Transformers in Natural Language Processing

Technical Report · September 2022

DOI: 10.13140/RG.2.2.18062.84809

1 author:

Md Mahmud Hasan
Frankfurt University of Applied Sciences
4 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Transformers in Natural Language Processing

Md Mahmud Hasan
Department of Computer Science and Engineering
Frankfurt University of Applied Science
Frankfurt am Main, Germany
md.hasan3@stud.fra-uas.de

***Abstract*—Natural Language Processing(NLP) is a technique that allows computers to understand, interpret, and use human languages. A lot of research has been done on NLP for quite some time, but it is now becoming a core part of artificial intelligence (AI). Since the introduction of Transformers, NLP has experienced a paradigm shift, with the training point of a model shifting from a blank model to a pre-trained architecture. Due to its ability to work without recurrent connections and convolutions, it has the potential to revolutionize deep learning architectures. According to previous research, Transformers make it so simple to introduce new language models that it happens quickly. It is now proven that Transformer-based NLP made textual analysis models to be exhaustively and endlessly trained, and gave AI-based textual analyzers the capacity to efficiently process long text. In this paper, we will look at how the transformers framework became the de facto standard in a wide variety of NLP-related domains and find out why language models, a subset of transformers, are used to handle nearly all natural language processing tasks today.**

***Index Terms*—Natural Language Processing, Transformers, NLP, Machine Learning, GPT, BERT, T5**

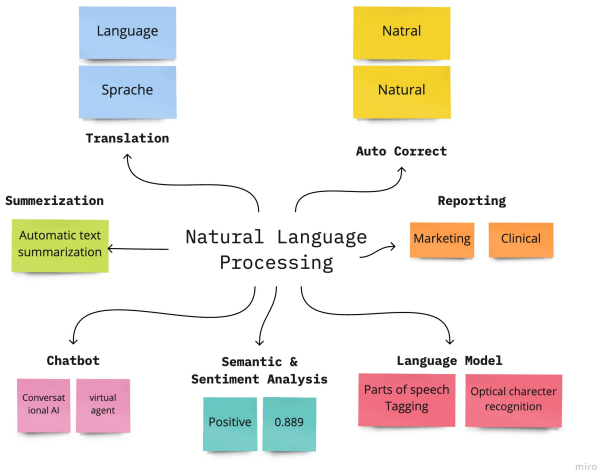

Fig. 1. Natural Language Processing

I. Introduction

The term natural language processing (NLP) refers to the field of computer science, specifically to the field of artificial intelligence, which aims to give computers the ability to understand text and spoken language in the same way humans can [1]. A computer can communicate with a person by using human language using Natural Language Processing (NLP). Furthermore, Natural Language Processing allows computers to interpret text, hear speech, and read text. In order to bridge the gap between human and computer communication, NLP draws from a variety of disciplines, including computational linguistics and computer science (Figure-1).

Combining computational linguistics with statistical, machine learning, and deep learning models, NLP aims to model human language based on its rules. This combination of technologies empowers computers to 'understand' human language at its most basic level, including the speaker or writer's intent and sentiment as well as the text or voice data they process.

This technology enables computer programs to translate text between languages, respond to spoken commands, and summarize large quantities of text, even in real time [1]. In the form of voice-activated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences, NLP has likely played a role in our daily life. As well as assisting business operations and improving employee productivity, NLP is also increasingly being used to simplify mission-critical business processes at the enterprise level.

Thanks to Prof. Dr. Martin Simon.

II. Background

Ferdinand de Saussure, a Swiss linguist who taught three courses at the University of Geneva in the early 1900s, pioneered the concept of language as a "system" [2]. An idea is represented by a sound in a language; nevertheless, an idea might have more than one meaning depending on the situation. This was the very first thought that was ever conceived of regarding "Language as a Science."

According to him, meaning is formed within language, in the connections and contrasts between its constituent elements [2]. In Saussure's view, "meaning" emerges from the contrasts and connections between words in a language. Communication can only take place when there is a common language system. Saussure saw society as a system of "shared" social norms that make it possible for people to think in a reasonable, "extended" way, which leads to decisions and actions on the part of individuals. A similar perspective can be extended to contemporary programming languages.

Albert Sechehaye and Charles Bally, two of Saussure's coworkers, understood the significance of his ideas and

took the unprecedented step of gathering ”his notes for a manuscript” as well as the course notes of his students after his death [2]. They developed the Cours de Linguistique Générale from these, which was released in 1916. The book set the groundwork for what has come to be known as the structuralist method, which was initially applied to linguistics before spreading to other disciplines, such as computers.

In the year 1950, Alan Turing published a paper in which he described a test for a ”thinking” machine [3]. He asserted that a machine could be said to be capable of thinking if it could participate in a discussion using a teleprinter and replicate a human so perfectly that there were no discernible distinctions. Soon later, in 1952, the Hodgkin-Huxley model demonstrated how the brain creates an electrical network by using neurons [4]. These occasions contributed hugely to the conception of Artificial Intelligence (AI), Natural Language Processing (NLP), and the advancement of computers.

In 1957, Noam Chomsky published Syntactic Structures [5], a book that would forever alter the way people thought about language and computing by proposing that a language’s sentence structure would need to be altered before a computer could grasp it. In order to accomplish this, Chomsky developed a grammar system known as Phase-Structure Grammar, which painstakingly transformed real English phrases into a structure that computers could understand. The overarching objective was to develop artificial intelligence, or a computer system, that could mimic the thinking and communication processes of the human brain.

John McCarthy introduced LISP (Locator/Identifier Separation Protocol) [6], a computer language in 1958 that is still in use today. The ”typewritten” remark and answer procedure known as ELIZA was created in 1964 with the intention of imitating a psychiatrist by applying techniques of reflection. There was no comprehension on the side of the computer; it simply accomplished this via rearrangement of sentences and adherence to relatively simple grammar rules. The Automatic Language Processing Advisory Committee, or ALPAC for short, was established by the American National Research Council (NRC) in 1964. This group was tasked with monitoring the development in the field of natural language processing.

The NRC and ALPAC stopped supporting research on machine translation and natural language processing in 1966, starting the first AI and NLP pause [7]. After twelve years of research, there were still no machines that could even come close to having a simple conversation, and machine translations were still more expensive than manual human translations. Many thought that development in artificial intelligence and natural language processing (NLP) had reached a dead end in 1966.

The field of study in Natural Language Processing and Artificial Intelligence was able to recover from the broken expectations established by extreme enthusiasts after nearly fourteen years of pause. In some aspects, the AI hiatus had ushered in a new era of novel conceptions, with earlier theories of machine translation being abandoned in favor of new theories that encouraged new study. Early NLP research had a tendency to combine statistics with linguistics, but this was supplanted by a focus on pure statistics. Simple approximations have replaced in-depth study, and the evaluation process has grown more robust, both of which were fundamentally reoriented in the 1980s.

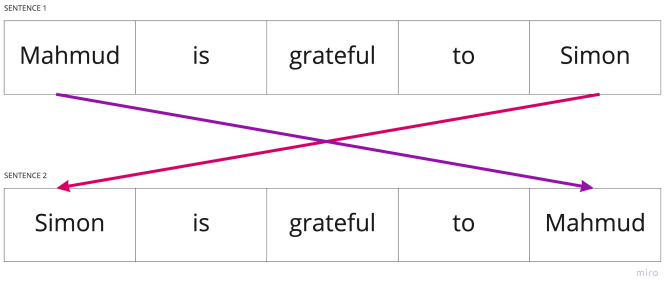

Fig. 2. Sequence is important

The use of statistical models for NLP analyses saw a sharp increase in popularity in the 1990s [1] [7]. Keeping up with the enormous volume of internet material has made purely statistical NLP techniques extraordinarily useful. Since they can identify and keep track of linguistic data clusters quantitatively, N-Grams were proven beneficial. At some point, besides statistical data, it was important to do linguistic analysis as well. For linguistic analysis sequence of words is a very important dimension. For instance, in Figure-2 we can say that even though the data is same, the meaning is quite opposite which happens only for the position of the word.

Yoshio Bengio and his colleagues proposed the first neural ”language” model in 2001 by making use of a feed-forward neural network [8]. The feed-forward neural network is an artificial neural network that does not employ connections to generate a cycle. In this kind of network, information only travels in one direction from input nodes to output nodes, passing via any hidden nodes en route.

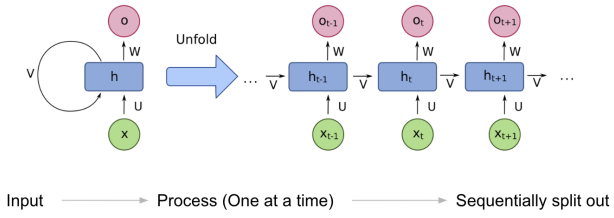

Fig. 3. RNN - Handling sequence data [9]t

Sequence data requires that the data be processed in a specific order, and prior to RNNs, there was no effective method for handling this type of data. LSTMs outperformed RNNs in that they remembered the inputs from earlier sequences for lengthy sequences [9]. Additionally known as the vanishing gradient problem, this was a serious issue for RNNs [10]. In order to keep the weights of the early inputs

from zeroing out, LSTMs keep track of the information that is significant in the sequence. A Gated Recurrent Unit is an updated variant of the RNN (GRU). Despite being quite similar to the LSTM, it varies in that it has unique gates for storing extended sequence information. For a while, RNNs and LSTMs were the workhorses of the NLP industry, relied on by practically everyone. However, attention networks quickly took their position as superior architecture.

From 2015–2016, attention-based networks gained popularity [11] [12]. A certain subset of the data input can be focused on using attention networks, a form of neural network. The network can focus by defining a given subset. On numerous NLP tasks, including Neural Machine Translation, Language Modeling, and Question Answering issues, these models have been shattering performance records. Additionally more effective and using less processing power are attention networks. This is a big advancement because training RNNs frequently calls for a lot of computational power in the form of a GPU, which is not always available.

Modern NLP design has been dominated by the Transformer model, a particular type of attention-based network released in 2017 [13]. Although the Transformer handles sequence data similarly to RNNs, there is no requirement that the data be input into the model in a specific sequence. As a result, parallelization enables the Transformer model to train more quickly and with a larger amount of data. The Transformer model paved the way for the GPT 3, BERT, ERNIE 2.0, and XLNet eras of NLP, which are currently trending and added incredible value in the field of natural language processing.

III. Transformers

Transformers were initially presented for the first time in the 2017 NIPS article titled Attention is All You Need by researchers collaborating with Google [13]. Transformers are meant to operate with sequence data and will take an input sequence and utilize it to generate an output sequence. The first segment of a transformer is an encoder, which primarily acts on the input sequence, and the second is a decoder, which functions on the intended output sequence during training and predicts the next item in the sequence. For instance, in a machine translation issue, the transformer might use a string of English words and repeatedly anticipate the subsequent German word until the entire sentence has been translated. In Figure-4, The encoder is on the left in the diagram below, and the decoder is on the right, showing how a transformer is put together.

Transformers are made up of N number of encoders and decoders. In their proposal paper, they used six encoders and six decoders. They are all extremely similar to one another, encoders. The architecture is identical across all encoders. Decoders have a common characteristic, making them quite similar to one another. Each encoder is made up of two layers: the self-attention layer, and a feed-forward neural network layer as shown in Figure-4. The inputs to the encoder initially pass through a layer for self-attention. As it encodes a particular word, it helps the encoder consider other words

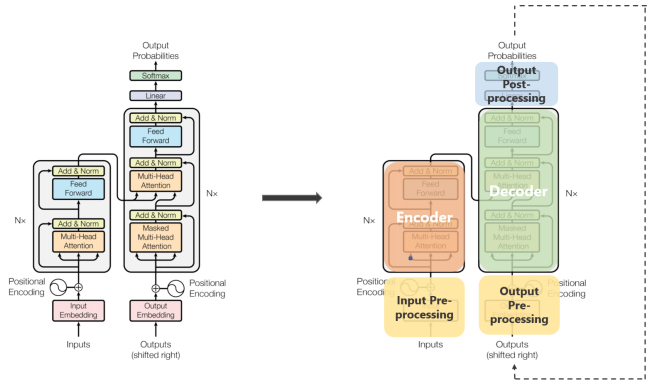

Fig. 4. Transformer Architecture [13]

in the input text. Both of those levels are included in the decoder, but in between them is an attention layer that aids in keeping the decoder's attention on key elements of the input text. Figure-5 describes how it encodes a sentence in each encoding layer.

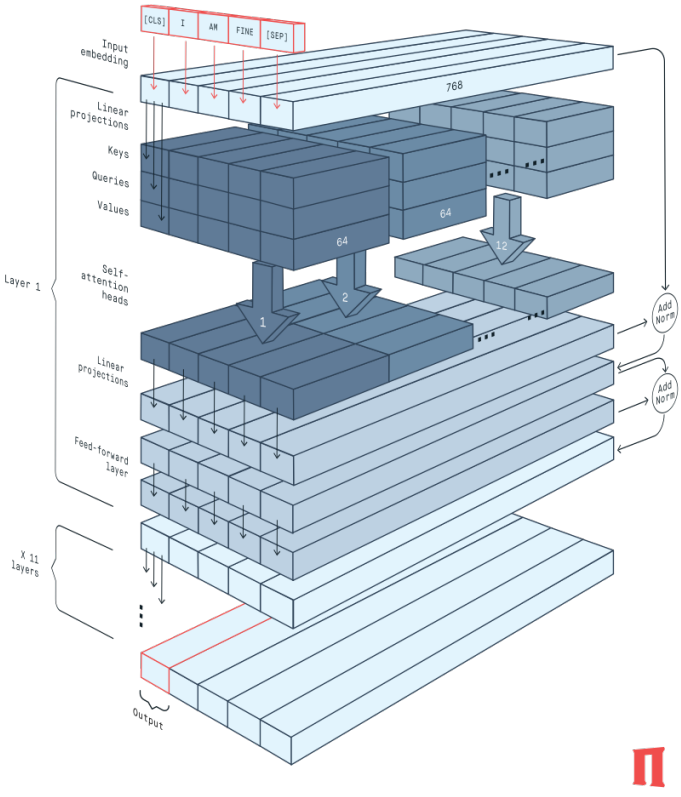

Fig. 5. BERT Encoding [14]

The positional encoding of the individual words is a small but significant component of the model. A sequence depends on the order in which its pieces appear, thus since there are no recurrent networks that can recall how sequences are fed into a model, it must somehow assign each word or component in our sequence a relative position. The embedded representation

(n-dimensional vector) of each word is expanded to include these locations.

The equation in Figure-6 can be used to explain the simplicity of the attention mechanism.

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

Fig. 6. Transformer Attention [15]

The query (a vector representation of a single word in the sequence) is stored in a matrix denoted by Q, the keys (vector representations of all the words in the sequence), and the values (vector representations of all the words in the sequence) are stored in a matrix denoted by K. V is made up of the same word sequence as Q for the purposes of the encoder and the decoder, which are both multi-head attention modules. However, V differs from the sequence represented by Q for the attention module that is taking into account the encoder and decoder sequences.

To further simplify things, we may state that the values in V are multiplied and added with certain attention-weights a, where the weights are determined by the equation in Figure-7.

$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Fig. 7. Transformer Attention [15]

That is to say, the weights 'a' is determined by the degree to which each word in the sequence (represented by Q) is affected by every other word in the sequence (represented by K). In addition, the SoftMax function is used to distribute the weights a between 0 and 1. Then, all the words in the sequence that are introduced in V are given those weights (same vectors as Q for encoder and decoder but different for the module that has encoder and decoder inputs).

The attention mechanism can be parallelized into several other processes that can be employed in conjunction with it, as shown in the Figure-5. The linear projections of Q, K, and V are used to repeatedly repeat the attention mechanism. As a result, the system can benefit from learning from various Q, K, and V representations. By multiplying Q, K, and V by weight matrices W that are acquired during training, these linear representations are created.

Matrices Q, K, and V are distinct for each position of the attention modules in the structure, depending on whether they are in the encoder, the decoder, or in between the encoder and the decoder. The rationale for this is to focus on either the entire encoder input sequence or a portion of the decoder input sequence. The encoder and decoder are connected by a multi-head attention module, which makes sure that the input sequences from both are considered up to a certain location.

A pointwise feed-forward layer follows the multi-attention heads in the encoder and decoder. This little feed-forward network, which can be thought of as a distinct, identical linear transformation of each element from the given sequence, has identical parameters at every point.

A. Benefits over RNN

Two of the most significant advancements in the field of NLP were accomplished by Transformers. First, they made it possible to process whole sequences in parallel, enabling the speed and capacity of sequential deep learning models to be scaled to previously unattainable levels. Second, they included "attention mechanisms" that made it feasible to follow the relationships between words in both forward and backward-looking text sequences all of which were coming from extremely long sentences [16].

Vaswani et al. [13] state that a number of reasons led them to give up using recurrence and convolutions, including:

- Self-attention layers were found to be faster than recurrent layers for shorter sequence lengths, and they can be constrained to just consider a neighborhood in the input sequence for extremely long sequence lengths.
- The number of consecutive operations required by a recurrent layer is dependent on the duration of the sequence, whereas the number of sequential operations required by a self-attention layer is always the same.
- In convolutional neural networks, the kernel width has a direct impact on the long-term relationships that can be built between pairs of input and output positions. Long-term dependency tracking would necessitate the use of huge kernels or stacks of convolutional layers, which could raise the computing cost.

B. Types of Transformers

Shortly after the invention of Transformers, previous NLP models have been mostly supplanted by the Transformer model framework.

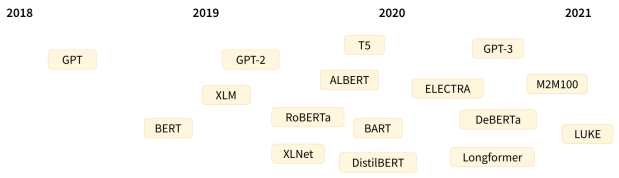

Fig. 8. Transformers Model Timeline [17]

Here are some most commonly used Transformer models:

- GPT - Generative Pre-trained Transformer [18]
- BERT - Bidirectional Encoder Representations from Transformers [19]
- T5 - Text-To-Text Transfer Transformer [20]

While BERT is based on the Transformer encoder, the GPT model solely employs the Transformer structure's unidirectional decoder (bidirectional). T5 makes use of an encoder-decoder Transformer construction that is fairly similar to the original design. After the creation of the Transformer, several significant models were released. The hugging face community recently posted an image (Figure 8) in which they evaluated a language model.

Additionally, these general architectures vary in terms of the quantity and size of the components that make up an encoder or decoder (i.e., the hidden size, number of layers, and the number of heads used for self-attention). A notable point is that, depending on the kind of activity that needs to be accomplished, either the entire architecture or only the encoder or decoder can be used as the language model. This is summarized in the table I.

Some subsets Transformers		
Type	Model	Tasks
Encoder	DistilBERT, ALBERT, BERT, RoBERTa, ELECTRA	Named entity recognition, extractive question answering, Sentence classification
Decoder	Transformer XL, GPT, GPT-2, CTRL	Text generation
Encoder-decoder	Marian, mBART, T5, BART	Generative question answering, summarization, translation

TABLE I
TRANSFORMER TYPES

C. Applications of Transformers

Transformers provides APIs and tools for downloading and training state-of-the-art pre-trained models with minimal effort. Pretrained models can save a lot of time and resources needed to train a model from scratch while lowering compute expenses and carbon footprint [21]. These models facilitate typical tasks in several modalities, including:

- **Computer Vision:** object detection, image classification, and segmentation.
- **Natural Language Processing:** named entity recognition, text classification, summarization, question answering, translation, language modeling, multiple choice, and text generation.
- **Multimodal:** information extraction from scanned documents, table question answering, optical character recognition, video classification, and visual question answering.
- **Audio:** audio classification and automatic speech recognition.

D. How to use Transformers

Transformers offer JAX, TensorFlow, and PyTorch framework compatibility. As a result, it is possible to train a model using only three lines of code in one framework and load it for inference in another, depending on the stage of the model's lifecycle. Additionally, models can be exported to file types like ONNX and TorchScript for use in deployment in real-world settings [17].

The simplest way to use a pre-trained model for inference is with the pipeline(). The pipeline() can be utilized right out of the box for a wide variety of tasks in many modalities. The following table (Table II) lists several supported tasks in the field of NLP:

Transformers in NLP		
Task	Description	Identifier
Text classification	Add a label to a given text sequence	pipeline(task="sentiment-analysis")
Text generation	generate text in response to an instruction	pipeline(task="text-generation")
Name entity recognition	assign a label to each token in a sequence (people, organization, location, etc.)	pipeline(task="ner")
Question answering	analyze the text for an answer given some context and a question	pipeline(task="question-answering")
Fill-mask	predict the correct masked token in a sequence	pipeline(task="fill-mask")
Summarization	summarize a sequence of text or documents	pipeline(task="summarization")
Translation	translate text from one language to another	pipeline(task="translation")

TABLE II
TRANSFORMER SUBSETS IN NLP

For instance, to classify ESG KPI it is possible to use a fine-tuned Language model with just a few lines of code.

```
#import libraries
from transformers import AutoTokenizer,
    AutoModelForSequenceClassification
import pandas as pd

tokenizer = AutoTokenizer.from_pretrained("nbroad/
    ESG-BERT")
model = AutoModelForSequenceClassification.
    from_pretrained("nbroad/ESG-BERT")

train_df = pd.DataFrame(data=pipe(sentences,
    truncation=True))
# Printing classified mean score
print(train_df.groupby('label').mean())
```

Listing 1. Using nboard/ESG-BERT model

The full code is also available in this GitHub repository.

IV. Result and Discussion

I have used the pre-trained domain-specific ESG-BERT mode [22]l which is fine-tuned to perform text classification on Sustainable Investing text data. The result of this model is amazing as it was successfully able to classify each sentence based on sustainability level perfectly. A snippet of the results is shown in Figure 9.

index	sentence	esg_label	esg_score
0	integrated sustainability report turn change into opportunity embrace sustainability integrated sust...	Product_Design_And_Lifecycle_Management	0.9519550800323486
1	founded in 1871, the technology company offers safe, efficient, intelligent and affordable solutions...	Energy_Management	0.30309638381004333
2	in 2021, continental generated sales of 33.8 billion and currently employs more than 190,000 people ...	Product_Design_And_Lifecycle_Management	0.2649711072444916
3	continental ag 2021 integrated sustainability report 3 group sustainability scorecard 2021 continuin...	Air_Quality	0.9391571283340454
4	allocated business with emission-free mobility and industry und industrie in millions 991 n. a. circ...	Employee_Health_And_Safety	0.96169513463974
5	scope 1 includes emissions from the burning of fossil fuels as part of continentals own processes, a...	Air_Quality	0.5855714678764343
6	co2 emission factors correspond to co2 equivalents (co2e).	Air_Quality	0.8377958536148071
7	2 contains a small amount of imputed data for parts of the continental group that did not report dat...	Customer_Privacy	0.7343450784683228
8	3 calculated using the market-based calculation method of the ghg protocol.	GHG_Emissions	0.9652166366577148
9	where contract-specific emission factors were not available, the standard emission factors from defr...	Air_Quality	0.9657467007637024
10	4 includes the relevant production and research and development locations.	Supply_Chain_Management	0.36295393109321594

Fig. 9. ESG classification

There were 75 reports in total and it is nearly impossible to classify such a way without Transformer technology. Using BERT makes it not only possible but also much easier in comparison with RNN.

V. Conclusion

In the realm of natural language processing, transformers are potent deep learning models with many applications. The issues with RNN, such as parallel processing and dealing with long sequences of text, were effectively addressed and resolved. Furthermore, training a model has become much easier. Using cutting-edge transformers for common tasks like sentiment analysis, question-answering, and text summarization is incredibly simple for developers thanks to the transformers package offered by TensorflowHub [23] and HuggingFace [24]. In addition to this, it is feasible to fine-tune pre-trained transformers in order to better perform one's own natural language processing jobs. The only drawback of Transformer is that training models still require a significant amount of memory and processing resources. In addition, the Transformer option is still considered as a poor solution for hierarchical data. The popularity of Transformers has revitalized the entire field of Natural Language Processing, and as a result, new language models are being introduced at a rapid pace. We can conclude that new generations of scientists will be benefited from the development of a variety of transformer subsets.

References

[1] E. D. Liddy, *Natural Language Processing*. Encyclopedia of Library and Information Science, 2001.

[2] M. Baig, *About "Course in General Linguistics" by Ferdinand de Saussure*. GRIN Verlag, 07 2010.

[3] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. October, pp. 433–60, 1950.

[4] H. A. HODGKIN AL, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *PMC, PMCID : 1392413*, 1952.

[5] N. Chomsky, *Syntactic Structures*. De Gruyter Mouton, 1957.

[6] J. McCarthy, "History of lisp," *SIGPLAN Not.*, vol. 13, no. 8, p. 217–223, aug 1978. [Online]. Available: https://doi.org/10.1145/960118.808387

[7] D. Khyani and S. B S, "An interpretation of lemmatization and stemming in natural language processing," *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, vol. 22, pp. 350–357, 01 2021.

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, *Neural Probabilistic Language Models*. JMLR.org, 05 2001, vol. 3, pp. 137–186.

[9] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," *CoRR*, vol. abs/2001.08317, 2020. [Online]. Available: https://arxiv.org/abs/2001.08317

[10] R. Cahuantzi, X. Chen, and S. Güttel, "A comparison of LSTM and GRU networks for learning symbolic sequences," *CoRR*, vol. abs/2107.02248, 2021. [Online]. Available: https://arxiv.org/abs/2107.02248

[11] P. Karmakar, S. W. Teng, and G. Lu, "Thank you for attention: A survey on attention-based artificial neural networks for automatic speech recognition," *CoRR*, vol. abs/2102.07259, 2021. [Online]. Available: https://arxiv.org/abs/2102.07259

[12] S. Chaudhari, G. Polatkan, R. Ramanath, and V. Mithal, "An attentive survey of attention models," *CoRR*, vol. abs/1904.02874, 2019. [Online]. Available: http://arxiv.org/abs/1904.02874

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[14] Peltarion. [Online]. Available: https://peltarion.com/knowledge-center/modeling-view/build-an-ai-model/blocks/multilingual-bert

[15] What is transformers? [Online]. Available: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[17] How transformers work. [Online]. Available: https://huggingface.co/course/chapter1/4?fw=pt

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *Journal of Machine Learning Research*, 2018.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[21] B. Mensa-Bonsu, T. Cai, T. Koffi, and D. Niu, *The Novel Efficient Transformer for NLP*. Springer, 08 2021, pp. 139–151.

[22] N. Broad. Esg - bert. [Online]. Available: https://huggingface.co/nbroad/ESG-BERT

[23] Tensorflow hub. [Online]. Available: https://www.tensorflow.org/hub

[24] Hugging face ai community. [Online]. Available: https://huggingface.co/