Mathematical Statistician and Engineering Applications
ISSN: 2326-9865
DOI: https://doi.org/10.17762/msea.v70i1.2314

Exploring the Power of Transformer Models in Hospitality Domain

Jyoti Parsola

Asst. Professor, School of Computing, Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Article Info
Page Number: 324-330
Publication Issue:
Vol. 70 No. 1 (2021)

Article History
Article Received: 25 January 2021
Revised: 24 February 2021
Accepted: 15 March 2021

Abstract
Despite decades of medical advancements and a rising interest in precision healthcare, the great majority of diagnoses are made after patients start to exhibit observable symptoms of sickness. However, early disease indication and detection can give patients and caregivers the opportunity for early intervention, better disease management, and effective use of healthcare resources. Deep learning and other recent advancements in machine learning provide a fantastic chance to fill this unmet demand. Transformer designs are very expressive because they encode long-range relationships in the input sequences via self-attention methods. The models we offer in this work are Transformer-based (TB), and we provide a thorough description of each one in contrast to the Transformer's typical design. This study focuses on text-based task (TB) models used in Natural Language Processing (NLP). An examination of the key ideas at the core of the effectiveness of these models comes first. NLP's flexible architecture allows it to incorporate various heterogeneous concepts (such as diagnoses, treatments, measurements, and more) to further improve the accuracy of its predictions. Its (pre-)training results in disease and patient representations can also be helpful for future studies (i.e., transfer learning).

1. Introduction

Transformer models, like the GPT-3, have shown exceptional performance in a range of natural language processing tasks, including as language comprehension, question answering, text production, and machine translation. These models effectively capture long-range dependencies and context information because they analyze incoming data using a self-attention mechanism.

Transformer models may be used in a variety of use cases in the hospitality industry, such as:

1. Customer service: Transformer models may be trained on a lot of data about customer service to provide automatic answers to frequent client questions like hotel availability, booking status, or restaurant reservations.

2. Personalization: Transformer models may be used to analyze consumer information, such as prior reservations, preferences, and comments, to provide tailored recommendations and experiences. Examples of these experiences include customized amenities in rooms, restaurant ideas, or activities.

3. Sentiment analysis: Transformer models may be taught to analyze customer feedback from numerous sources, including reviews, social media, and surveys, to spot positive or negative attitudes and to glean insightful information for service enhancement.

4. Translation: Transformer models may be used to provide in-the-moment translation services for clients who speak various languages, enabling hotels and other hospitality firms to give foreign visitors a better customer experience.

The strength of transformer models in the hotel industry comes in their capacity to handle enormous volumes of unstructured data, draw up insightful conclusions, and provide clients individualized experiences. Transformer models will become more crucial as the hospitality sector develops since they let companies adapt to the shifting demands and tastes of their clients.

Natural Language Processing (NLP) has evolved via a series of ever-improving models, starting with bag of words in 1954 and on through TDIF in 1972, RNNs and LSTMs in 1997, and Transformers in 2017. Many people are unaware that 20 years ago, text classification and Naive Bayes algorithms were already automating medical codes for patient visits with 85%+ accuracy. Healthcare has even longer used chatbots to automate customer support. The accuracy and use cases of these early models were amazing. And ten years ago, clinical notes might be used to predict illnesses like sepsis and other disorders using LSTMs, which are excellent for time series and sequential data like language. Transformers are not used in the majority of NLP commercial healthcare markets today. Transformers have surpassed human performance at accuracy levels of 99%+, however, which is where they have thrived. For instance, radiology medical coders may be fully automated with the exception of a tiny exception pool. Additionally, machine translation and summarization tasks using NLP are now practicable in the medical field. Transformers allow for the conversion of difficult medical jargon into more understandable terms for patients.

Transformers is a machine learning sequence model that Google created in 2017. It determines the weighting of word vectors using a mathematical concept called attention. Transformers depend on several matrix multiplications, which is simply another way of stating that they pay attention. Because matrix multiplication makes use of GPUs, it performs well in parallel computing. Before transformers, LSTMs were all the rage in NLP, but they couldn't be trained concurrently since they relied on each word in a sequence. Transformers, e.g., GPT-3, take into account an entire sequence at once, whereas LSTMs are state dependent; the earlier portions of a sequence are essentially lost by the end of a long sequence. LSTMs could never scale to a training data size with millions of records and billions of parameters. Transformers are computationally slower than LSTMs (Big O notation of O(N2)) since they consider the weighting of every word against every other word at simultaneously for prediction (autoregressive), although parallelization with GPUs enables them to run and complete quicker.

All of this implies that transformers are superior at NLP tasks like document categorization, sentiment analysis, machine translation, question and answer, and summarization because they can anticipate deeper connections inside words. And in the field of healthcare, it includes precision medicine (creating a prescription schedule specifically for each patient), greater diagnostic accuracy prediction, and more automation of repetitive jobs like computer data input and medical coding. Before transformers, a rule-based branching logic with a combination of keyword extraction using a graph-based algorithm like TextRank would be the best structure

for creating a patient summary of their health record. When put into production, these early models were very difficult to maintain and had poor fluency. This explains why there haven't been any real automated narrative summaries used in healthcare; instead, doctors have depended on other doctors to manually read and summarize the medical records on their behalf. Finally, Transformers can be used as a machine learning model to solve a healthcare issue like document summarization.

2. Literature Survey

Using attention-based models, Chorowski et al. [1] suggested a unique method for voice detection. On a number of benchmark datasets, they show how effective their method is and how it outperforms more established models.

Vaswani et al. [2] introduced the Transformer, a revolutionary neural machine translation architecture that computes representations of input and output sequences only through self-attention methods. The Transformer considerably reduces training time while achieving state-of-the-art outcomes on a number of machine translation benchmarks.

Based on BERT, a previously trained language model, Alsentzer et al. [3] presented publicly accessible clinical embeddings. They show how these embeddings may be utilized to enhance the performance of a number of clinical NLP tasks, such as relation extraction and named entity identification.

Gururangan et al. [4] investigated the efficacy of task and domain adaptation for language models that have already been trained. They demonstrate how optimizing a pre-trained language model on a particular domain or job may result in substantial performance gains.

Zhou et al. [5] unveiled Sentix, a brand-new pre-trained sentiment analysis model created exclusively for cross-domain sentiment analysis. On a number of benchmark datasets, they show how effective their method is and how it outperforms more established techniques.

A single model may learn to carry out numerous natural language processing tasks concurrently without any task-specific supervision, according to the idea of unsupervised multitask learning for language models described by Radford et al. in [6].

In order to improve the modeling of sequential information in natural language, Wang et al. [7] created the R-Transformer, a variation of the Transformer model that incorporates a recurrent neural network (RNN).

In a comparison between the Transformer and RNN-based models for multilingual neural machine translation, Akew et al. [8] discovered that the Transformer performed better on numerous benchmark datasets than the RNN-based models.

GloVe, a technique for learning word representations that combines global co-occurrence data with local context windows to capture both semantic and syntactic information, was presented by Pennington et al. [9].

BART is a denoising sequence-to-sequence pre-training approach for natural language creation, translation, and comprehension tasks that Lewis et al. [10] suggested. BART employs a Transformer architectural version and produces cutting-edge outcomes on several benchmark datasets.

3. Proposed Model

A. System Overview

The transformer design is founded on the idea of attention, allowing the model to concentrate on crucial input sequence segments while discarding unimportant data. The input sequence's weighted sums are computed using this attention process, and the output sequence is subsequently computed using those results. Residual connections, layer normalization, and feedforward networks are further features of the transformer that assist the model be more stable and effective. Because they use a series of transformer blocks to convert the input sequence into an output sequence, they are known as transformers. Each transformer block has a feedforward layer that transforms the input sequence at each tier of the model as well as a self-attention mechanism. One of the most used deep learning architectures in recent years, the transformer architecture is extensively utilized in natural language processing as well as other fields including computer vision and voice recognition. The encoder-decoder design of the original Transformer model is shown graphically below.

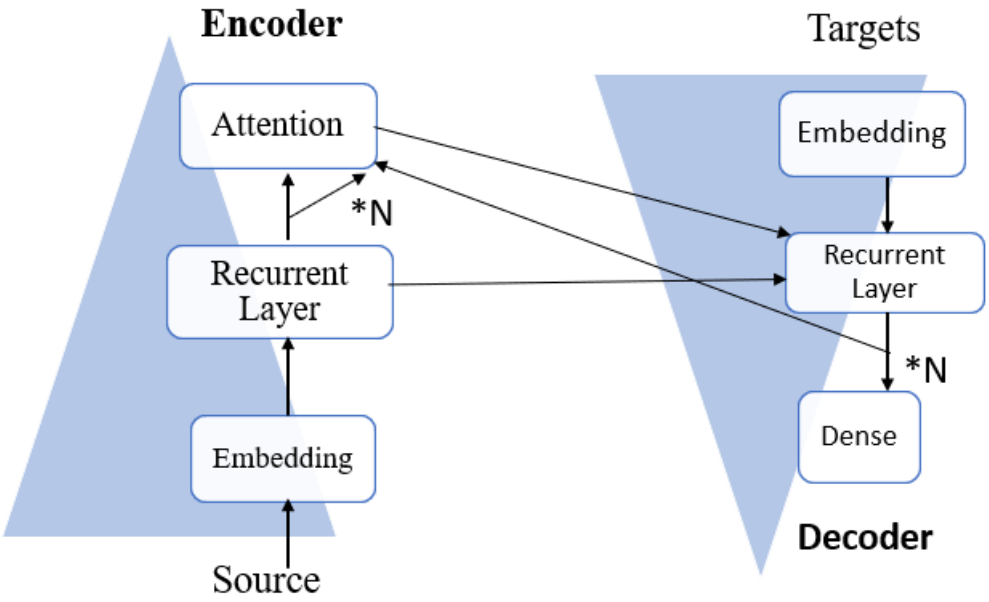

Figure 1. Encoder -Decoder transformer

Encoder:

The model was made to translate a series of texts from one language to another using machine translation. The input sequence in the source language was converted into a similar output sequence in the target language using the encoder-decoder architecture. The source sequence was encoded using the Transformer model's encoder component, while the target sequence was

created using the decoder component. A position-wise feedforward layer came after a stack of similar self-attention layers to make up the encoder. The decoder also included a position-wise feedforward layer and a stack of identical self-attention layers.

The transformer-based encoder part encodes the input sequence $\mathbf{X}1{:}n$ to a *sequence* of *hidden states* $\mathbf{X}1{:}n$, thus defining the mapping:

$f\theta\text{enc}{:}\mathbf{X}1{:}n \rightarrow \mathbf{X}1{:}n$.

Decoder:

Given the prior tokens and the encoded input, they are trained to predict the subsequent token in the sequence. In order to capture the dependencies and interactions between the various components of the sequence, as well as how to produce text that is fluent and cohesive, the model is encouraged to learn from this training aim. The decoder-only models' autoregressive and conditional text generation are two more distinguishing characteristics. Text is created using decoder-only models autoregressively, which means that each word is produced dependent on the words that came before it in the sequence. This is what enables models like the generic models to provide replies that are both logical and relevant to the circumstances. the capacity to produce text that is pertinent to a given input by conditioning it on that input, such as a prompt or a list of keywords.

The transformer-based decoder part then models the conditional probability distribution of the target vector sequence $\mathbf{Y}1{:}n$ given the sequence of encoded hidden $\mathbf{X}1{:}n$:

$p\theta dec(\mathbf{Y}1{:}n|\mathbf{X}1{:}n)$.

B. Algorithm

Pre-processing

1. Used a tokenizer, such as BERT tokenizer, to break down input texts into smaller word units.

2. Using a fixed-size vocabulary to map the tokens to their associated integer IDs.

3. To build input tensors, pad the sequences to a predetermined length.

4. Carry out the procedure once again for the output sequences.

Encoder

1. Using an embedding layer to insert input tokens into continuous vector space.

2. Giving the embeddings positional encodings.

3. Running many layers of transform and encode neural networks (transformer encoder layers) on the embedded input.

4. Display the input sequence's final concealed representation.

Decoder

1: Using an embedding layer to embed output tokens into continuous vector space.

2. Giving the embeddings positional encodings.

3. Running numerous layers of feedforward and self-attention neural networks (also known as transform decoder layers) over the embedded output.

4. Attending to the encoder outputs and the previously produced decoder outputs using a masked multi-head attention method.

5. Using a linear layer and the softmax activation function to create the following token in the output sequence.

6. Reiterating steps 1 through 5 until an end-of-sequence token is produced or the output is at its predetermined maximum length.

Training

1. Calculating the cross-entropy loss between the real output sequence and the anticipated output sequence.

2. Using gradient descent methods, backpropagate the loss via the decoder and encoder layers and update the model parameters.

3. Until convergence, repeat steps 1-2 for further epochs.

A trained model may produce output sequences from fresh input sequences. To avoid instructor forcing during inference, we utilized the token that was just produced as the input for the subsequent step up until the end-of-sequence token or the predetermined maximum output length was reached.

4. Result

We can see health cases from clinical dataset which can see using at higher risk and lower risk comparison. We can also see anomalies here higher data outliers. It can be again used in pre-trained data learning model to classification and learning transfer learning to enhance system.

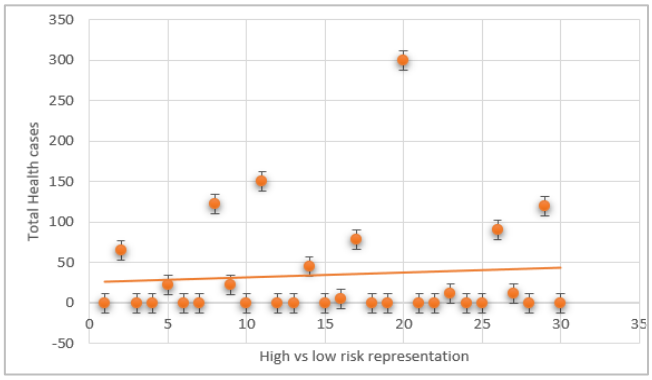

Figure 2. High Vs Low Risk representation

5. Conclusion

When processing input data, the transformer architecture makes no assumptions about recurrence or convolution patterns. The transformer design is thus appropriate for any sequence data. The same method may be used for computer vision (sequences of medical image patches) and reinforcement learning (sequences of states, actions, and rewards), as long as we can represent our input as sequence data. The input features are just a series of enhanced embeddings using position-wise feed-forward networks, multi-head attention methods, and layer normalizations. For this reason, transformers are the foundation of Abstractive Health's NLP. The results of the poll indicate that clinical health professionals are specialists at utilizing these models and have spent a lot of effort tailoring them to function particularly for the difficulties in the healthcare industry, such as with regard to lengthy paperwork and medical factuality. Fundamentally, we think that Transformers represents the cutting-edge AI that is already transforming healthcare.

References

1. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. arXiv 2015, arXiv:1506.07503
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
3. Alsentzer, E.; Murphy, J.R.; Boag, W.; Weng, W.H.; Jin, D.; Naumann, T.; McDermott, M. Publicly available clinical BERT embeddings. arXiv 2019, arXiv:1904.03323.
4. Gururangan, S.; Marasovic, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don’t stop pretraining: Adapt language models to domains and tasks. arXiv 2020, arXiv:2004.10964.
5. Zhou, J.; Tian, J.; Wang, R.; Wu, Y.; Xiao, W.; He, L. Sentix: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 568–579.
6. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. OpenAI Blog 2019, 1, 9.
7. Wang, Z.; Ma, Y.; Liu, Z.; Tang, J. R-transformer: Recurrent neural network enhanced transformer. arXiv 2019, arXiv:1907.05572.
8. akew, S.M.; Cettolo, M.; Federico, M. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. arXiv 2018, arXiv:1806.06957.
9. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
10. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv 2019, arXiv:1910.13461.