2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. Sep 3-4, 2021

2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) | 978-1-6654-1703-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRITO51393.2021.9596520

CoVShorts: News Summarization Application Based on Deep NLP Transformers for SARS-CoV-2

Hunar Batra, Akansha Jain, Gargi Bisht, Khushi Srivastava, Meenakshi Bharadwaj, Deepali Bajaj, Urmil Bharti
Department of Computer Science
Shaheed Rajguru College of Applied Sciences for Women
University of Delhi, Delhi, India
[1]i@hunarbatra.com, [2]jakansha2001@gmail.com, [3]bisht.gargi19@gmail.com, [4]khushi.srivastava145@gmail.com
[5]minakshi29bharadwaj@gmail.com, [6]deepali.bajaj@rajguru.du.ac.in, [7]urmil.bharti@rajguru.du.ac.in

Abstract*— Amidst the grueling SARS-CoV-2 pandemic which has affected the lives of people across the world, the accelerating growth in COVID-19 related news articles is making it difficult for the general public to stay up-to-date with all the information. News articles are a crucial medium to convey coronavirus-related information across the world to the public. Short summaries of news articles can assist the public in grasping a gist of an entire article without having to read it fully. With the evolution of Deep Learning in Natural Language Processing (NLP), we exploited the power of recent advances in pre-trained and transformer-based NLP models to perform text summarization over the COVID-19 Public Media Dataset. For this, we analyzed and compared the results of BERT, GPT-2, XLNet, BART, and T5. The first three models are among the most popular extractive summarization models and the last two are abstractive summarization models. We evaluated the results of our experiments using ROUGE scores (ROUGE-2 and ROUGE-L) and found that BERT, a transformer autoencoder, outperforms the other models under consideration in SARS-CoV-2 news summarization. Thus, we leveraged BERT in our web application "CoVShorts" to summarize COVID-19 articles. Further, we visually analyzed the dataset to depict the most used words in COVID-19 news articles using *Word Cloud* to validate the accuracy of the summarization task. CoVShorts will serve the public by helping them in gaining brief, concise, and to-the-point summaries quickly.

Keywords— ***Coronavirus, COVID-19, SARS-CoV-2, Pre-Trained NLP models, Abstractive summarization, Extractive summarization, GPT, BERT, BART, T5, XLNeT, ROUGE, Word Cloud, CoVShorts.***

I. Introduction

Towards the end of 2019, a novel coronavirus identified as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) caused an outbreak of the Coronavirus disease of 2019 (COVID-19), a respiratory ailment. This virus's spread has overrun the globe, radically altering many facets of human life. Since the beginning of 2020, an enormous amount of COVID-19 news articles has been published, with several hundred new articles being published every day. This astounding amount of data production caused an information overload, making it difficult for the public to keep up with the most recent findings. Many studies were spurted in response to the outbreak to deal with the pandemic resulting in the COVID-19 information explosion. Most of the articles are related to social media analysis to reveal and understand the effects of SARS-CoV-2 on the public, while other studies are focusing on new tools that aid in keeping track of COVID-19 discoveries. According to [1], on average, 25% of the news articles published in 2020 revolved around the coronavirus epidemic, which suggests that the subject of COVID-19 has dominated, in fact, overwhelmed the online news.

As the Internet has made it quite simple and fast to access several kinds of news articles, it is becoming increasingly vital to summarize information to meet the expectations of users. The idea is to produce a concise summary, as well as retain important information content and the overall sense of the article. In general, approaches of automatic text summarization can be *extractive*, where we locate key phrases or sentences in the original text and extract only certain relevant phrases to form the summary, and *abstractive*, where a compact and clear summary of the source text that captures its important aspects using advanced Natural Language Processing (NLP) techniques is constructed [2]. Generally, abstractive summaries are closer to human-like interpretation, but extractive summaries have proved to yield better evaluations. Extractive summarizing approaches have been the dominant paradigm in the field of text summarization due to their simplicity and ease of use [3].

To extract important information among the vast and accelerating amount of news articles that are rising at an exponential rate is a major concern. The challenge here is to collect concise and clear key information from an ever-increasing number of news articles. Our research work tackles this information overload problem that is hindering the public in keeping them abreast of the latest COVID-19 related knowledge.

For this, we developed the "CoVShorts" web application that can generate coherent summaries for large news articles. To summarize COVID-19 News Articles, we compared both the abstractive and extractive summarization approaches using pre-trained transformer-based NLP models, namely - BERT, GPT-2, XLNet, BART, and T5. GPT-2, BERT, and XLNet are the extractive summarization models whereas T5 and BART are the abstractive summarization models. A huge set of articles from the "COVID-19 Public Media Dataset" were preprocessed and then run through these models to provide a summary, which was then reviewed and compared using ROUGE-2 and ROUGE-L [4]. According to the results, BERT performed exceptionally better than other models. Based on these results, we utilized BERT as the summarization model to develop our SARS-CoV-2 news articles text summarization web application. The CoVShorts generates a summary for the given news article. Thus, this application serves as highly valuable for users who rely on online news sources and want to stay up to date with more COVID-19 news with just a gist of the article.

The rest of the paper is organized as follows : We outline previous research works related to this field in section II. Section III elucidates the motivation behind our research and background study. The dataset, methodologies, and tools that we used to create our web application "CoVShorts" are described in section IV. Section V shows the results of our experiments. Section VI wraps up our research work and discusses the work scope in the future.

II. Related Work

The topic of automatic text summarization has been a field of interest for researchers in recent times and many authors contributed to this field. Nallapati et al. [5] worked on different corpora including CNN/Dailymail dataset for abstractive text summarization using Attentional Encoder-Decoder Recurrent Neural Networks outperforming the work done before them. But, soon with the introduction of the Transformer model architecture based on attention mechanisms [6], the approach to automatic text summarization was revolutionized. Miller et al. [3] presented the transformer as a base language model to enhance document summarizing outcomes substantially. This model for automatic extractive summarization was a significant improvement over previous methods in terms of quality. Both [3] and [5] are summarizing documents for all topics in general and did not target a particular subject or theme.

Due to the advent of the SARS-CoV-2 pandemic, the current state of the world necessitates a focus on COVID-19-related news articles. Hayatin et al. [7] published research work linked to the summarization of COVID-19 news articles, in which they presented transformers as a core language model for providing abstractive summaries of COVID-19 news articles, using architectural modification as the basis for constructing the model. Their work solely focused on abstractive text summarization using the MTDTG transformer model. The short descriptions used for validation did not capture the core of the COVID-19 articles of the dataset and thus were not adequate to grade the summaries generated.

We intend to contribute to the existing work by comparing both abstractive and extractive forms of automatic text summarization techniques on a dataset containing COVID-19 news articles only. We compare and evaluate these models by examining the performance of autoregressive, autoencoders, and sequence-to-sequence transformer models. This comparison helps in determining the best-suited model for providing a concise and meaningful synopsis of COVID-19 news. We aim to deploy the best-performing model and implement it on our web application: "CoVShorts: News Summarization for SARS-CoV-2 ".

III. Motivation and Background

The year 2020 brought with it a wave of pandemic - coronavirus disease caused by SARS-CoV-2. On January 30th, 2020, the World Health Organization declared the novel coronavirus to be a Public Health Emergency of International Concern and by March 11th, 2020 it was declared as a global pandemic [8]. SARS-CoV-2, as a newly identified virus, brought with it a plethora of information that people around the globe were unfamiliar with. This information is now available in the form of research articles and news articles and is easily accessible to the public. All COVID-19 articles on Scopus and Web of Science represented around 48 percent and 37 percent of research publications respectively [9]. Nevertheless, this abundance has made it difficult to read through every piece of information.

The objective of our research is to develop a platform where news articles that are elongated and verbose can be condensed into precise summaries containing only the relevant facts from the article. The primary focus is kept on news articles related to the coronavirus pandemic because it is a new area of study with a new set of terminology and vocabulary, necessitating a higher amount of attention to produce meaningful summaries. The motivation for this piece of research is the paucity of recent work aimed at summarizing news articles, particularly about coronavirus disease. This will facilitate people to gather more about the global pandemic without having to spend hours in reading long and tedious news articles. For our application CoVShorts, we explored both extractive and abstractive text summarization approaches.

In extractive text summarization, transformer models locate and concatenate the parts of the sentences that seem more relevant for the summary. We selected Bert Extractive Summarizer which puts the HuggingFace PyTorch transformers [3] to identify the phrases of paramount importance. To perform summarization, it first generates sentence embeddings, and then by applying K-means clustering, the sentences closest to the centroid are identified which are further used for the final summary.

Abstractive text summarization has a different approach which parses the source text to provide summaries with novel terms or phrases not from the source content. Here, the problem is considered as sequence-to-sequence; the encoder maps a sequence of tokens in the input $X = (X_1, X_2, \dots, X_M)$ of size M to a sequence of continuous representations $Z = (Z_1, Z_2, \dots, Z_M)$ and a decoder then produces the objective summary $Y = (Y_1, Y_2, \dots, Y_N)$ of size N token-by-token where $N < M$ [10]. It is observed that the summary generated is smaller in size and closer to a human-generated summary in comparison to the extractive models.

Both techniques have their strengths and weaknesses. Abstractive summarization is more allied with a human-like interpretation. But it is quite arduous to get an informative summary and consequently requires extensive training. Extractive summarization selects its words and phrases from the input resulting in a more intact and accurate result. But it will be considered useful only if the sentences extracted from the input text reflect the full information shown in the input [10].

IV. Methodology and Implementation

In this section, we will go through the dataset utilized, the preprocessing done on the dataset, selection of five pre-trained models used for our experiment, and deployment of the CoVShorts application, which is implemented to generate appropriate summaries, in further detail.

A. Dataset and Preprocessing

The "COVID-19 Public Media Dataset" used in this study is taken from the Kaggle platform, published by Anacode

[11]. This dataset is a collection of over 350,000 full-text internet articles gathered from online media between January 1 to December 31, 2020. This dataset features a collection of non-medical news articles related to SARS-CoV-2 which have been gathered from over 60 high-traffic blogs and news sites. The news sources which have been utilized to accumulate articles for the corpus consist of a range of media categories and it intends to showcase a representative cross-section of online content generated and accessed during this time.

Preprocessing of a dataset, according to the findings of several recent studies, can improve the accuracy of results by 2% [12]. We refined the dataset during the preprocessing stage by first lowercasing the entire dataset, then removing punctuation marks, nonsensical words, misspellings, or phrases such as HTML(Hypertext Markup Language) tags (which were included in the dataset while scraping data from websites) and embedded links.

B. NLP Model Selection

For NLP problems, transformers quickly became the model of choice, replacing older Recurrent Neural Network (RNN) models such as Long-Short Term Memory (LSTM) [13]. As in previous models, the transformer has an *encoder-decoder* architecture. Each encoder's function is to generate encodes that provide information as to which parts of the inputs are mutually relevant. The codes are passed to the next layer of the encoder as inputs. Every decoder layer does the opposite and takes all encodings to produce an output sequence with its integrated contextual information [6].

We examine the following five pre-trained transformer-based models for our research: BERT, GPT-2, XLNet, T5, and BART.

1) BERT: Bidirectional Encoder Representations from Transformers (BERT) is a neural network-based methodology for pre-training NLP [14]. It is an autoencoder-based transformer (uses only the encoder block of the transformer). BERT is both theoretically and quantitatively simple.

2) *GPT-2:* Generative Pre-trained Transformer 2 (GPT-2), an autoregressive transformer model, is an open-source artificial intelligence model developed by OpenAI. It is a 1.5 billion parameter transformer-based language model that was trained on a dataset of 8 million Web pages [15]. With 1024 tokens and a vocabulary of 50,257 words, it is built using transformer decoder blocks and has 4 model sizes consisting of different architectural hyperparameters - small, medium, large, and XL, with 124M, 355M, 774M, and 1.5B parameters, respectively. Because it was trained with a Causal Language Modeling (CLM) objective [16], it can predict the next token in a sequence.

3) XLNet: XLNet uses a non-traditional autoregressive training technique. Before permitting the model to predict the token n+1 based on the last n tokens, it permutes the tokens in the sentence. Rather than masking the first n tokens for n+1, XLNet provides a net that masks the preceding tokens in a certain permutation of 1, 2,..., Sequence Length [17]. The fundamental addition of XLNet is not the architecture, but a changed language model training that aims to learn conditional distributions for all permutations of tokens in a sequence.

4) T5: Text-To-Text Transfer Transformer (T5) is a pre-trained language model that stands out for using a unified “text-to-text” format for all text-based NLP tasks. It was first proposed by Vaswani et al. (2017) [18], where initially each task was converted to text-to-text format and then pre-trained on a variety of supervised and unsupervised activities [18]. This method is well-suited for generative tasks (such as machine translation or abstractive summarization) in which the model must generate text based on some input [19].

5) *BART:* Bidirectional and Auto-Regressive Transformer **(**BART), a denoising autoencoder, is used for pre-training sequence-to-sequence models. It is like the original Transformer model for neural machine translation, but with some differences from BERT (which only uses the encoder) and GPT (which only uses the decoder). The pre-training task entails changing the sequence of the original phrases at random and using a novel in-filling strategy that replaces text spans with a single mask token. It is trained by corrupting text using a random noising function and then building a model to recover the original text [20].

C. CoVShorts Implementation

We designed the CoVShorts web application using Flask, a micro web framework [21]. Fig 1 shows the complete CoVShorts Architecture. We deployed BERT, the best performing pre-trained NLP transformer model in our application. The application is powered by Google Colaboratory (Google Colab) GPU (Graphic Processing Unit) environment, a completely cloud-based Jupyter notebook environment. Colab is a low-cost notebook that supports popular libraries and does not require any setup.

As Google Colab has a limited amount of RAM and storage space, the dataset was split into eight subsets, each with a size of 100 articles. We used the NVIDIA Tesla T4 with around 15 GB of RAM in our Colab's environment. All the tests were performed within Google Colab's 12-hour limitation, which was sufficient for our runtime sessions.

The summarization model is running in a Colab environment which is using GPU and has been deployed with the Flask web framework using *ngrok*.

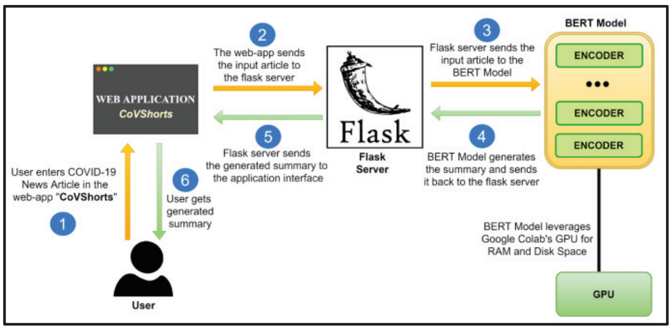


Fig. 1. Detailed Architecture of CoVShorts application

The User Interface of the CoVshorts application with a news article and its summary generated along with their word count is shown in Fig 2.

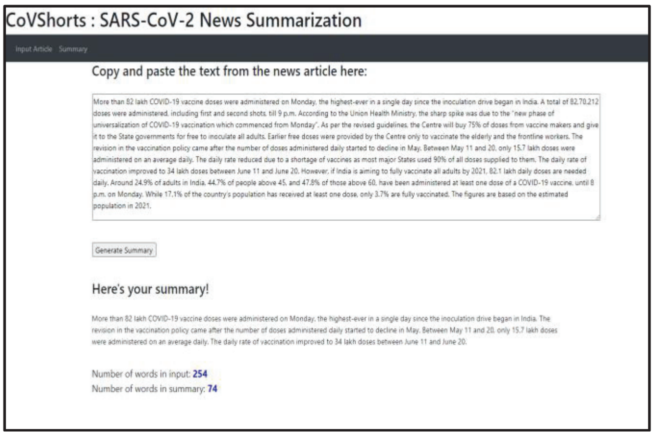


Fig. 2. User Interface of "CoVShorts" Web application for News article summarization

V. Results

This section presents the evaluation results of selected five pre-trained NLP models based on ROUGE scores. The quality of summaries generated by CoVShorts is assessed by Word Cloud.

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a software package and a metrics set (ROUGE-N, ROUGE-L, ROUGE-S, ROUGE-SU) that spontaneously determines the qualities of the generated summary by correlating it with the human-generated summary [4]. It is the most dependable and steady approach to evaluate the effectiveness of the summary generated by the model.

We instituted ROUGE score metrics to evaluate the news summaries generated by five pre-trained NLP transformer models. As the COVID-19 is relatively a new field of study for automatic text summarization using NLP, thus there is an absence of open-source datasets for COVID-19 news articles with gold standard summaries that could be used to evaluate the performance of automatic text summarization. Hence the notion of input-summary similarity is used to solve this problem in our work. Thus, the more similar a summary is to the input, the better its substance is [22]. For our experiments, ROUGE-2 and ROUGE-L scores were evaluated between the summary generated by CoVShorts and its article as the reference summary. ROUGE-2 achieved finer results among the ROUGE-N variants for moderate to large summaries when stopwords (commonly words used that provide very little information) were eliminated, so we decided to move forward with it [4].

ROUGE-L uses Longest Common Subsequences (LCS) for measuring the overlong matching sequence [4]. ROUGE-L comes in two flavors: Sentence-level LCS and Summary-level LCS [4]. We used Sentence-level LCS to compare the models in our work. The advantage of Sentence-level LCS is that it uses in-sequence matches rather than uninterrupted continuous matches for considering sentence-level word order.

We tested the performance of the chosen models on the "COVID-19 Public Media Dataset". For this, we extracted 800 news articles from the corpus and disjointed them in the range of 100 articles each (i.e 0-100, 100-200, 200-300, 300-400, 400-500, 500-600, 600-700, 700-800). Then for each set, we calculated the ROUGE-2 and ROUGE-L cumulative F measure scores for all five respective models.

$$\text{F-measure} = \frac{(\beta^2+1).P.R}{\beta^2.P+R} \quad (1)$$

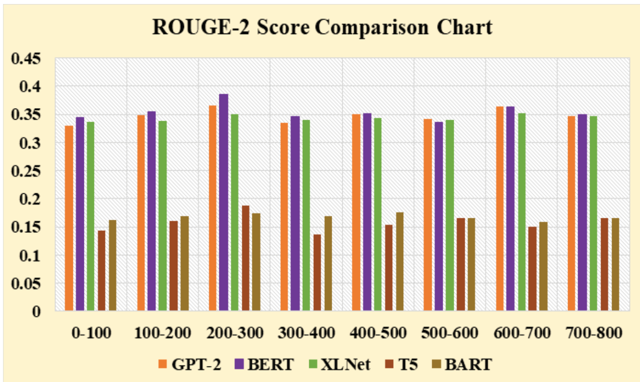
The number of overlapping words divided by the total number of words in the reference summary is known as Recall, and it indicates how much of the reference summary is captured by the summary generated [23]. The number of overlapping words in the generated summary is divided by the total number of words is calculated as Precision. It essentially determines how much of the generated summary was relevant. F-measure combines precision and recall as shown in (1) where β is the ratio of Precision (P) and Recall (R).

Results of our experiments demonstrate that extractive summarization models like GPT-2, BERT, and XLNet perform better and faster as shown in Table 1 in comparison to the other two models.

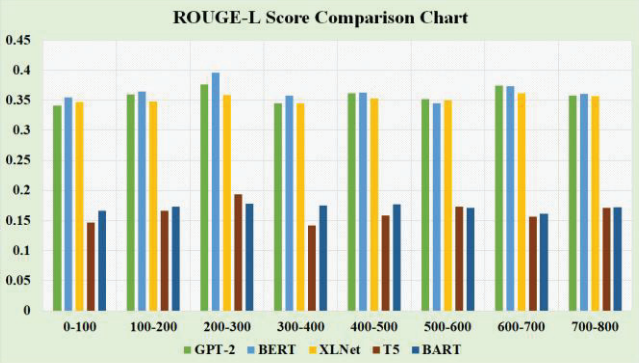
TABLE I
ROUGE F-MEASURE FOR NLP MODELS

Model	ROUGE-2	ROUGE-L
GPT-2	0.348	0.358
BERT	0.354	0.364
XLNet	0.343	0.352
T5	0.158	0.163
BART	0.167	0.172

While considering autoregressive transformer models, GPT-2 performed better than XLNet and on considering Seq2Seq transformer models, BART outperformed T5.



(a)



(b)

Fig. 3. Comparison of cumulative (a) ROUGE-2 F-measure (b) ROUGE-L F-measure

From Fig. 3, we can articulate that there is a tough fight amongst the three models namely GPT-2, BERT, and XLNet; BERT is performing well whereas T5 and BART are lagging. Hence, BERT proved that it is straightforward and experimentally dominant in comparison to other summarization models.

For CoVShorts summary validation, we used Word Cloud [24]. It is an effective visualization technique that is a simple and captivating way of representing the word frequency for a text. The size of each word depends upon the frequency or the total number of times the word appeared. They even exhibit the graphical representation of the data obtained from the text. The Word Cloud is also used on our dataset to find the most occurring words. It delivers an "overall image" of the summaries produced by BERT. It helped us to analyze the generated summaries to check whether they hold the essence of the COVID-19 topic or not. Fig 4 shows the Word Cloud output on the "COVID-19 Public Media Dataset" [11].

Fig. 4. Word Cloud output

Fig. 5 shows the most frequent keywords that appear which in turn indicates that the summaries produced by BERT are not straying away from the COVID-19 theme.

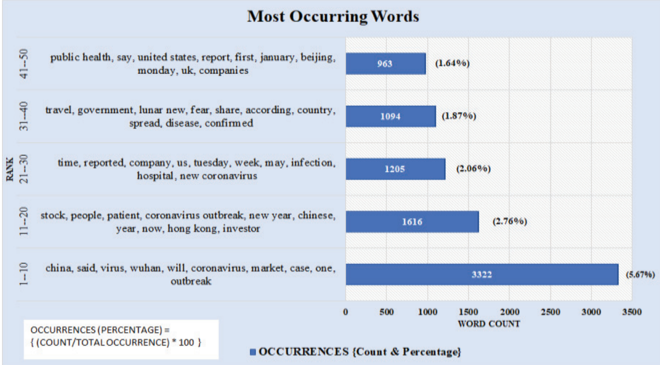


Fig. 5. Most occurring words in the summary generated by BERT

VI. Conclusion

With the evolution of Deep Learning, news article summarization has grown more advanced and efficient. For text summarization of COVID-19 news articles, we examine the performance of five NLP transformer models, three of which are extractive and the other two are abstractive using ROUGE score metrics. Among all the models tested, BERT, an extractive text summarization model produced the summary with the highest ROUGE score. Overall, the extractive NLP models outperformed the abstractive ones. BERT, an auto-encoding model, provided better results than the autoregressive and Seq2Seq-based other proposed models, owing to the reconstructing side of auto-encoders from the reduced encoding. Using Word Cloud, we reconfirmed that the summary generated by CoVShorts is concise and not deviating away from the actual theme of the news article. In the future, this project could be taken a step further by developing a mobile application for the user's convenience. In addition, rather than pasting the news article as input, the user can directly enter the URL for the news article, and the application will extract the article for summarization. To generate even better summaries, various other NLP models can also be incorporated.

References

[1] Krawczyk, Konrad, Tadeusz Chelkowski, Swapnil Mishra, Denise Xifara, Benjamin Gibert, Daniel Joseph Laydon, Seth Flaxman et al. "Quantifying the online news media coverage of the COVID-19 pandemic." *medRxiv* (2020).

[2] Sethi, Prakhar, Sameer Sonawane, Saumitra Khanwalker, and R. B. Keskar. "Automatic text summarization of news articles." In *2017 International Conference on Big Data, IoT and Data Science (BID)*, pp. 23-29. IEEE, 2017.

[3] Miller, Derek. "Leveraging BERT for extractive text summarization on lectures." *arXiv preprint arXiv:1906.04165* (2019).

[4] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In *Text summarization branches out*, pp. 74-81. 2004.

[5] Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. "Abstractive text summarization using sequence-to-sequence rnns and beyond." *arXiv preprint arXiv:1602.06023* (2016).

[6] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762* (2017).

[7] Hayatin, Nur, Kharisma Muzaki Ghufron, and Galih Wasis Wicaksono. "Summarization of COVID-19 news documents deep learning-based using transformer architecture." *Telkomnika* 19, no. 3 (2021).

[8] Velavan, Thirumalaisamy P., and Christian G. Meyer. "The COVID- 19 epidemic." *Tropical medicine & international health* 25, no. 3 (2020): 278.

[9] "COVID-19 research update: How many pandemic papers have been published?" IOP Publishing NatureIndex https://www.natureindex.com/news-blog/how-coronavirus-is-changing-research-practices-and-publishing (Accessed 4th July 2021).

[10] Liu, Yang, and Mirella Lapata. "Text summarization with pretrained encoders." *arXiv preprint arXiv:1908.08345* (2019).

[11] Janna Lipenkova "The COVID-19 Public Media Dataset by Anacode" https://www.kaggle.com/jannalipenkova/covid19-public-media-dataset (Accessed 7th June 2021)

[12] Ruhwinaningsih, Lira, and Taufik Djatna. "A sentiment knowledge discovery model in Twitter's TV content using stochastic gradient descent algorithm." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 14, no. 3 (2016): 1067-76.

[13] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM neural networks for language modeling." In *Thirteenth annual conference of the international speech communication association*. 2012.

[14] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[15] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners." *OpenAI blog* 1, no. 8 (2019): 9.

[16] Lample, Guillaume, and Alexis Conneau. "Cross-lingual language model pretraining." *arXiv preprint arXiv:1901.07291* (2019).

[17] Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. "Xlnet: Generalized autoregressive pretraining for language understanding." *arXiv preprint arXiv:1906.08237* (2019).

[18] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." *arXiv preprint arXiv:1910.10683* (2019).

[19] Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. "mT5: A massively multilingual pre-trained text-to-text transformer." *arXiv preprint arXiv:2010.11934* (2020).

[20] Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).

[21] Grinberg, Miguel. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[22] Louis, Annie, and Ani Nenkova. "Automatically assessing machine summary content without a gold standard." *Computational Linguistics* 39, no. 2 (2013): 267-300.

[23] Kavita Ganesan, "An intro to ROUGE, and how to use it to evaluate summaries", https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/ (Accessed 26th June 2021)

[24] Heimerl, Florian, Steffen Lohmann, Simon Lange, and Thomas Ertl. "Word cloud explorer: Text analytics based on word clouds." In *2014 47th Hawaii International Conference on System Sciences*, pp. 1833-1842. IEEE, 2014.