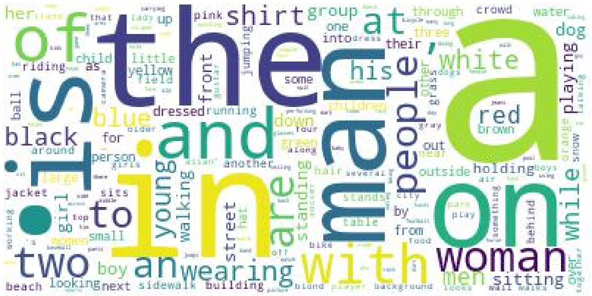


Fig. 4: Word Cloud with Stop Words

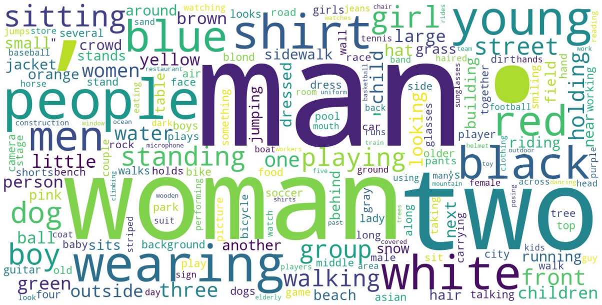


Fig. 5: Word Cloud without Stop Words

The diagram shown in Figure (1) shows the overview of the proposed Transformer model. The network structure for this project is built similar to the given architecture. The section on the left-half of the model is the encoder and the one on the right-half is the decoder of the transformer. The 'inputs' of the encoder are the German sentences, and the 'outputs' entering the decoder will be the English sentences of the Multi30K data-set.

The entire process of sentence-translation can be broken down into five processes: (1) embedding, (2) positional encoding, (3) masking, (4) multi-head attention, and (5) feed-forward.

A. *Embedding*

Word-embedding vectors are one of the hot topics in text representations for NLP. They provide significant improvements over the formerly renowned one-hot encoding vectors in terms of sparsity reduction and better contextual information understanding. When each word is fed into the network, the Embedding layer retrieves its embedding vector, which the model learns to train using gradient descent.

B. *Positional Encoding*

To understand the meaning of a word with respect to the context and its position in the text (sentence) – for the model to make more sense of the sentence – position-specific values are generated, called the positional encoding matrix. It is a constant which when added to the embedding matrix alters it in a position-specific way. These values are generated using the following equations (referred from Vaswani et al. (2017) [1]):

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right) \tag{2}$$

where pos refers to the order in the sentence, i refers to the position along the embedding-vector dimension d_{model}.

C. *Masking*

The intent of creating masks is two-fold: (1) to 'zero' the attention outputs wherever there is just padding, and (2) to prevent the decoder 'peaking' ahead at the rest of the translated sentence when predicting the next word.

D. *Multi-Head Attention*

Calculating attention is fairly straight-forward and Figure (3) explains this. The purpose of using multi-head attention layers is to determine the influence of all the other words in a given sequence on a single word under focus. The attention-mechanism can be described by Eq. (3):

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q is a matrix that contains the query (vector representation of one word in the sequence), K are all the keys (vector representations of all the words in the sequence) and V are the values (vector representations of all words in the sequence). The *softmax* function is applied to have a distribution between $[0, 1]$.

E. Feed-forward Network

The final part of the transformer consists of two linear operations with a *Rectified Linear Unit* (ReLU) [9] activation and a dropout operation in between them. This layer simply deepens the network to better analyze patterns in the attention layer. In order to prevent high variance in values and for faster training and better generalization ability, data normalization is essential.

Learning Rate (LR)	Overfit at Epoch	Minimum Validation Loss	Test-set BLEU Score
0.0001	18	1.693	34.83
0.0003	13	1.658	34.89
0.0005	6	1.617	**36.30**
0.0007	8	1.609	36.23
0.0009	7	1.665	34.28

TABLE I: Comparison for Learning Rate with Test-set BLEU Score

V. Experimental Analysis and Results

The task of machine translation is well implemented using the Transformer model as explained by Vaswani et al., (2017) [1]. The values of the hyper-parameters have been set to those recommended by many researchers for machine translation problems using transformers. As part of data preprocessing, the training data is obtained from a word-tokenizer using a library in python-langauge. Using these generated tokens, the new data-set is created by removing extra spaces and punctuation in the sentences. The experiments run for this project are within the scope of tweaking the learning-rate for optimization as well as the number of hidden neurons to be dropped for the encoder and decoder mechanisms. The comparison of performances by changing the learning-rate can be seen in Fig. (6) and Table I.

The metric used for model performance evaluation is BLEU score and perplexity, where the latter is how well a probability model predicts a sample. Perplexity is directly proportional to the model-loss, which can be explained with Eq.s (4) and (5) as the exponentiation of entropy. It can also be understood as the inverse probability of the test-set, normalized by the number of words (in NLP terms):

$$PP(W) = P(w_1 w_2 ... w_N)^{-\frac{1}{N}} \tag{4}$$

where W is the entire word sequence, $(w_1 ... w_N)$ are the terms in the sequence W and N is the number of words in that sequence. This is equivalent to:

$$PP(W) = 2^{entropy} \tag{5}$$

where $entropy = -\frac{1}{N} \log P(w_1 w_2 ... w_N)$

The best performance of the designed model is obtained by using a learning-rate of 0.0005, that is a BLEU score of 36.30 and a final test-loss of 1.665 and perplexity of 5.284 (Table II).

Learning Rate	BLEU Score	Test Loss	Perplexity (PPL)
0.0005	36.30	1.665	5.284

TABLE II: Final Test Results

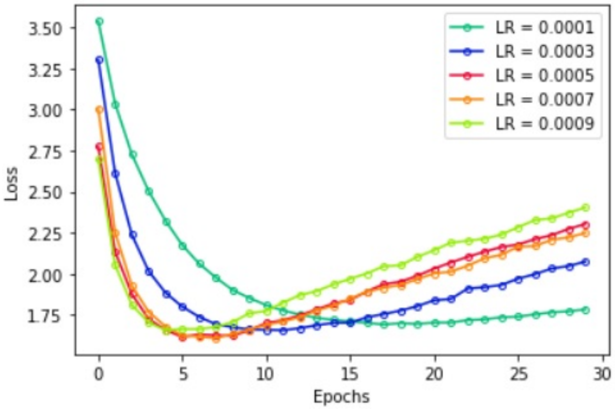


Fig. 6: Loss Performance with Learning Rates

VI. Conclusion

Machine Translation is by far one of the most demanding tasks in Natural Language Processing given the fact that the human language is highly ambiguous and flexible in nature. In this assignment, we implement a bilingual Transformer model (proposed by Vaswani et al., (2017) [1]) in PyTorch environment for the task of machine translation between German and English languages. The applications of machine translation go beyond the scope of translating text or speech between languages. One such use is performing round-trip translations, that is translating from one language to another, and then back to the original language, hence generating a paraphrased representation of the original sentence. Recent advances in the development of language translation systems have been made using Deep Pre-trained Bidirectional Transformers by Devlin et al., (2018) [10] and Crosslingual Language Modelling Pre-training by Lample and Conneau (2019) [11].

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[2] J. Hutchins and E. Lovtskii, "Petr petrovich troyanskii (1894–1950): A forgotten pioneer of mechanical translation," *Machine translation*, vol. 15, no. 3, pp. 187–221, 2000.

[3] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[5] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[6] I. Calixto, T. E. deCampos, and L. Specia, "Images as context in statistical machine translation," 2012.

[7] J. Hitschler, S. Schamoni, and S. Riezler, "Multimodal pivots for image caption translation," *arXiv preprint arXiv:1601.03916*, 2016.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] G. Lample and A. Conneau, "Cross-lingual language model pretraining," *arXiv preprint arXiv:1901.07291*, 2019.

Contributions

Bhargav Lad *(ID 1117021)* 25%
Implementing the transformer model code in PyTorch was one of the tasks undertook by Bhargav Lad. He successfully implemented the transformer model used for the project with the assistance of Pathikkumar Patel. Bhargav also contributed towards the report by providing insights about the model architecture.

Manik Dhingra *(ID 1116823)* 25%
Manik Dhingra was responsible for the algorithm to generate human-readable sentences from the data streams, which was later used for obtaining the BLEU score on the test-set, along with a major contribution in the report for the project in terms of figures, tables and research about the topic of transformers for machine translation. This was done cumulatively by Manik and Shreya.

Shreya Bandyopadhyay *(ID 1105134)* 25%
The tasks of data visualization and hyper-parameter tuning were the responsibilities of Shreya Bandyopadhyay. Furthermore, Shreya contributed majorly towards the report by working together with Manik on the data, figures and tables presented in the report.

Pathikkumar Patel *(ID 1117477)* 25%
Pathikkumar Patel was responsible for testing the implemented model on different data-sets alongside Bhargav in the development of the transformer model. The topics of data-set specification and testing of the model in the report were contributed by Pathik.