IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 6, June 2022

DOI: 10.17148/IJARCCE.2022.11698

NEXT WORD PREDICTION AND PARAPHRASING USING NATURAL LANGUAGE PROCESSING

[1]S Nithin, [2]Sameer Pandit, [3]Tanuja Shastri, [4]Yash Joshi, [5]Dr.Rashmi Amardeep

Department of Information Science and Engineering,

Dayananda SagarAcademy of Technology and Management, Bengaluru-560082[1-5]

Abstract: The use of online tools for a faster generation of essays or sentences has always been very important. We propose a one-destination website that provides users with the facilities to be able to choose between Essay generation, next-word prediction, or paraphrasing. The tools used typically help users not just with typing but also with grammatical errors, better sentence construction, and quicker more efficient outputs. Our project uses GPT, web scraping, and transformers. Pre-trained transformers have attained a state-of-the-art performance for various NLP tasks. We mainly focus on providing users with a one-stop destination for their needs related to the topic. The website saves users time having to find results for their needs. Moreover, the site helps reduce plagiarism while still providing a good result for the user's needs.

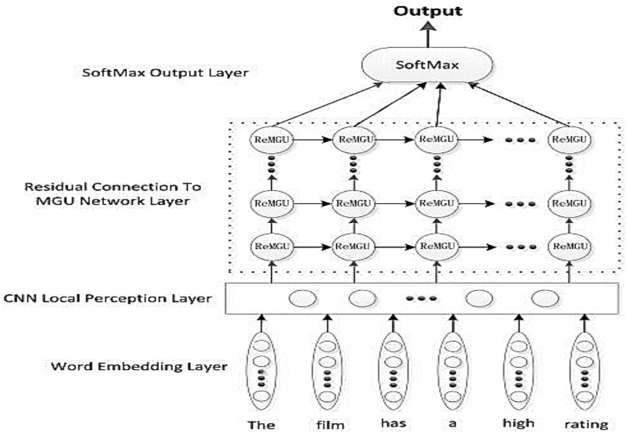
Keywords: Latent Semantic Analysis, Paraphrasing, Next word prediction, Natural language processing, BERT, transformers, LSTM, Character prediction.

I. INTRODUCTION:

Next word prediction provides user assistance while typing out a sentence. The program searches through the Internet database and compares the word in question with many older inputs by users or any other information that might be stored related to what the next word could be. On going through all the possible outcomes, the program provides options related to which word would fit the best. The user now may decide to use one of the options provided or add a new option for future reference. This helps with reducing time spent on typing out sentences.

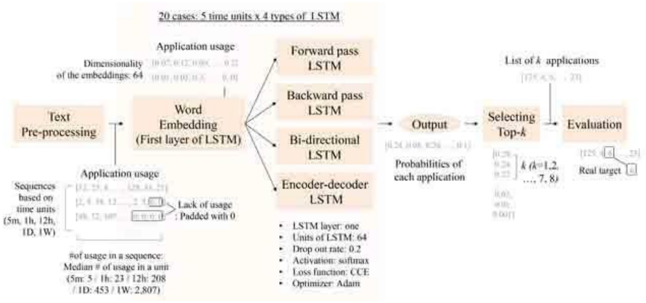
Our project's second branch deals with essay generation. This is the more complex branch out of the three. This section of the project deals with the generation of essays based on a keyword that the user inputs. The output's accuracy may vary based on the user's keyword used. The program uses web scraping to find sentences that most likely fit the description of the keyword and are out together based on other essays that contain the keyword in question. This provides a more accurate output.

Using GPT models along with web scraping and transformers the website scours through the internets database to find relations between required output and the user's input. Due to the vast size of the internets database, execution time may be stretched but only due to the program finding the best most suitable output for the user as per their requirements.



DOI: 10.17148/IJARCCE.2022.11698

Paraphrasing is the final branch for our project to be complete. This web tool is used to paraphrase specific sentences or words provided by the user. This is more used for users when they need to simplify or shorten sentences for better format. Using all these models our project deals with essay generation, next word prediction as well as paraphrasing. Using deep learning and pretraining models has resulted in accurate results regarding many different languages. In modern times, to attain the best results OpenAI GPT (Generative Pre-Training) and BERT (Bidirectional Encoder Representation from Transformers)


20 cases: 5 time units x 4 types of LSTM
Application usage
Dimensionality of the embeddings: 64
Text Pre-processing
Word Embedding (First layer of LSTM)
Application usage
Sequences based on time units (5m, 1h, 12h, 1D, 1W)
Lack of usage : Padded with 0
#of usage in a sequence: Median # of usage in a unit (5m: 5 / 1h: 23 / 12h: 208 / 1D: 453 / 1W: 2,807)
Forward pass LSTM
Backward pass LSTM
Bi-directional LSTM
Encoder-decoder LSTM
LSTM layer: one
Units of LSTM: 64
Drop out rate: 0.2
Activation: softmax
Loss function: CCE
Optimizer: Adam
Output
Probabilities of each application
Selecting Top-k
k (k=1,2, ..., 7, 8)
List of k applications
Evaluation
Real target

This method of providing users with an effective source for their needs with regard to next word prediction, essay generation, or paraphrasing will provide a positive outcome and results once implemented further.

II. LITERATURE SURVEY

Past and present research relevant to the current work is found in the more general field of transfer learning which has a specific focus on language transfer. Transfer learning typically can be an effective strategy to adopt models to lower-resource languages by training a model for a source language prior to further training (parts of) the model for a target language. In machine translation, a model can be adapted by initially training it for a high-resource language pair after which the model should be partially retrained for a low-resource language.

Now, moving on to task-oriented dialogue modeling, this requires substantial amounts of domain-specific manually labeled data. The objective maximizes the likelihood over the word sequence:

S = {w1, ..., w|S|}: L1(S) = X |S| i=1 log P (wi |w0, w1, ..., wi−1)

This formula is used when leveraging transfer learning through generative pretraining on large unlabelled corpora that come in question.
Bidirectional Encoder Representations from Transformers or BERT decomposes the input sentence(s) into Word Piece tokens. Word Piece tokenization helps improve the representation of the input vocabulary and reduce its size, by segmenting complex words into subwords.
With times changing, more recent work suggests that the Transformer implicitly encodes syntactic information in such a manner that the dependency parse trees (Hewitt and Manning, 2019; Raganato and Tiedemann, 2018), anaphora (Voita et al., 2018), and subject-verb pairings (Goldberg, 2019; Wolf, 2019).

Moving on to GPT2, its architecture closely resembles the Transformer model decoder structure. However, GPT-2 is a very large Transformer that is based on a language model that requires training on a large dataset. A team at google known as The Google Brain team introduced the Transformer, which incorporates an encoder-decoder architecture to create a sequence-to-sequence (Seq2Seq) model without the use of convolutional (CNN) or recurrent (RNN) neural networks.

III. METHODS

Parts-of-speech-tag:

Previous research reveals that the Transformer's attention heads may specialize in specific linguistic events. Individual attention heads in GPT-2 are investigated to see if they target certain sections of speech. We quantify the proportion of a given head's total attention that is focused on tokens with a certain part-of-speech tag, averaged across a corpus.

GPT2:

GPT2. Recently, deep learning and pre-training models have shown remarkable outcomes in a variety of linguistic tasks. Fine-tuning pre-trained models like Elmo (Embeddings from Language Models), OpenAI GPT (Generative Pre-

DOI: 10.17148/IJARCCE.2022.11698

Training), GPT-2, and BERT (Bidirectional Encoder Representations from Transformers) has become the ideal method for cutting-edge outcomes.

GPT-2 is the GPT's successor. Despite the fact that both GPT-2 and BERT can generate text, Wang and Cho discovered that GPT-2 generations are of higher quality. In fact, GPT-2 is said to be so powerful that it poses considerable risk of malevolent exploitation. As a result, OpenAI has decided to keep its largest model (1.5B parameters) closed so that more time may be spent debating its implications.

Using the OpenAI GPT-2 source code1 and fine-tuning the available medium-size model, we created patent claims in this study (355M). Overall, we're amazed by how complex and coherent the generated patent claims can be, though not all of the writing is of comparable quality. We were also amazed by how few training steps were required for GPT-2 to generate the first text that resembled a patent claim (one step equals one single batch/gradient update). It's only a matter of time before the largest and most powerful model is made public. As a result, it is preferable to begin early and evaluate the influence on patent research.

Web Scraping:

Web scraping, also known as web harvesting or web data extraction, is a type of data scraping that is used to gather information from websites. Using the Hypertext Transfer Protocol or a web browser, web scraping software can directly access the World Wide Web. While a software user can perform web scraping manually, the word usually refers to automated procedures carried out by a bot or web crawler. It's a type of copying in which specific data is obtained and copied from the internet, usually into a central local database or spreadsheet for retrieval or analysis later.

Web scraping is the process of retrieving a web page and extracting information from it. Fetching is the process of downloading a webpage (which a browser does when a user views a page). As a result, web crawling is an important part of web scraping, as it allows you to collect pages for further processing. After the data has been fetched, extraction can begin. A page's content can be analyzed, searched, reformatted, and the data put into a spreadsheet or a database. Web scrapers often extract information from a page in order to use it for another purpose. Finding and copying names and phone numbers, companies and their URLs, or e-mail addresses to a list is an example (contact scraping)

Some websites employ techniques such as detecting and blocking bots from crawling (viewing) their pages to prevent web scraping. As a result, web-scraping systems rely on DOM parsing, computer vision, and natural language processing algorithms to emulate human browsing and scrape web page content for offline parsing.

Transformers:

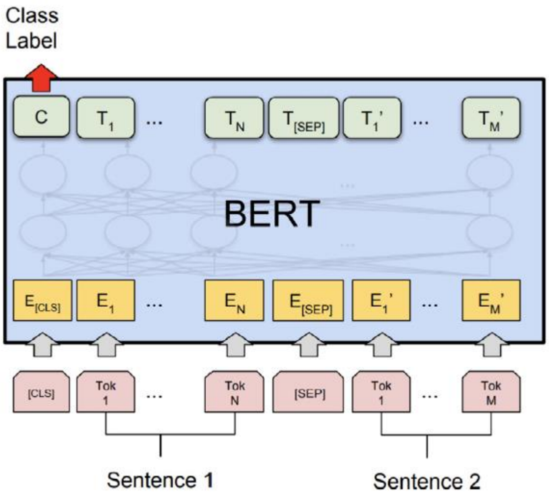
For several Natural Language Processing (NLP) tasks, pre-trained Transformer-based models have attained state-of-the-art performance.
These models, on the other hand, frequently have billions of parameters, making them excessively resource- and computation-intensive for low-capability devices or applications with severe latency requirements.

BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), XLNet (Yang et al., 2019), MegatronLM (Shoeybi et al., 2019), Turing-NLG (Rosset, 2020), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020) are all popular pre-trained These Transformers are—for example when BERT was first launched, it significantly enhanced the state of the art for eleven NLP jobs (Devlin et al., 2019)

BERT:

BERT (Bidirectional Encoder Representations from Transformers) is pre-trained utilizing two objectives: masked language model (MLM) and next sentence prediction, based on the original transformer design (NSP). MLM involves masking a percentage of the input tokens at random and predicting such tokens using the left and right context.

DOI: 10.17148/IJARCCE.2022.11698



The NSP task is a binary classification task in which the model must determine if two phrases are sequential or not. As training corpora, BooksCorpus (Zhu et al., 2015) and the English Wikipedia (16 GB in total) were employed. BERT outperformed the state of the art in eleven NLP tasks, including GLUE, SQuAD, and SWAG, at the time of publishing (Zellers et al., 2018).

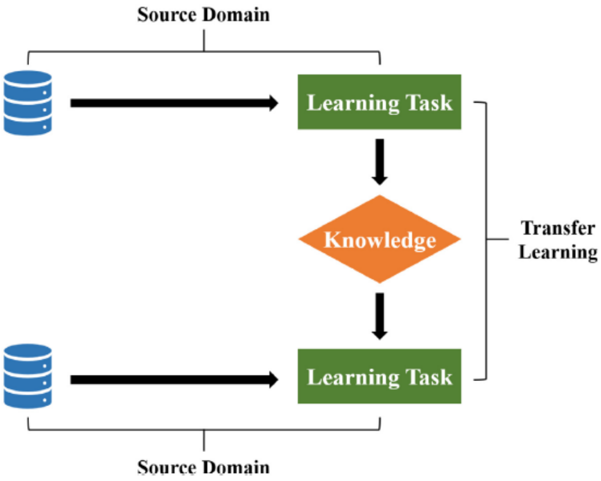
IV. PROPOSED APPROACH

Initially, we were proposed to use Latent semantic analysis (LSA) but on careful evaluation of our options to come up with software to successfully run the tasks required of it, we decided to proceed with GPT2, Parts-of-speech, and web scraping to be used. It is shown that these methods have proven to be more useful and more efficient.

Through our research, we found that most findings point towards our chosen methods being the best forms of attaining our goal for the project at hand.

With little to no drawbacks to using these methods for our project, we decided that implementing this will not only result in a more efficient and effective project being made but also ease the process of doing so along the way.

Using parts of speech, GPT2 and transformers we have seen more accurate and more logical outputs from our project. Over time with more and more updates that can potentially be made to the project, the chosen approach seemed to best fit our needs along with the user requirements.



There are other one-of-a-kind strategies that have been found to be effective for lowering the size and inference time of BERT-like models, in addition to the aforementioned methods.

Sharing of parameters. ALBERT (Lan et al., 2020) uses the same architecture as BERT, but weights are shared across all encoder units, resulting in significant memory savings. ALBERT also allows for the training of larger and more complex models: While BERT's performance peaks at BERTLARGE (BERTXLARGE's performance plummets), ALBERT continues to improve until it reaches the vastly larger ALBERTXXLARGE model

(L = 12; H = 4096; A = 64).

Squeezing the weight. The compression method of weight squeezing (Chumachenko et al., 2020) is comparable to knowledge distillation in that the student learns from the teacher. Instead of learning from intermediary outputs as in knowledge distillation, the teacher model's weights are mapped to the student via a learnable transformation, and the student learns its weights directly from the teacher.

Embedding Matrix Compression is a technique for compressing matrices. The embedding matrix is a lookup table for the embedding layer, which takes up around 21% of the total BERT model size. One option to compress it is to reduce the vocabulary size V. Remember from Section 2 that BERT's vocabulary is acquired using a WordPiece tokenizer, which uses the vocabulary size to determine how fragmented the words in the input text are.

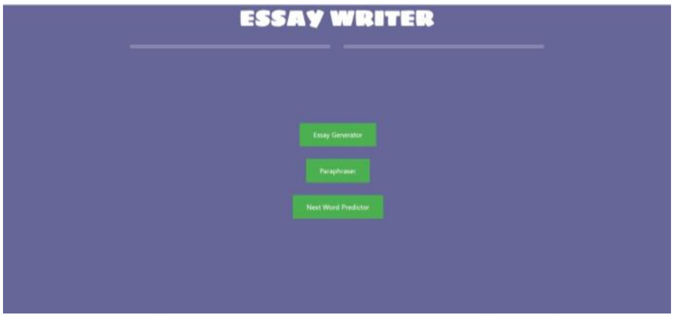
A broad vocabulary provides for more flexibility for out-of-vocabulary words and a better portrayal of unusual words. 94 percent of the tokens formed with a 5k vocabulary size match those created with a 30k vocabulary size (Zhao et al., 2019b). As a result, even with a limited vocabulary size, the majority of the words that appear frequently enough are covered, making it fair to reduce the vocabulary size to compress the embedding matrix. Another option is to replace the existing one-hot vector encoding with a "codebook-based" encoding, in which each token is represented by a combination of indices from the codebook. The token's final embedding can therefore be calculated as the sum of the embeddings found in each of these indices (Prakash et al., 2020).

VI. OUTPUT

The designed website contains a page that has the three main options a user may choose from, depending on their needs. These options are “Next Word Predictor”, “Essay Generator” and “Paraphraser”, each having its own functionalities.

When a user decides on which option, he/she would like to use and select it, they are taken to a new page dedicated to that specific operation.

The user may then enter the essay, sentence, or word based on which option they chose depending on the required output.



VI. CONCLUSION

In conclusion, the methods used for attaining the required results for a user are primarily GPT2, Transformers, and Web Scraping. On testing, we found that the results shown were up to the mark and accurate with the required output. While testing we found that Essay Generation has 86% accuracy, Paraphrasing has 78% accuracy and finally, and Next Word Prediction has 75% accuracy.

Finally, while complex and application-dependent, properly configuring the learning rate can lower the neural network's loitering time.

REFERENCES

[1] Budzianowski, Paweł, and Ivan Vulić. "Hello, it's GPT-2--how can I help you? towards the use of pre-trained language models for task-oriented dialogue systems." arXiv preprint arXiv:1907.05774 (2019).

DOI: 10.17148/IJARCCE.2022.11698

[2] Chang, Jia-Wei, Jason C. Hung, and Kuan-Cheng Lin. "Stability-enhanced lyric generator with music style transfer." Computer Communications 168 (2021): 33-53.

[3] Lee, Jieh-Sheng, and Jieh Hsiang. "Patent claim generation by fine-tuning OpenAI GPT-2." World Patent Information 62 (2020): 101983.

[4] Casola, Silvia, Ivano Lauriola, and Alberto Lavelli. "Pre-trained transformers: an empirical comparison." Machine Learning with Applications 9 (2022): 100334.

[5] van Geemert, Rene. "Use of GPT for stabilization and acceleration of search mechanisms in industrial core computations." Annals of Nuclear Energy 136 (2020): 107013.