College of Professional Studies

Northeastern University San Jose

MPS Analytics

Course: ALY6020: Predictive Analytics

Assignment:

Module 6 Final Project - Transformers

Submitted on:

Feb 16, 2024

Submitted to:

Prof: BEHZAD AHMADI

Submitted by:

ARCHIT BARUA

NIKSHITA RANGANATHAN

INTRODUCTION

Over the past few years, the domain of Natural Language Processing (NLP) has undergone a significant evolution with the introduction of a model architecture called the Transformer. This report provides an overview of the seminal paper "Attention is All You Need" by Vaswani et al., which introduced the Transformer model and discusses its innovative approach to language processing, including its advantages over previous models like RNNs and LSTMs, and its potential future impact on the field of AI.

NLP HISTORY MILESTONES

Natural Language Processing (NLP) has experienced a rapid evolution over the past two decades. This evolution is marked by significant milestones that have shaped the way we enable computers to understand and generate human language.

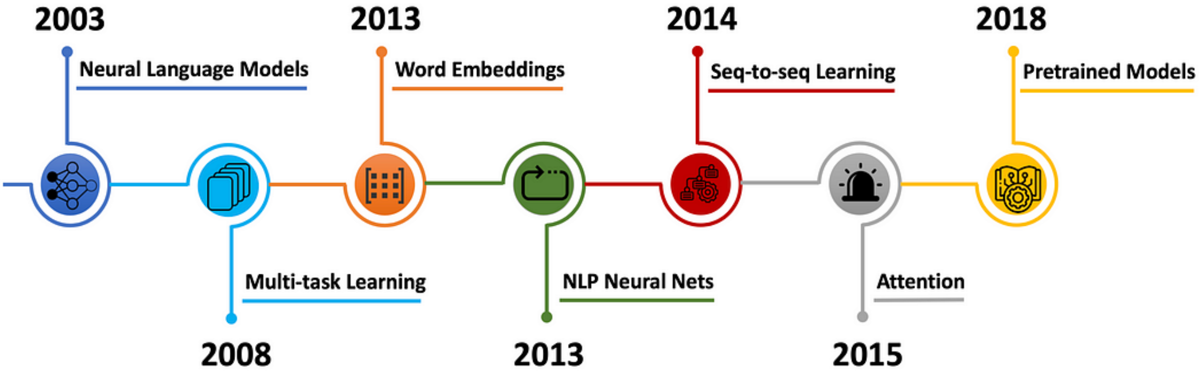


Figure 1

- Neural Language Models (2003)
  The introduction of Neural Language Models marked a shift from rule-based to data-driven approaches in text prediction. These models laid the groundwork for future developments by demonstrating the potential of neural networks in understanding the sequential nature of language.

- Multi-task Learning (2008)
  Multi-task Learning allowed for the simultaneous training of models on multiple NLP tasks, analogous to an educational approach where learning is integrated across subjects. This approach led to more robust models by leveraging shared knowledge, enhancing their ability to generalize and perform across different linguistic tasks.

- Word Embeddings (2013)
  The introduction of Word Embeddings provided a means to encode words into dense vectors where semantically similar words are mapped closer together. This representation became fundamental in numerous NLP models, allowing for a nuanced understanding of word meanings and relationships.

- Neural Networks in NLP (2013)
  2013 saw the adoption of neural network architectures, including RNNs and CNNs, becoming the standard for complex NLP tasks. Neural networks demonstrated a remarkable ability to handle a variety of NLP applications, from machine translation to question-answering systems, surpassing the performance of traditional models.

- Seq-to-seq Learning (2014)
  It emerged as a technique to convert sequences from one domain to another, utilizing encoder-decoder architectures.It became a cornerstone methodology for tasks like language translation, text summarization, and speech recognition, enabling end-to-end training of models.

- Attention Mechanisms (2015)
  The Attention Mechanism allowed models to selectively focus on relevant parts of the input sequence, addressing long-range dependency challenges. This development enhanced model interpretability and performance, particularly in seq-to-seq tasks, by mimicking human-like focus and relevance assessment.

- Pretrained Models (2018)
  The introduction of Pretrained Models such as BERT and GPT changed the NLP landscape, offering models pre-trained on vast datasets that could be fine-tuned for specific tasks.

ABOUT THE PAPER

Published in 2017, the paper "Attention is All You Need" presented a novel neural network architecture known as the Transformer. This model eschewed the previously dominant recurrent neural network (RNN) paradigm in favor of an architecture based solely on attention mechanisms. The Transformer has since set new standards for a variety of NLP tasks, most notably in machine translation.

TRANSFORMER MODEL ARCHITECTURE

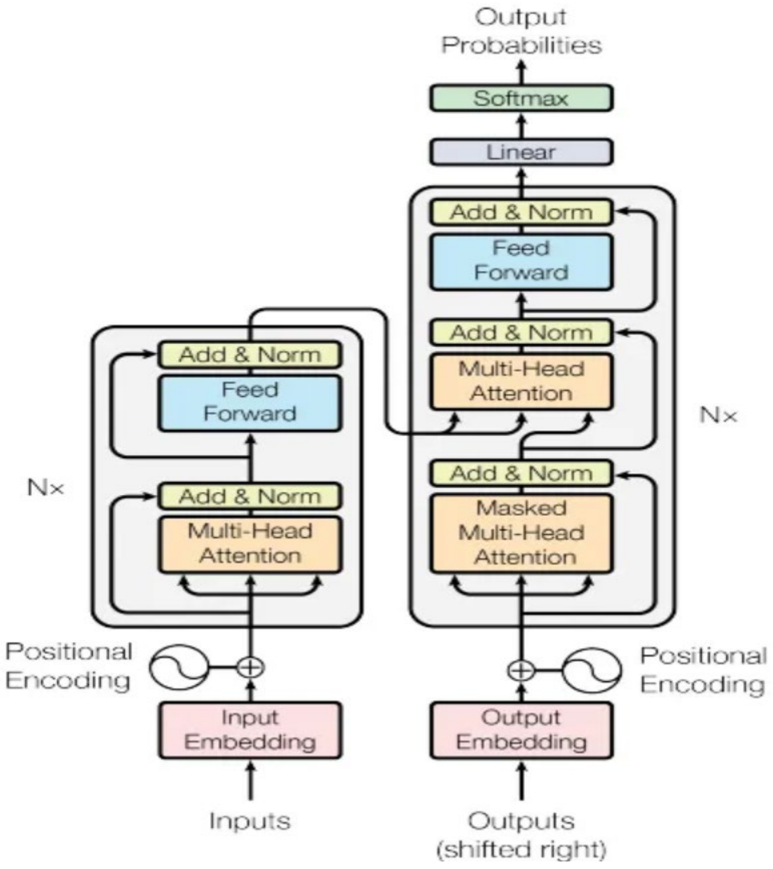


Figure 2

Input Embedding
This is responsible for converting raw text data into a structured format that can be processed by neural networks. Specifically, it translates words into high-dimensional vectors, capturing the semantic essence of each word.

Positional Encoding
It is crucial in the Transformer architecture as it injects information about the sequence order of words. This step is essential because, unlike previous architectures like RNNs, the Transformer processes words in parallel and needs a way to incorporate the notion of word order.

Multi-Head Attention
Multi-Head Attention allows the model to direct its focus to different segments of the input sequence simultaneously. By doing so, it gains various contextual insights from different representation subspaces at different positions, significantly enhancing the understanding of context within the input sequence.

Add & Norm
Following the attention mechanism, the Add & Norm step performs two key functions. First, it adds the input to the output of the attention layer (residual connection), and then it normalizes this output. This process helps stabilize the learning process across different layers in the network.

Feed Forward
Each position's output from the Multi-Head Attention layer is fed into a position-wise Feed Forward network. This network further processes the data, allowing the model to refine its understanding and make more nuanced connections within the data.
The Encoder consists of a set of layers, each containing the Multi-Head Attention and Feed Forward networks, among other components. These layers are stacked on top of each other, repeated 'N' times, where 'N' represents the number of layers, to iteratively enhance the model's ability to capture complex patterns in the data.

Output Embedding & Positional Encoding (Decoder)
In the Decoder part of the architecture, Output Embedding and Positional Encoding are applied in a manner analogous to the Encoder. This symmetry ensures that the model maintains contextual awareness as it generates the output sequence.

Masked Multi-Head Attention (Decoder)
The Decoder employs Masked Multi-Head Attention to ensure that the prediction for each output word is conditional only on the previous words, not on any future words. This masking is critical for the model to generate coherent and contextually appropriate sequences.

Linear and Softmax Layers
The final stages of the Decoder include Linear and Softmax layers. The Linear layer projects the Decoder's output into a larger vector space representing a probability distribution over the possible words. The Softmax layer then converts these values into actual probabilities, determining the most likely next word in the sequence.

LIMITATIONS OF RNNs & LSTMs

RNNs and LSTMs have been instrumental in the development of sequential data processing, especially in the context of NLP. RNNs are designed to handle a sequence of data points by maintaining a memory (hidden state) of previous inputs. LSTMs are an extension of RNNs designed to better capture long-range dependencies and prevent the vanishing gradient problem. Despite these advancements, both architectures face limitations that restrict their efficiency and effectiveness.

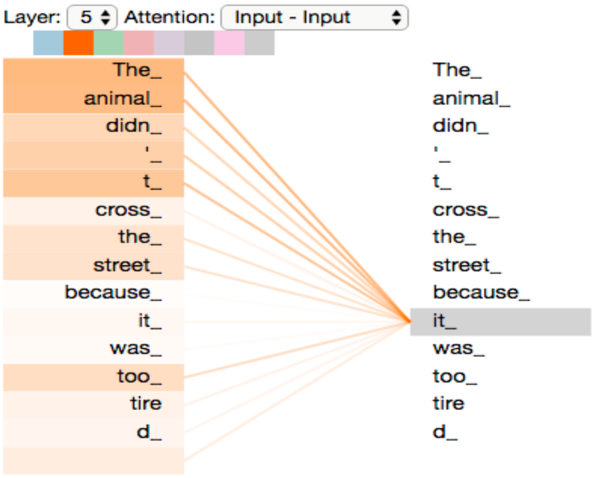
- Time-Consuming: RNNs and LSTMs process data sequentially. This characteristic means that each step depends on the computations of the previous step, leading to longer training and inference times.

- Overfitting: These networks have a tendency to overfit, especially when dealing with complex models with large numbers of parameters. Overfitting happens when a model becomes too familiar with the training data, absorbing its noise and details, to the point where it struggles to perform effectively on new, unseen data.

- Memory Limitations: RNNs and LSTMs are challenged by long sequences. While LSTMs were designed to better remember information over long sequences than traditional RNNs, they still face difficulties when the sequences are very long or when the required information is at the beginning of a long sequence.

- Challenges with Parallelization: Due to their sequential data processing, RNNs and LSTMs cannot fully leverage parallel computing architectures, which limits their performance and scalability when working with large datasets or in environments where computational resources like GPUs are available.

SELF-ATTENTION



Layer: 5 Attention: Input - Input

Figure 3

Self-attention, a variant of the attention mechanism, allows the model to look at other positions in the input sequence to better encode a word in a certain position. This mechanism forms the backbone of the Transformer model, enabling direct dependencies without regard to their distance in the input sequence.

MULTI-HEAD ATTENTION

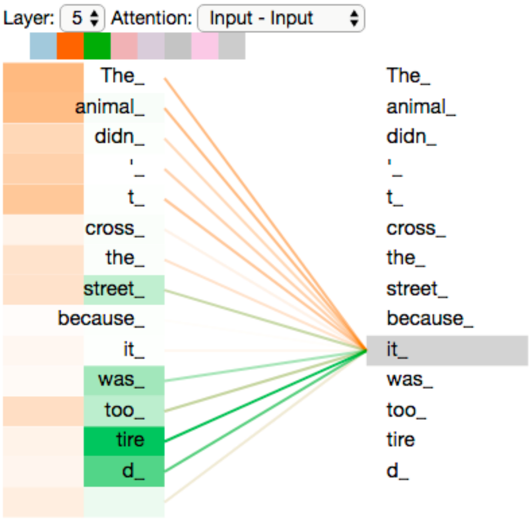
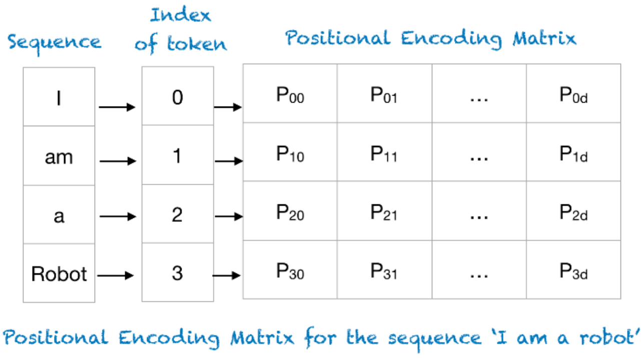


Figure 4

Multi-head attention enhances the self-attention mechanism by enabling the model to simultaneously focus on information from various representation subspaces at different points in the sequence. This approach provides a more nuanced understanding of the input by capturing various aspects of word meaning and context.

POSITION ENCODING



Sequence	Index of token	Positional Encoding Matrix			
I	0	P_{00}	P_{01}	…	P_{0d}
am	1	P_{10}	P_{11}	…	P_{1d}
a	2	P_{20}	P_{21}	…	P_{2d}
Robot	3	P_{30}	P_{31}	…	P_{3d}

Positional Encoding Matrix for the sequence 'I am a robot'

Figure 5

Since Transformers do not process sequences using recurrence, they require a method to incorporate information about the order of the sequence. Positional encoding introduces information about the order of words in a sequence into the model, allowing it to account for the position of each word relative to others.

CONCLUSION

The Transformer's groundbreaking approach to parallel processing and its impressive performance on NLP tasks suggest that it represents the future of AI in language processing. Its influence extends to the development of subsequent models like BERT and GPT, and its principles are being explored in other areas of AI, indicating its potential to drive further innovations.

REFERENCES

Vaswani, A. (2017). *Attention is All you Need.* https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Google Cloud Tech. (2021, August 18). *Transformers, explained: Understand the model behind GPT, BERT, and T5* [Video]. YouTube. https://www.youtube.com/watch?v=SZorAJ4I-sA

Yannic Kilcher. (2017, November 28). *Attention is all you need* [Video]. YouTube. https://www.youtube.com/watch?v=iDulhoQ2pro