Transformer Applications in NLP (A Machine Translation)

SM Habibur Rahman Shabu
Matriculation: 1338315
M.Sc in High Integrity Systems
Frankfurt University of Applied Sciences
Frankfurt, Germany
Email: sm.shabu@stud.fra-uas.de

Abstract—**With the recent progress of Natural Language Processing (NLP) applications are depending on improvement of the performance, more precisely the improvement of deep learning mechanism. Transformer is a architecture based on deep learning is promising a great deal of improvement in the field of NLP. In this paper I tried to provide an appropriate and elaborated anatomy of Transformer and relation with NLP applications as well as implementation of an machine translation example.**

Keywords: Transformer, Natural language processing, machine translation, Neural network, Attention.

1. Introduction

In this technological era, we have come to a point where artificial intelligence and machine learning are being considered as the main component of future enhancement of humankind and the process has already begun. Communication is the key to technological enhancement. The core of human communication is language. By reducing the language barrier, we can enable more opportunity and make the world way closer. The artificial intelligence can be the key factor to reduce language barrier and easier or faster access to different sector and to the whole world. Today the technological improvement already set foot to making our life easier by enhanced machine learning technology like separate way of language translation with great accuracy. Also, very faster and easier for the people to get access to the technology. Even though there are so much scope to improve in this area. Transformer have become a huge factor in many different areas of machine learning in particular for Natural Language Processing (NLP).

To get familiar with the core part of the topic of this paper it is also necessary to understand the key structure of the transformer and the components it is build on. This will help us to get a better sense about the topic.

2. Neural Network

Neural network (NN) or artificial neural networks (ANNs)[7] is a big family which in the end of eighties and the beginning of nineties became very popular, but it became less popular in the end of nineties. Finally, the neural network with deep learning made a big come back in recent years and nowadays the most popular factor in artificial intelligence. One of the main components of the transformer is the Neural network and for that we should focus on understanding it a bit closely. I will try to briefly provide you with the basic concept of the neural network in this part.

Neural network (NN) is one of the part of machine learning which is branch of Artificial intelligence and NN is the core part of deep learning algorithms. As the name shows the connection with human brain and indeed there is a deeper connection with the human brain, The human brain is the inspiration and the structure is built upon the idea of human brain, and the idea is imitating the biological neurons connections or signals with each other. In artificial neural network the nodes are compared to neurons. There are different layers of nodes in a network, one of them is input layer and another one is hidden layer which can be multiple times and the last one is the output layer. Each of the connections of the neurons from previous layers have related weight and threshold. The threshold value decides if the connection is active or not, If the output value of any node is more than the threshold value only then the node become active and provide data to the next layer or else no data get forwarded to the next layer.

Training data are used by the neural network to learn and the accuracy enhances with time. After gaining the proper accuracy with the help of learning algorithms this becomes powerful tool for artificial intelligence. Google search algorithm is the most famous neural network.

What we have seen so far in this section is the structure and idea of the neural network, now we can focus on how the neural networks actually work. Linear regression model which is used to predict the value of a variable based on another variable, combined with input data, weights, threshold value (bias) and the output data are considered as singular node. The composed formula might resemble like this:

$$\sum wixi + bias = w1x1 + w2x2 + w3x3 + bias \qquad (1)$$

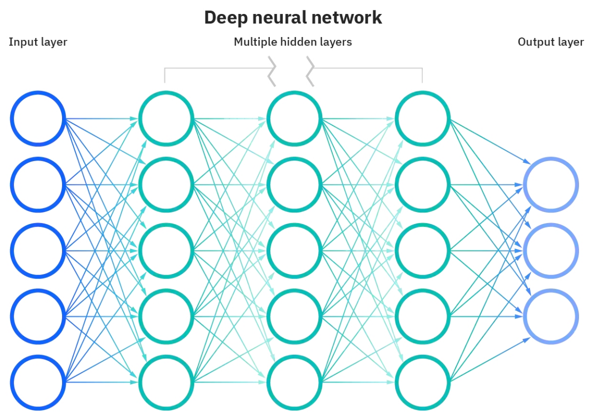

Figure 1. Deep Neural Network structure (IBM Cloud Education 2020)

$$output = f(x) = 1 if \sum \text{w1x1 + b} \geq 0; \quad 0 if 0 > \sum \text{w1x1 + b} \tag{2}$$

weights for each node are assigned after ascertaining input layer. Weights assist to decide which variable are more or less important, Bigger one indicate remarkable contribution to output whereas the other one indicates the opposite. Based on summation of all the input multiplied with weights the activation function determines whether to active the node by comparing with threshold value and the result become input for the next layer. This procedure of passing data through layers is known as feedforward network in neural network. Feedforward network is one of the important aspects of training data with transformer. Most used Neural network is feedforward network but there are other popular one, Feedforward network called Recurrent Neural Network (RNN) when the data go backward from output to input and memorize or remember the previous inputs in memory. In this paper we will focus one feedforward network in transformer structure for machine translation.

3. Natural language processing (NLP)

Natural language processing (NLP)[6] is part of computer science, more specifically included in artificial intelligence(Figure 2). NLP is something that provide computers ability to understand and process languages in exact term a human being does, In particular the text and voice of language. To understand the meaning of text and voice data, NLP uses different technologies for instance statistical analysis, machine learning or different deep learning models which actually work as rule-based modeling of actual human language. When we talk about understanding the meaning of text or voice it means that the intension or sentimental perspective of the writer or speaker is considered, and the purpose is to do this with the help of NLP so that computer can do what human being can do. Most common applications are translating from one language to another, summarization of large amount of text, voice command responses etc. This process enables a large amount of improvement in quality of life, business, and productivity in many sectors.

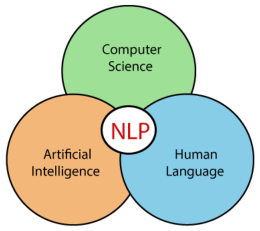

Figure 2. Area of NLP (javatpoint.com/nlp)

Human language is complex, full of ambiguity. There are so much irregularity and variations in any human language which makes the work of NLP very complex. To make sense and able to accurately understand and see through the texts the NLP must be properly taught by the scientist from the very beginning. There are several task which are important for make sense of texts or voice data some of which are given below:

Speech recognition: As we speak each of us sound different and there are so much uncelar sound with so many grammatical mistakes. It is most unorganized when people speak. To convert speech to text the main challenge is to overcome the unclear voice, different accent, talking speed etc.

Grammatical tagging: From text part of speech distinctly identify word or part of the text which is the process of grammatical tagging.

Distinguishing sense of word: In text there can be more than one meaning of a word which is ambiguity and to make disambiguation it is necessary to distinguish the meaning or sense of which the word use used in a sentence.

Sentimental Evaluation: Tries to identify subjective characteristics from text, such as attitudes, humor, emotions, and doubt.

3.1. Methods and techniques

Python and tools: There are various available tools and libraries of python programming which has broad range of usefulness in the NLP. There is also natural language toolkit (NLTK) which is available on open-source platform for everyone, and this includes wide range of libraries, programs and learning materials. The NLTK resources used for all these NLP task as well as finding root level of task and reasoning with methodological conclusion of the text.

3.2. NLP Transformation

Earlier NLP applications were rule based system which did not had the capability to handle large amount of text data and could only produce certain application based on NLP task. The main problem with hand coded application was huge volume of voice and text data. With the introduction of machine learning and deep learning in NLP are providing much more learning capabilities along with large amount of data. Being able to process and train with large amount of data in deep learning it is providing such automatic classification, extraction and labeling the elements to create meaning of the text and voice data. The most common technique of deep learning is Neural network or recurrent neural network (RNN) which introduce more sense and learning capability from enormous text and vocal data collections (raw, unlabeled, and unstructured).

3.3. Fields of NLP applications

There are several area of computer technology uses the NLP and few popular one are discussed below:

Translation tool: modern translation machine is bigger than just replacing word from one language to another. Google translate is the most popular application of NLP. As the translation is much more than replacing word the machines are capable of providing meaningful and keeping the sentimental part alive after translation. It is not always perfect but the amount of accuracy or understandability that produce from such translation machine are very powerful and useful. In this paper our target is to show the whole technique of machine translation using the NLP with transformer technique.

Detection of Spam: Fishing emails or text can be a nightmare and people may not think of it detection of spam as NLP application but detecting fishing text uses NLP technology which does the classification to identify the fishing mails based on the language or text of the email. There are several indicators for an email to be spam, such as mis spelled company name or using too much financial phrases etc. It is considered that the spam detection works almost as per expectation based on expert opinion but there is still some scope for argue against the perfection of spam detection technology.

Virtual assistant and chatbot: Voice control or automation is now becoming very useful and reliable technology. Many of the online platform are using virtual chat assistant to solve customer problems. So far it is not perfect yet very useful. For basic help in any customer care a chatbot is very helpful, at least up to a level after that the machine can transfer the responsibility to human agent. Another life changing invention is virtual assistant, for instant Siri, Alexa or google assistant. These virtual assistants can communicate like a real person and help with many tasks to make life easier. The communication or understanding of human language or providing appropriate response, the NLP technology is used. Both chatbots and virtual assistant works both ways, first understand the instruction or questions based on text or voice to text and provide a response corresponding to the provided instruction or questions.

4. Machine Translation

The most popular NLP application is machine translation application and here I will provide a detail information about the process and elements of an NLP example application with transformer which perform German to English translation. Bellow the whole process and components will be described step by step.

4.1. The Transformer

Transformer is largely utilized in the disciplines of computer vision (CV) and natural language processing (NLP). Transformers are meant to analyze sequential input data, like natural language similar to recurrent neural networks (RNNs) or LSTMs. Translation and text summarization tasks are such applications of transformer. Transformers made two significant contributions that were described in the 2017 paper[3] ”Attention Is All You Need.” First, they made it feasible to analyze whole sequences in parallel, enabling sequential deep learning models like RNN to increase their speed and capacity at previously unheard-of rates. The increased attention component to the sequence mechanism is what advances the transformer. To understand why transformer advances the available best network such as LSTMs, first let’s look at the LSTMs Network.

4.1.1. LSTMs (Long Short Term Memory). Long Short Term Memory Networks (LSTMs) [1] are an unique type of RNN that can recognize long-term dependencies. They were first presented by Hochreiter Schmidhuber (1997), and several authors developed and popularized them with their research works. They are currently frequently utilized and perform incredibly well when applied to a wide range of issues. Intentionally, LSTMs are created to prevent the long-term reliance issue. The network doesn’t strain to learn; rather, remembering knowledge for extended periods of time is generally their default tendency. As with standard RNNs, LSTMs also feature a chain-like structure, but the repeating module is built differently in LSTMs. There are four neural network layers instead of just one, and they interact in a very unique way. The layers are shown in yellow boxes and the pink circles depict operations that use points, such as vector addition in figure 3. In this paper we will just provide an brief understanding of LSTMs so that we can understand why transformer is state of the art so far.

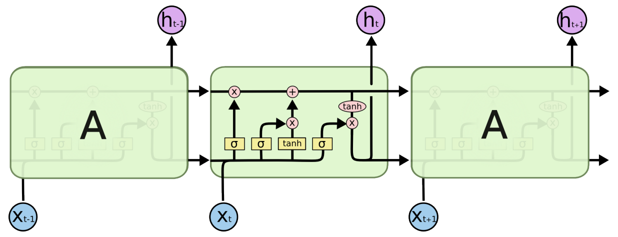

Figure 3. LSTMs Structure (colah’s blog 2015)

Let’s consider text translation and in figure 3 the X_t’s are the input words from text. The arrow from previous module shows the dependency of current translating word from the previous one and for that the computation has to be sequential.

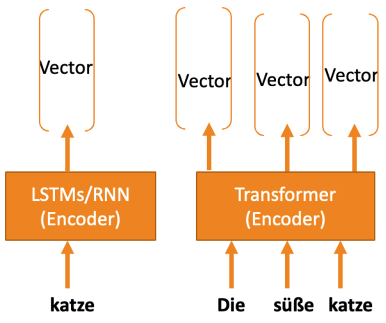

Figure 4. LSTMs vs Transformer's input mechanism

For that today's fast computing GPU's proper utilization is not possible. This is where the transformer come in handy. For train model the transformer provided a way to overcome this dependency and perform better than the LSTMs. The process of working with chunky for loops ends with transformer and it allows the whole sentence to enter the model in parallel (figure4). The time minimization opens the door for adding more layer to the model.

4.1.2. Transformer model. First of all, let us have a look at the structure of transformer model (figure 5) which contains two broader parts: encoder on the left block and decoder on the right block. The core part of this structure is attention. I will try to explain each of the component next and how these can work as a whole and step by step[9][10].

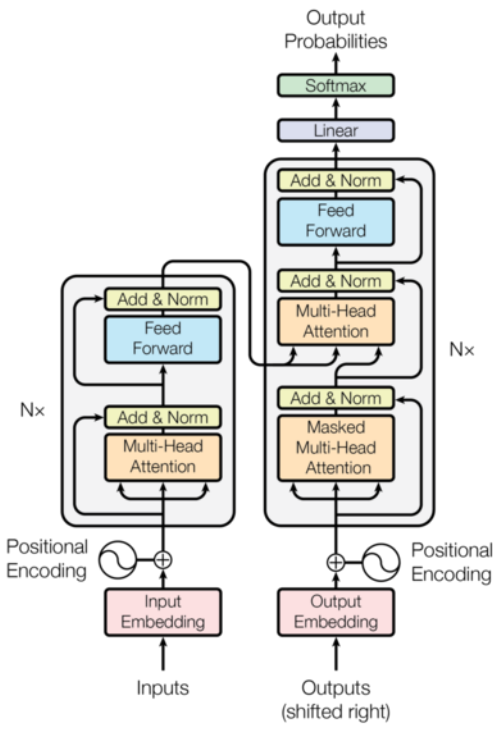

Figure 5. The Transformer model [3]

To execute this approach, we actually need to comprehend six processes:

- **The Inputs Embedding**
- **The Positional Encodings**
- **Masking**
- **The Multi-Head Attention**
- **The Feed-Forward layer**
- **Normalization**

Embedding: The first step is to feed the network with help of word embedding layer. The computer does not understand human words, it understands number, it gets vector and matrices! The embedding layer turn word into learned vector. The vector represents corresponding input word [Figure 4]. Since neural networks learn through numbers, each word is represented by a vector with continuous values.

The Positional Encodings: To understand or make sense of a sentence it is necessary to know the meaning of the words and the position of that word in a sentence. After embedding the words into vector we get the meaning and we should provide something to the input to let the network know about the position of the word. In the paper "Attention Is All You Need."[3] they have used the function mentioned below to provide positional information to the network.

$$PE_{(pos)} = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \quad (3)$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \quad (4)$$

We will not go that much into the mathematical factor but will provide a brief description of these two function. Based on input vector index it creates positional vector with either sin or cosine function. When the input vector index is even the function provide positional vector with sin function and for odd index vector use cosine function to compute. As a result, the embedded input vector is added with the positional information and then feed the network.

Masking: Masking plays the role to prevent unwanted attention provided to the output and specifically stop the decoder to look ahead of already translated sentence while predicting next word [2]. In another word the masking makes use of the word until the current word that are predicting and does not consider the words that are not predicted yet. The numbers wherever the input is ahead will not be able to make a contribution when computing the outcomes after we subsequently impose mask to the attention scores.

Multi-Head Attention: After we get the embedded values of input with positional information then we begin processing the layers of our transformer model. Answers like what portion of the input the model should concentrate on are provided by attention. In Figure 6 The multi-headed attention [3] layer is portrayed. Key, value, and query are abbreviated as V, K, and Q which are the term related to how attention layer work and the attention function presented as converting these terms into an output. V, K, and Q are

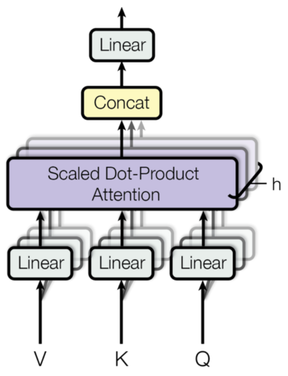

Figure 6. Multi-Head Attention consists of multiple parallel attention layers [3]

abstract vector that extract different component of an input word. For Multi-Head Attention there are multiple weight matrices. Computing of output is done based on weighted sum of the values.

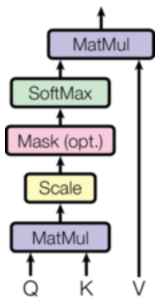

Figure 7. Scaled Dot-Product Attention [3]

For calculating the attention, the equation is below:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad (5)$$

Diagram in Figure 7 provide a detail explanation of each part or step of the equation. Each arrow represents a different aspect of the equation in the illustration of Figure 7. For larger input value softmax function become sensitive and for that learning can be slow or even stopped. To scale the dot product scaling is required and the scaling is done with the help of embedding dimension d_k[8].

The Feed-Forward layer: the next layer of transformer model is Feed-Forward layer. Earlier we have seen the detail explanation of neural network and more specifically Feed-Forward neural network which is the main component in this layer. What Feed-Forward network does in transformer model is it deepen the network and apply linear layers to investigate patterns in the output of attention vector.

Normalization: Normalization [4] in deep neural network contribute to reduce training time by normalize the functions of neurons. In transformer model normalization stops the layers' range of values from drastically changing.

Encoder and Decoder workflow: As we are doing German to English translation, the inputs to the encoder will be the German sentence, and the 'Outputs' entering the decoder will be the English sentence. Above we have learned details about all the components representing transformer model and they are combined with the Encoder and Decoder block (Figure 5). In this section we will see how these components come together in these two blocks and perform the transformer model to predict words. First let us have a look at the encoder block where the German words are feed as input with embedding and positional encoding the positional vector go through the multi-head attention layers to provide an attention vector. With attention vector the feedforward layer refined output which give more meaning to the words and make the attention vector useful for next step in decoder block.

The decoder block also has the same structure as encoder, but it has another extra layer before making connection with the encoder block. In Decoder block feed the output words in our case English words. As encoder the English words also go through as embedding and positional vector to an attention layer. In this layer the masking is necessary so that the output words do meaningful contribution. Output vector of this attention layer now provided as an input combined with the output of decoder block to the next attention layer in decoder block. This second attention layer is where the magic happens, in this layer both Input and output words, in this case German and English are mapped with attention vector to get the best possible translating word. Again, we use the feed forward layer to transform the attention vector to a digestible form for next layer. The next layer is Linear layer which indeed another feedforward network which used to expand the dimension into the number of words in the English language. Furthermore the softmax layer transform it into a probability distribution which is then become human interpretable. And the final word we get is the word which correspond to the highest probability. Overall, this decoder predict the next word. We execute this for multiple time stamp and at the same time Nx number of times which is for both encoder and decoder. (Figure 5). This continues until the end of the sentence and we get the whole translated sentence afterwards.

4.2. Implementation

4.2.1. Tools and libraries. To implement German to English machine translation the python programing language was used. Python is easy to learn as well as has a large number of libraries and tools based on NLP. We have used some of these libraries and the important ones are given below:

PyTorch: PyTorch is a python programing language based opensource framework of machine learning. Pytorch is very useful for our purpose, as for our machine translation we will use the build in transformer model in pyTorch framework.

Spacy: Spacy is a tool for creating information extraction, finding meaning or making sense of natural language but in our case spacy used to pre-process text for deep learning. It also tokenize the words from sentences.

Torchtext: This library used for handling dataset. In this implementation Torchtext used because it includes capability to building dataset that can swiftly iterated over to build language translation model.

Dataset: For German (de) to English (en) machine translation the Multi30k [5] database was used. The training, validation and testing data details are given below.

Train:

(en) 29000 sentences, 377534 words, 13.0 words/sentence

(de) 29000 sentences, 360706 words, 12.4 words/sentence

Val: (en) 1014 sentences, 13308 words, 13.1 words/ sentence

(de) 1014 sentences, 12828 words, 12.7 words/ sentence

Test: (en) 1000 sentences, 11376 words, 11.4 words/ sentence

(de) 1000 sentences, 10758 words, 10.8 words/ sentence

4.2.2. Disscussion. Training: With the dataset and as described above the code was done with python 3.9. As for environment there was a limitation of the computer that it did not have GPU. For that I have used CPU to train the model. For training it takes very long time. Out of 1000 Epoch only 10 was done after 16 hours of training but still after 16 hours the model was trained pretty well. The evaluation was done after saving every checkpoint. For validation and later for testing a custom string ”ein pferd geht unter einer brücke neben einem boot.” was used. In figure 8 After the first Epoch the translation is very unstable but later with each epoch the sentence started to make sense and got better.

Testing: For evaluating machine translation text, BLEU automatically evaluate as a metric. It evaluates machine translated text with a exquisite reference translation. For this implementation and with 16 hours of training it obtained 31.83 bleu score which is very impressive compared to the score of paper “Attention Is All You Need”[3] which was 28.4 bleu score with 3.5 days of training for English to German translation and on the dataset WMT 2014 Englishto-German translation task which was improvement of over 2 BLEU than the existing best result.

5. Limitation and Future of Transformer

For the implementation of machine translation the hardware is a very big limitation. I have used CPU to train the model instead of a GPU which increase huge amount of time to train but I was able to provide a insightful aproach of building a transformer based application in NLP. For the future of transformer overcoming the hardware limitation will bring much more opportunity to grow not just in NLP but in other domains as well. So far the transformer is seen as language model but it can be utilize in other domain as well.

6. Conclusion

In this paper I have tried to provide a detail understanding and representation of Transformer based application in the field of Natural Language process (NLP). The transformer which is a very recent state of the art invention and the improvement it provides is solely based the attention. For the implementation of translation task of German to English the result showed promising outcome. Furthermore, the comparison was also proved the work was properly executed. In future the work can be extended to other deep learning related work like image or audio processing.

Acknowledgments

I would like to thank my supervisor, Prof. Dr. Jörg Schäfer, Director of Computer Science (Bachelor) at Frankfurt University of Applied Sciences for his warmhearted supports and contributions for organizing the Learning from Data Seminar and valuable guidelines. I would also like to thank my colleagues of this course for their contribution over different topics for enhancing the knowledge related to this course.

References

[1] Hochreiter, Sepp Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[2] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

[3] 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 Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016

[5] Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German Image Descriptions. In Proceedings of the 5th Workshop on Vision and Language, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

[6] IBM Cloud Education, Artificial intelligence (17 August 2020) , https://www.ibm.com/cloud/learn/neural-networks

[7] IBM Cloud Education, Artificial intelligence (2 July 2020) , https://www.ibm.com/cloud/learn/natural-language-processing

[8] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. Massive exploration of neural machine translation architectures. CoRR, abs/1703.03906, 2017.

[9] Samuel Lynn-Evans. ”How to code The Transformer in Pytorch” (Sep 27, 2018). Towards Data Science , https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ecb0ed

[10] Transformers From Scratch (18 Aug 2019), http://peterbloem.nl/blog/transformers