<u>CSCE 5290:</u> Natural Language processing
Project Increment -1

Project Title: Text Summarization of COVID-19 Articles using various NLP methods.

Team Members:

1. Harsha Vardhana Buddana [11521994]
2. Roshan Sah [11574385]

Motivation:

Covid is a one of the kind new age viruses that shocked the world. We had very few resources that could tell us the sources, effects and handling this crisis. But our researchers were on it and started digging more about the treatment of this dangerous virus. As the researcher increases the covid related analysis published number also increases. But one cannot go through all these papers for his solution. Hence, we came up with the summarization model which summarizes the COVID-19 related articles which can answer the questions by summarizing the abstracts of the papers.

Objective:

To gather articles that include text summarization terms that focus on the medical domain of Covid-19 words.

Significance:

Not everybody has time to go through the complete research paper for their questions. Hence, this model would be a great tool for people who look for fast and effective solutions to their questions. Also, under developing or developing countries doesn't have the resources like hospitals and doctors. Hence, this would be a great tool to know effective answers for their questions without any misinformation.

Approach:

Unsupervised system for comprehending scientific literature that accepts questions in natural language with a focus keyword and retrieves precise responses from the CORD19 corpus of scientific papers.

Related Work

Extractive Summarization

Tan et al. [7] employed pre-trained BERT and GPT2 to synthesize Corona-related article summaries. Unsupervised extractive summarization and abstractive summarization make up the model's two components. A pre-trained BERT model is used for the first half, and the GPT-2 model for the second. The sentences are transformed into sentence embedding in the first stage using a pretrained BERT model. The set of sentence embeddings is then subjected to a k-medoid clustering to produce a set of cluster centers. An extractive summary is formed from the preceding sentences. Then, from the extractive summary, a number of keywords are extracted using POS-tagger. The GPT2 model is given keyword-reference summary pairs, and upon training, system summaries are produced.

A recurrent neural network-based extractive summarization is suggested in this study. The extractive technique locates the text's informative passages. For evaluating sequences like text, recurrent neural networks are particularly effective. Sentence encoding, rating of sentences, and compilation of summaries are the three stages of the suggested methodology. An approach called coreference resolution is utilized to enhance the performance of the summarization system. Identifying mentions in the text that relate to the same thing outside the text is known as coreference resolution. Finding the main theme of the text through this technique aids in the summarizing process. [1]

Lakshmi Krishna et al. [11] compared pre-processing and model building for extractive summarization that were carried out for a small number of documents. BERT, Text rank, and GPT2 algorithms have been used to clean the text as much as possible. The performance of the GPT-2 algorithm produced the best results out of these models. This study is entirely reliant on pre-trained models; hence, training the model may result in a higher ROUGE score via better encoding representations of nodes.

Awane Widad et al. [15] presented a Question Answering tool based on BERT fine-tuned on the SQuAD benchmark. This tool takes CORD-19 dataset and uses 'Anserini', an open-source information search toolbox built around Lucene. This tool retrieves the relevant paragraph to the given question. Then this relevant set of paragraphs are sent to the BERT Text summarizer and then we use BERT pre-trained on SQuAD for answering of the questions.

Abdullah Javaid Chaudhry et al. [2] explored if "Termolator", a tool for extracting characteristic terms for a domain can be used to enhance the performance of extractive summarization approaches. They have used multiple approaches such as modified versions of TF-IDF vectors, BERT, modifications pf Word2Vec, K-Means clustering, Lex Rank, and template summarization. Evaluated the models with ROUGE family of metrics and concluded that the usage of characteristic terms for a domain as found by "Termolator" improves the performance of extractive summarization approaches with regards to the F-score. A model known as CAiRE-COVID has been presented by Su et al. [13].

The three major modules of CAiRE-COVID are information retrieval, question-and-answer, and summarization. After receiving a user query, the information retrieval module retrieves the top n most pertinent paragraphs. The most pertinent sentences found in the preceding stage are listed as the answer by the question-answering module. To choose the pertinent sentences from each of then paragraphs as the responses to the question, the question-answering module is applied to each of the n paragraphs. The top k paragraphs are then specified after these n paragraphs are once more reranked in accordance with the high-lighted replies. These k paragraphs are provided to the summarizer module, which then uses them to produce an extractive summary and an abstractive summary. The abstractive summary is produced using the UniLM and BART models, and the extractive one is produced using the cosine similarity of the sentences to the query.

Abstractive Summarization:

C Limploypipat et al. [8] In this article, they described how an LSTM neural network was used to abstractly summarize Covid-19 news. Also incorporate an attention mechanism into the encoder decoder neural network to help it focus on particular words and perform better. They produce training data sets with data augmentation and testing data sets from COVID-19 CBC News stories for our experiments. The early findings of the studies demonstrate that summarization can produce shorter paragraphs that are succinct and simple for readers to understand.

In order to help overworked medical professionals locate reliable scientific information, Andre Esteva et al. [3] introduced a tool called CO-Search, a semantic, multi-stage search engine. CO-Search is intended to handle sophisticated searches across the COVID-19 literature. The two sequential components that make up CO-Search are a hybrid semantic-keyword retriever, which uses an input query to provide a sorted list of the 1,000 documents that are the most relevant, and a re-ranker, which further ranks the documents by relevance. Each document receives a relevance score from the re-ranker, which is determined by comparing the results of an abstractive summarization module with a question-answering module that measures how well each item responds to the query.

Shengli Song et al. [12] proposed an LSTM-CNN based ATS framework (ATSDL) that can construct new sentences by exploring more fine-grained fragments than sentences. ATSDL is composed of two main stages the first one which extracts phrases from source sentences and the second generates text summaries using deep learning. LSTM-CNN based ATS framework, named ATSDL. We apply LSTM model that was originally developed for machine translation to summarization and combine CNN and LSTM together to improve the performance of text summarization. After training, the new model will generate a sequence of phrases. This sequence is the text summary that is composed of natural sentences. (ii) In order to solve the key problem of rare words, we use phrase location information, so we can generate more natural sentences. (iii) The experiment results show that ATSDL outperforms state-of-the-art abstractive and extractive summarization systems on both two different datasets.

Deep Learning for Text Summarization:

Hayatin et al. [4] offered transformers as a core language model for producing abstractive summaries of COVID-19 news articles, utilizing architectural modification as the basis for developing the model, in research work related to the summarizing of COVID-19 news articles. They only used the MTDTG transformer model for abstractive text summarization in their research. The short summaries utilized for validation were insufficient to evaluate the summaries produced since they failed to capture the essence of the COVID-19 articles of the dataset.

Milad Moradi et al. [9] proposed an innovative method for summarizing that makes use of contextualized embeddings produced by the Bidirectional Encoder Representations from Transformers (BERT) model, a deep learning model that recently displayed cutting-edge outcomes in a number of natural language processing tasks. To find the most pertinent and instructive sentences within the input documents, they mix various BERT iterations with a clustering technique and compared the summarizer to a number of methods that have been previously reported in the literature using the ROUGE toolbox.

For extractive summarization, Rezaei et al. [10] used two deep learning architectures. Performing feature extraction and creating a feature-sentence matrix for the text sentences is the initial stage. Some of the most crucial sentence characteristics for text summarization are extracted at this stage, including sentence position, sentence length, TF-IDF, and title similarity. The Auto-Encoder neural network and the Deep Belief Network are the next two neural network types to receive this matrix as input. These networks augment the matrix. The sentence scores are calculated using this matrix, and the most significant and high-scoring sentences are chosen to be included in the summary.

A mechanism termed deepMINE has been proposed by Joshi et al [5]. The two primary components of this system are the Mine Article and the Article Summarization. The user enters the necessary keywords in the first section, and the system searches the article titles provided by CORD-19 to return related articles and links. The second component uses deep learning and natural language processing to summarize an input article.

Dataset:

In order to develop a treatment and preventative measures against the COVID-19 [14], the scientific literature needs to be surveyed by the global health and research community. The COVID-19 Open Research Dataset (CORD-19) was created by the White House and top research organizations in response to this challenge in order to bring in the NLP expertise to help uncover the solution within the literature or provide insights to the general public. Over 59,000 research articles, including over 47,000 full-text articles about the COVID-19 or associated disorders, are included in this dataset. The dataset contains research papers from way before 2020. Hence, we segmented the dataset and dropped the research papers that are before 2020.

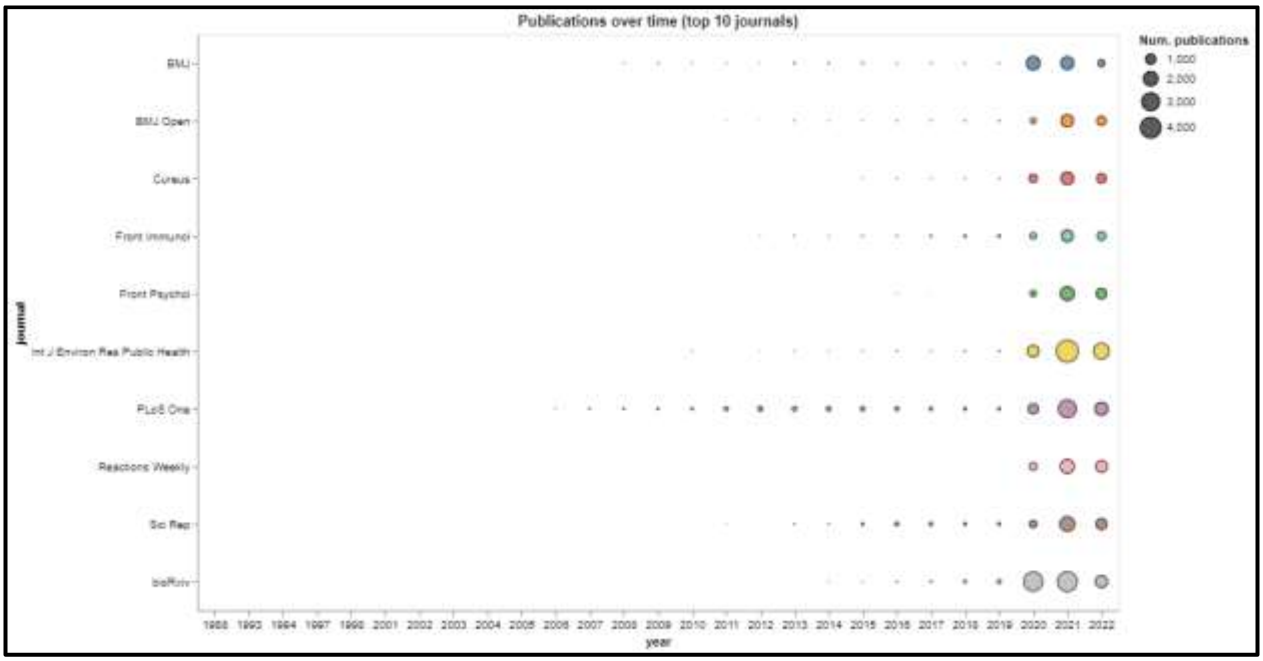

Fig 1: Distribution of Papers

In the above figure, it showcases the distribution of the research papers over the years. Then we segmented the dataset in such a way that we only retained the research papers from year 2020 and dropped the remaining.

Detail design of Features:

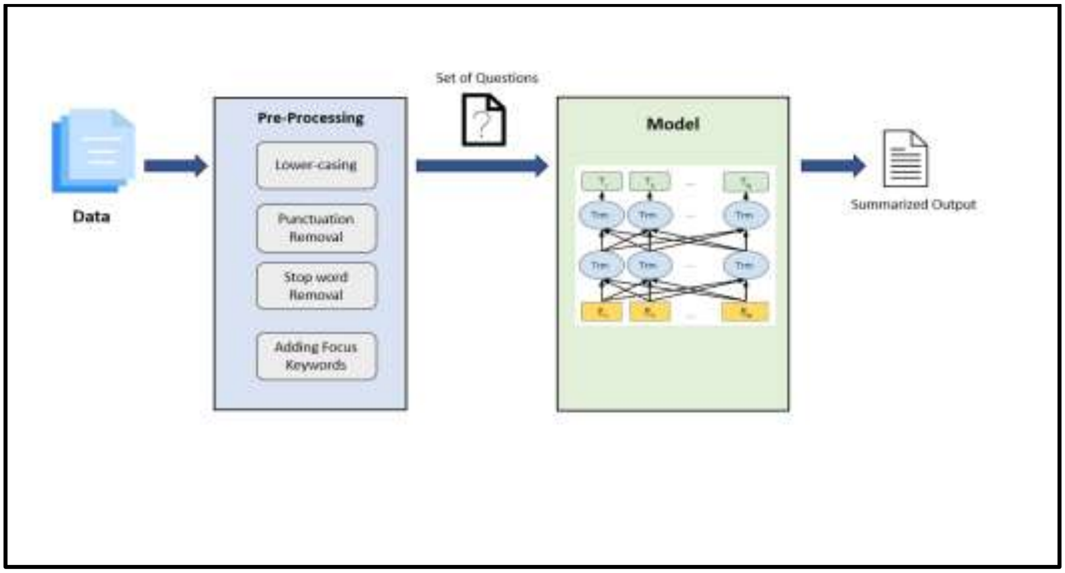

Fig 2: Data Pre-processing/Tokenization Methods

Data preprocessing is a vital step in building a Machine Learning/Deep Learning model. The quality of the preprocessing determines the model's performance [6]. There are various techniques that can be employed to clean the raw text we have. We chose to build a Text pre-processing pipeline in which we firstly lower-case the corpus we have at hand for uniformity and eliminating the punctuation marks and then we get rid of the most frequent stop words that do not add great significance to the context of the text. Stemming and Lemmatization are few very well-known text pre-processing methods, but in- stead of doing it manually. We employed the BERT models to the Stemming and Lemmatization. After these basic text pre-processing techniques. We have added Focus Keywords which can focuses specifically on few things related to the COVID and its symptoms. We included these specific keywords because the dataset is huge and has many research papers. To keep our focus on few topics would make our search and summaries precise. As part of cleaning the data, we dropped Null value columns, duplicate titles and consider research papers from the year 2020. We create a data frame in which we try to hold the abstracts of the papers which contains terms related to the COVID and its symptoms. Created a Data Frame such that, the data frame contains the abstracts of the paper which focuses on the given focus words. Later we used sentence similarity from the SpaCy library to calculate the similarity of the given sentence to that of the summarized answer.

Analysis

(463005, 7)
(275107, 7)

	sha	title	doi	abstract	publish_time	authors	journal
10	2ecd2df3b19e54d13d8877422fca3328f0fc5256	COVID-19 Vaccine Priority for People With Neur...	10.14740/jnr665	no data providedcovid-19 vaccine priority for ...	45402	Pfeffer, Gerald; Jacob, Sarah; Preston, Jeffrey	J Neurol Res
13	746889ad718cbc5154785a88863282e1d31c71c1	A 'Multiomic' Approach of Saliva Metabolomics,...	10.1016/j.gastha.2021.12.006	background and aims the sars-cov-2 pandemic ha...	44926	Pozzi, Chiara; Levi, Riccardo; Braga, Daniele;...	Gastro Hep Advances
14	c42b3898a5825cca3f5d59df41aaaeabe54827c2	Chapter 8 Infectious Disease Emergencies	10.1016/b978-0-323-83375-2.00008-5	covid-19 has added new relevance to the relati...	44926	Aliyeva, Gulnara Davud	Rapid Response Situations
15	d64935bc0335df545954b34b3bbf8ca249dfd73	Two viruses, one prescription: slow down	10.1016/j.trpro.2021.12.034	the covid-19 pandemic has devastated communiti...	44926	Tolley, Rodney; Tranter, Paul	Transportation Research Procedia
16	0268b9f3837b8e568b3ec59c69b285b41d725d69	A Workspace Typology for Enterprise Collaborat...	10.1016/j.procs.2021.12.017	the global covid-19 pandemic and the need for ...	44926	Bahles, Sebastian; Schwade, Florian; Schubert,...	Procedia Computer Science

Fig 3: Sample of dataset

The above figure showcases the snapshot of the dataset. The dimensions of the original dataset are (463005,7). The dataset is so huge for the BERT model to run on the local machine and as we do not have GPU machine, we segmented the dataset to run the model comfortably and after dropping the articles the new segmented dataset is of size 275107

Implementation & Results

Text summarization using BERT

BERT is a free and open-source machine learning framework for natural language processing. BERT uses the surrounding text to provide context in order to help machines understand the meaning of ambiguous words in text. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with question-and-answer datasets.

Transformer, an attention mechanism that recognizes contextual relationships between words in a text, is used by BERT. Transformer's basic design consists of two independent mechanisms: an encoder that reads the text input and a decoder that generates job predictions. Only the encoder mechanism is required because BERT's aim is to produce a language model.

After all the pre-processing steps, we created a set of questions that was fed to model. The model takes in the set of questions and tries to search for the relevant research papers from the dataset using the keywords. It retrieves the abstract of the related research papers and then tries to summarize all these abstracts and make a summarized answer.

```
# QUESTIONS

search=[
'What is the effectiveness of drugs being developed and tried to treat COVID-19 patients?',
'Clinical and bench trials to investigate less common viral inhibitors against COVID-19 such as naproxen clarithromycin, and minocyclinethat that may
'How are potential complications of Antibody-Dependent Enhancement ADE in vaccine recipients being researched?',
'Exploration of use of best animal models and their predictive value for a human vaccine',
'Capabilities to discover a therapeutic not vaccine for the disease, and clinical effectiveness studies to discover therapeutics, to include antivira
'Alternative models to aid decision makers in determining how to prioritize and distribute scarce, newly proven therapeutics as production ramps up a
'What research and work is being done to develop a universal vaccine for coronavirus',
'What work and research has been done to develop animal models and standardize challenge studies',
'What work and research has been done to develop prophylaxis clinical studies and prioritize in healthcare workers',
'Approaches to evaluate risk for enhanced disease after vaccination',
'Assays to evaluate vaccine immune response and process development for vaccines, alongside suitable animal models in conjunction with therapeutics'
]

# MAIN FOCUS KEYWORDS

focus=['drugs','drugs','antibodies','animal model','therapeutic','models','vaccine','model','drugs','vaccine','animal']
```

Fig 4: Set of questions and Keywords

In the above figure, we created a set of questions. Based on these questions and the keywords the BERT model searches the articles with the best matching abstracts using the cosine similarity score.

What is the effectiveness of drugs being developed and tried to treat COVID-19 patients?

Summarized Answer:

This study aimed to compare clinical outcomes between mild covid-19 patients receiving antiviral drugs and those without. Health-care systems are using repurposing drugs to cure the patients from this infection. Using drugs to treat covid-19 symptoms may induce adverse effects and modify patient outcomes.

results limited to 5 for ease of scanning

pub_date	title	excerpt	rel_score
44408	Antiviral treatment could not provide clinical benefit in management of mild COVID-19: A Retrospective Experience from Field hospital	This study aimed to compare clinical outcomes between mild covid-19 patients receiving antiviral drugs and those without. Conclusion: antiviral treatment could not provide superior clinical outcomes to supportive care in mild covid-19 patients.	0.847536
44042	Designing therapeutic strategies to combat severe acute respiratory syndrome coronavirus-2 disease: COVID-19.	Drug candidates currently under consideration and undergoing clinical trials for covid-19 treatment are highlighted.designing therapeutic strategies to combat severe acute respiratory syndrome coronavirus-2 disease: covid-19.	0.845013
44563	In silico study of remdesivir with and without ionic liquids having different cations using DFT calculations and molecular docking	Health-care systems are using repurposing drugs to cure the patients from this infection.	0.840591
44511	P025 Risk assessment of covid 19 in patients with Juvenile Idiopathic Arthritis	Patients with inflammatory rheumatic diseases undergoing immunosuppressive treatment are considered immunocompromised.	0.840143
44228	Effectiveness of early therapeutic intervention in phases one and two after COVID-19 infection: systematic review.	Conclusion studies have reported that effective drugs for treating covid-19 exist.	0.839931

Clinical and bench trials to investigate less common viral inhibitors against COVID-19 such as naproxen clarithromycin, and minocyclinethat that may exert effects on viral replication

Summarized Answer:

Several antibody drugs have successfully entered clinical trials and achieved impressive therapeutic effects. The antiviral drugs affecting viral replication and those modulating the immune response, reduce the infected cells and viral load significantly.

results limited to 5 for ease of scanning

Fig 5: Output of given Questions

We set to retrieve only the top 5 titles with the highest relevancy score and from the above figure we can see that we get the highest relevancy score of **84.75%**. And we can see our set of keywords in the abstract so we can clearly say that our model is also working based on the keywords. As part of our evaluation metrics, we included Cosine Similarity. The questions set compares with the each abstract of the articles and get the similarities score.

Implementation Status Report

Work Completed

We have created a Question answering and Text summarizing using the BERT model.

- Harsha – Related Work, Analysis
- Roshan – Dataset, Detail Design of features

We both worked equally on Implementation and Results.

Work to be Completed

We are planning to compare this BERT model to other models and try to look for different evaluation metrics.

Project Increment -2

Model:

1. **BERT:**
The Transformer encoder reads the entire sequence of words at once, in contrast to directional models, which read the text input sequentially (from right to left or left to right). Although it would be more accurate to describe it as non-directional, it is therefore thought of as bidirectional. This trait enables the model to understand a word's context depending on all of its surroundings (left and right of the word).

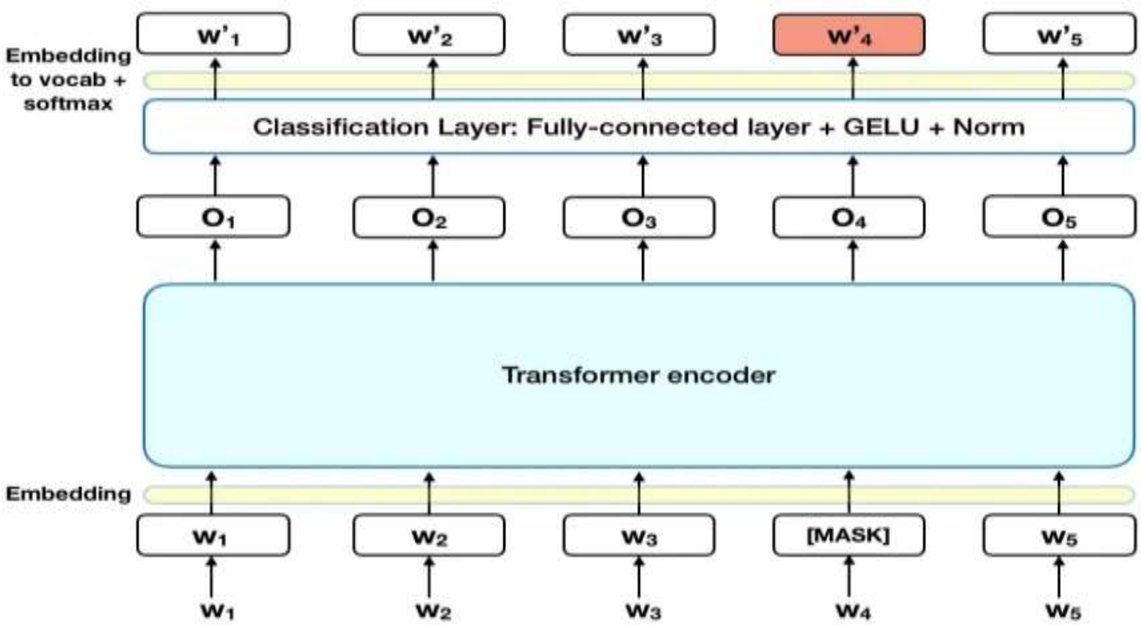

Fig 6: Workflow of BERT

A high-level explanation of the Transformer encoder can be found in the figure above. A series of tokens are used as the input and are first embedded into vectors before being processed by a neural network. The result is a series of vectors of size H, each of which corresponds to a token from the input with the same index.

2. **LSTM:**

LSTM (Long short-term memory) is one of the more intricate subfields in deep learning. LSTM is a difficult concept to comprehend. It deals with algorithms that attempt to emulate the way the human brain functions and find the underlying correlations in the sequential data provided.

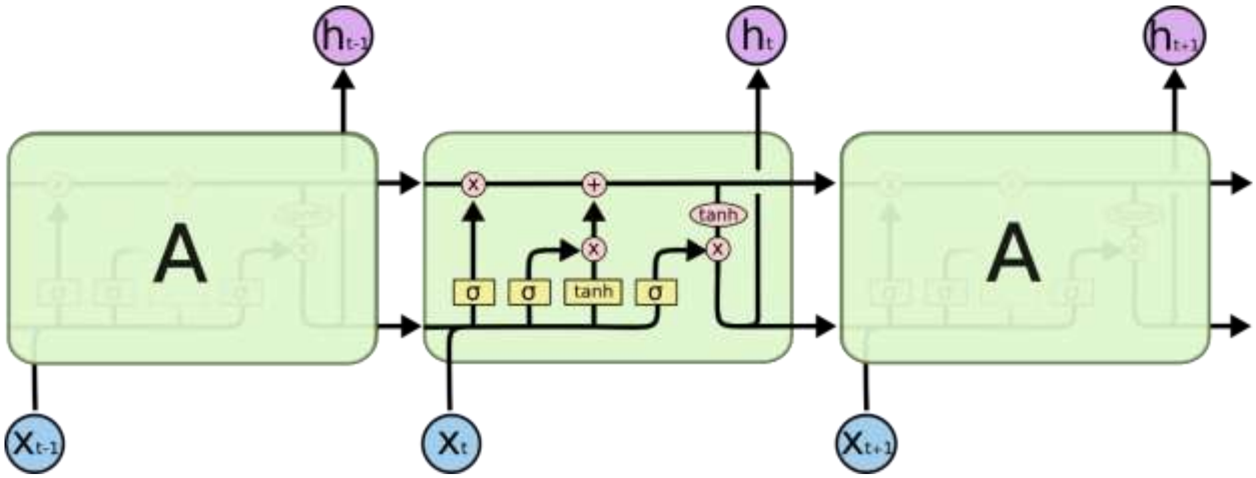

Fig 7: Architecture diagram of LSTM

Typical LSTM networks are made up of various memory units called cells (the rectangles that we see in the figure 7). The cell state and the hidden state are the two states that are being passed to the following cell. The three main mechanisms, referred to as gates, are used to manipulate the memory blocks, which are responsible for memory. Gates in LSTMs control the addition and removal of information from the cell state. These gates may allow information to enter and exit the cell. It has a sigmoid neural network layer and a pointwise multiplication operation that help the mechanism. Between zero and one, the sigmoid layer outputs a number, with zero denoting "nothing should be let through" and one denoting "everything should be let through."

Dataset:

We have taken the segmented dataset from the metadata of CHORD-19 dataset and our dataset contains the data from the article which is published after 2020. We have taken 10000 data from this to train our model as it takes long to train the model. The detailed of this dataset is discussed above in the increment 1. For the training and testing purpose, we only take abstract and title of dataset as there is no use of others column. The sample of dataset is shown below:

	title	abstract
0	Correction.	[This corrects the article on p. 147 in vol. 33.].
11	A bioactive compliant vascular graft modulates macrophage polarization and maintains patency with robust vascular remodeling.	Conventional synthetic vascular grafts are associated with significant failure rates due to their mismatched mechanical properties with the native vessel and poor regenerative potential. Though di...
12	Strategies to increase rural maternal utilization of skilled health personnel for childbirth delivery in low- and middle-income countries: a narrative review.	BACKGROUND Skilled attendance at birth is considered key to accomplishing Sustainable Development Goal (SDG) 3.1 aimed at reducing maternal mortality. Many maternal deaths can be prevented if a wo...
13	A 'Multiomic' Approach of Saliva Metabolomics, Microbiota, and Serum Biomarkers to Assess the Need of Hospitalization in Coronavirus Disease 2019	Background and Aims The SARS-CoV-2 pandemic has overwhelmed the treatment capacity of the health care systems during the highest viral diffusion rate. Patients reaching the emergency department ha...
14	Chapter 8 Infectious Disease Emergencies	COVID-19 has added new relevance to the relationship between the world's public health and infectious diseases and makes timely the topic of this chapter, that is, infectious disease emergencies. ...

Fig 8: Sample of dataset

Analysis of Data:

We find out the missing and null value of our dataset which is shown in figure 9 below. We get 1184 null value in abstract and 1 title. We drop all the null values and duplicates from our dataset and get 8810 datapoint for both abstract and title attributes.

```
[ ]  df.isna().sum()

     title          1
     abstract    1184
     dtype: int64
```

Fig 9: Null Value

After getting the cleaned dataset, we see the most common or repeated word of abstract and title by means of world cloud as shown in figure below respectively. The most bold words represent the most common words in the dataset.

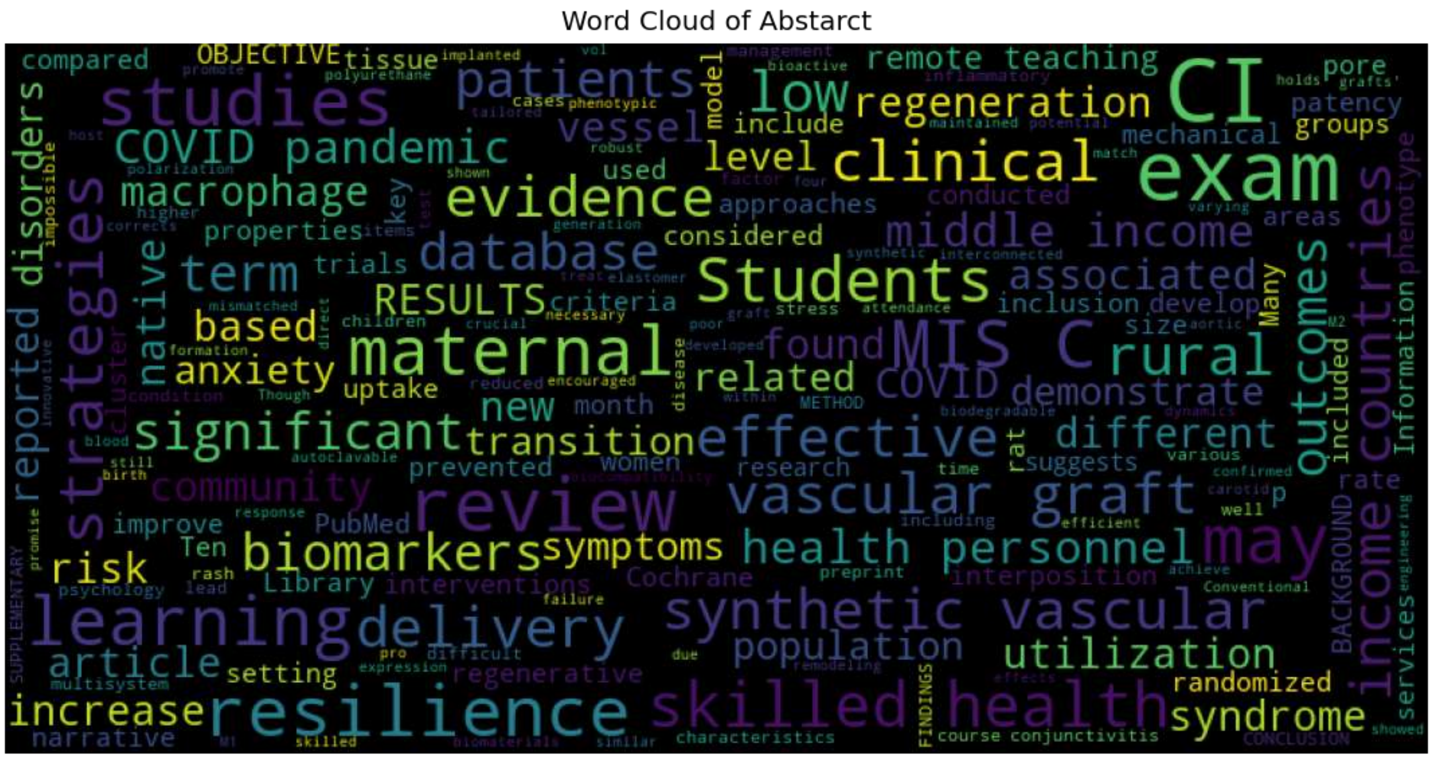

Fig 10: Word Cloud of Abstract

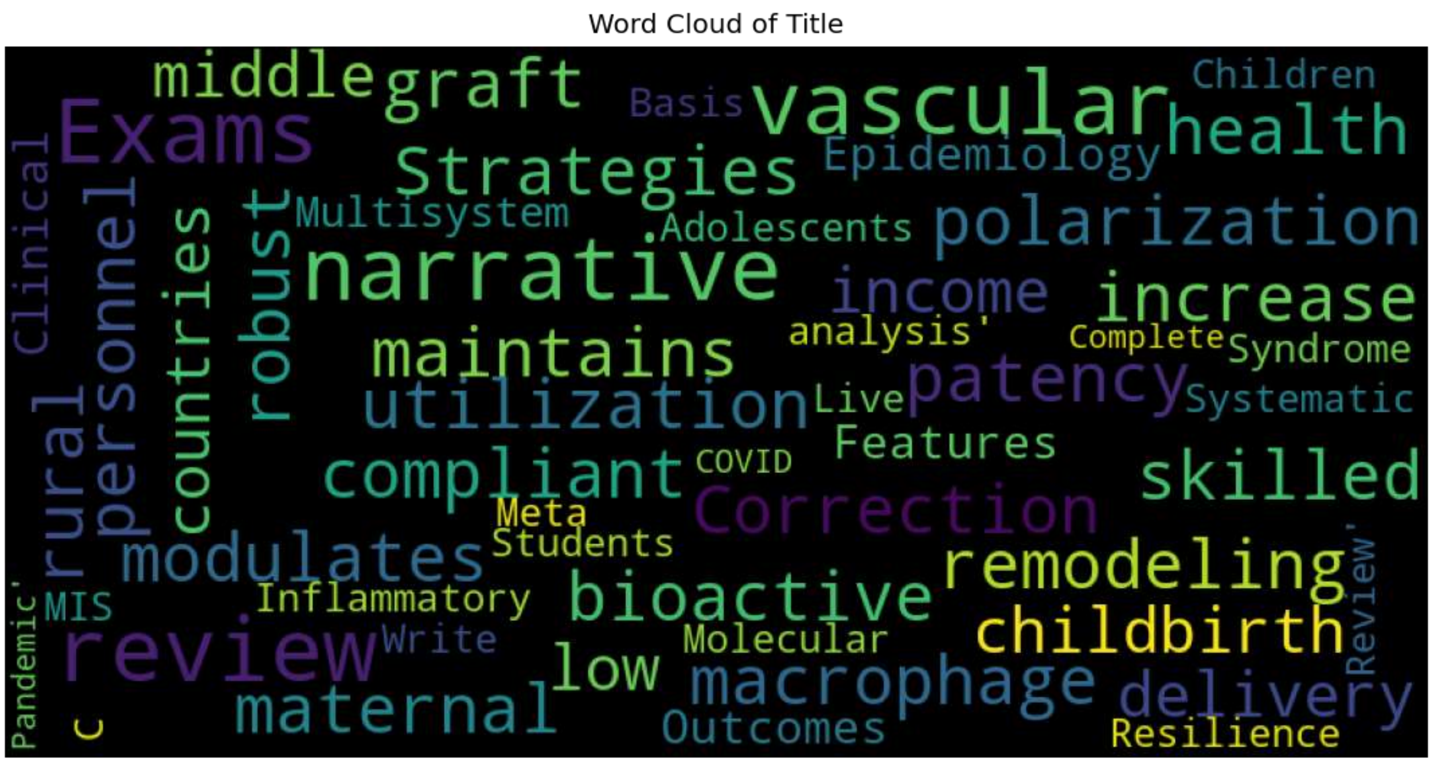

Fig 11: Word Cloud of Title

After the cleaning, we did the preprocessing same as before for BERT model but we add contraction mapping here. word count of abstract and title are shown in figure 12 below. We can clearly see that abstract have more word count than the title. For tokenization, we use the Keras preprocessing library to tokenize the training dataset. After the tokenization, we add extra token 'sostok at starting and "eostok' at end point to distinguish the

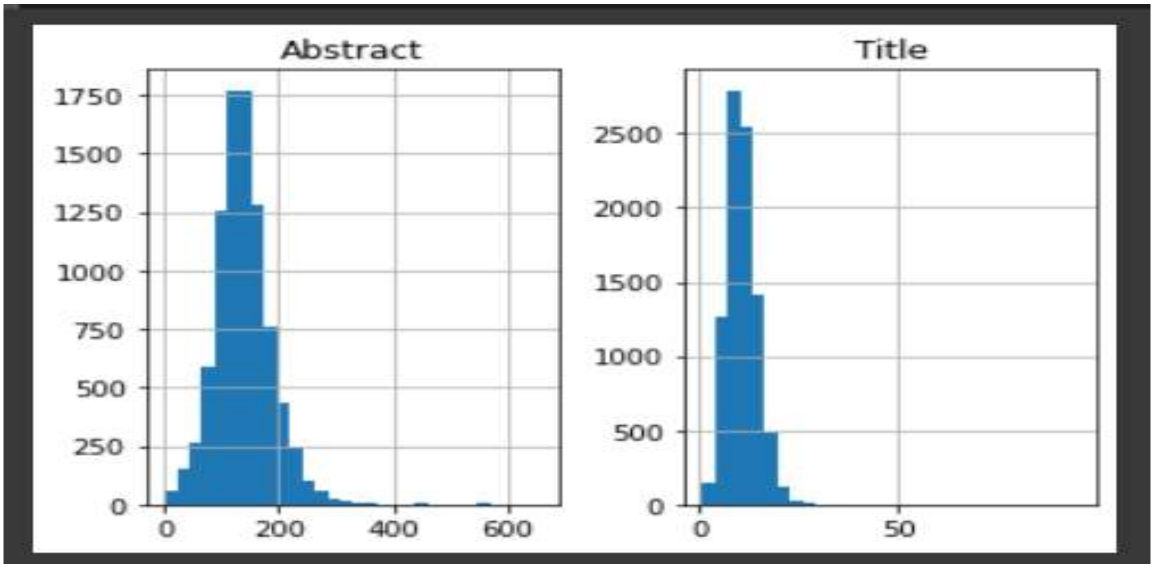

Fig 12: Word Count of dataset

different title of dataset which will be helpful for the validation. We split the dataset in the ratio or 9:1 ratio for training and testing purpose respectively. After the tokenization, we check the percentage and total coverage of rare words in the dataset and get **62.23%** and **3.799** respectively in the title attribute. We get **84.358%** of rare words and **19.54** of total coverage of rare words in the vocabulary of abstract datapoint.

Implementation

Pseudocode for LSTM:

1. Include the current input along with the previous concealed state and internal cell state.
2. Values are calculated into four different gates as shown below:
   - By elementwise multiplying the relevant vector with the appropriate weights for each gate, get the parameterized vectors for the current input and the prior hidden state for each gate.
   - Apply the appropriate activation function to the parameterized vectors for each gate element.
3. By first calculating the element-wise multiplication vector of the input gate and the input modulation gate, then calculating the element-wise multiplication vector of the forget gate and the prior internal cell state, and finally adding the two vectors, you can determine the current internal cell state.
4. Element-wise multiplication with the output gate is used to determine the current concealed state by first determining the element-wise hyperbolic tangent of the internal cell state vector.

We have use the LSTM model which is shown in figure 13. We have taken three LSTM hidden layer and at last of decoder we have added the attention layer to the decoder. We have taken latent dimension of 150 and embedding dimension of 50 and get 2,110,225 total and trainable parameter which are same for both. The first encoder input of LSTM model takes the maximum length of abstract which is set to 100. The embedding layer takes the vocabulary of summary and dimension of embedding with the first encoder input. We use the rmsprop optimizer to increase the learning rate where rmsprop algorithm takes big step in the horizontal way to makes process faster and sparse categorical cross entropy to converts the integer sequence to a one-hot vector. We have also use the early stopping which stop training of the model when our validation loss increases. In our case, our model stop at validation loss of 1.4977 which the least loss of the model.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 100)]        0           []

 embedding (Embedding)          (None, 100, 50)      734350      ['input_1[0][0]']

 lstm (LSTM)                    [(None, 100, 150),   120600      ['embedding[0][0]']
                                 (None, 150),
                                 (None, 150)]

 input_2 (InputLayer)           [(None, None)]       0           []

 lstm_1 (LSTM)                  [(None, 100, 150),   180600      ['lstm[0][0]']
                                 (None, 150),
                                 (None, 150)]

 embedding_1 (Embedding)        (None, None, 50)     103750      ['input_2[0][0]']

 lstm_2 (LSTM)                  [(None, 100, 150),   180600      ['lstm_1[0][0]']
                                 (None, 150),
                                 (None, 150)]

 lstm_3 (LSTM)                  [(None, None, 150),  120600      ['embedding_1[0][0]',
                                 (None, 150),                     'lstm_2[0][1]',
                                 (None, 150)]                     'lstm_2[0][2]']

 attention_layer (AttentionLaye  ((None, None, 150),  45150      ['lstm_2[0][0]',
 r)                              (None, None, 100))               'lstm_3[0][0]']

 concat_layer (Concatenate)     (None, None, 300)    0           ['lstm_3[0][0]',
                                                                  'attention_layer[0][0]']

 time_distributed (TimeDistribu  (None, None, 2075)  624575      ['concat_layer[0][0]']
 ted)

==================================================================================================
Total params: 2,110,225
Trainable params: 2,110,225
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig 13: Our LSTM Model

Results

The graph below shows that our training loss start from 2.3 and validation loss from 1.8362 which is beginning phase of training the model. We set the epochs of 50 and batch size of 128 which means the model will be trained to 50times and batch size indicates that the training data is count as 1 input for 128 datapoints. As we set the early stopping, the training of the model stops at epoch 29 as the model is getting the same validation loss. So, at end of the training the training loss is 1.317 and validation loss of 1.4977. The training

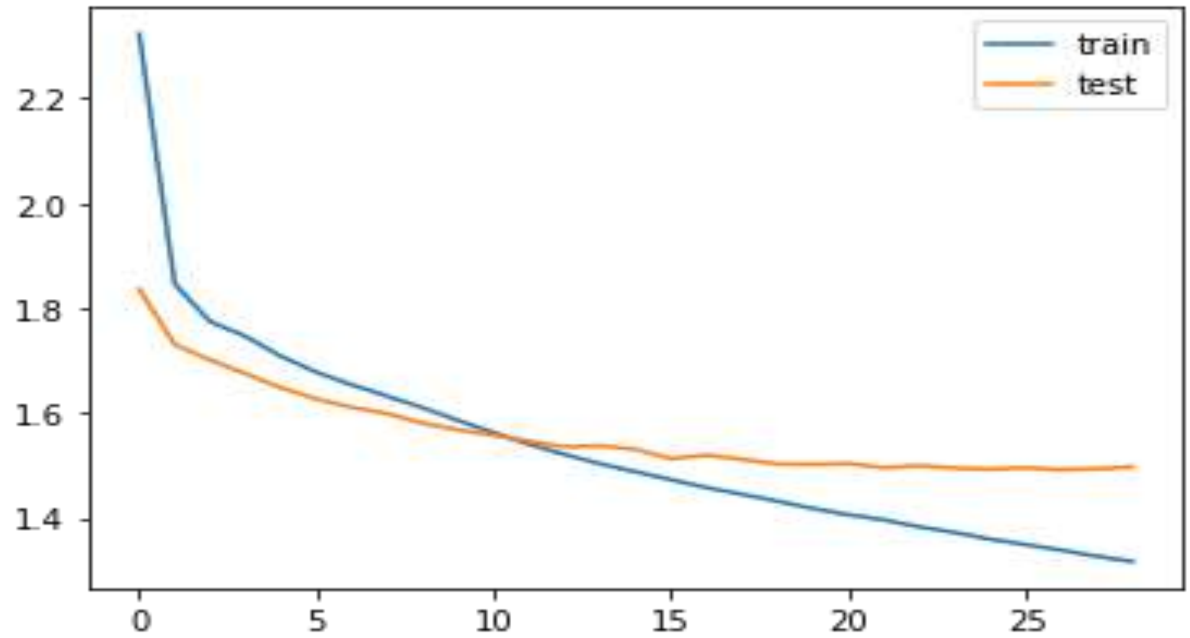

Fig 14: Training loss Vs Validation loss

loss of our model decreases constantly from the beginning to end of the training period but for the validation loss our model decreases till epochs 24 but after that this remain constant and slightly increases and decreases from the epochs 24 to 29. We set the inferencing encoder and decoder to get our predicted abstractive summary of the title. Before predicting the title, the extra token which is added at the starting and ending point are removed and then our summary title is predicted. We have taken the range of 100 data to predict the title summary as shown in figure 15. We get the abstractive summary of title which have same meaning as of original title of our dataset.

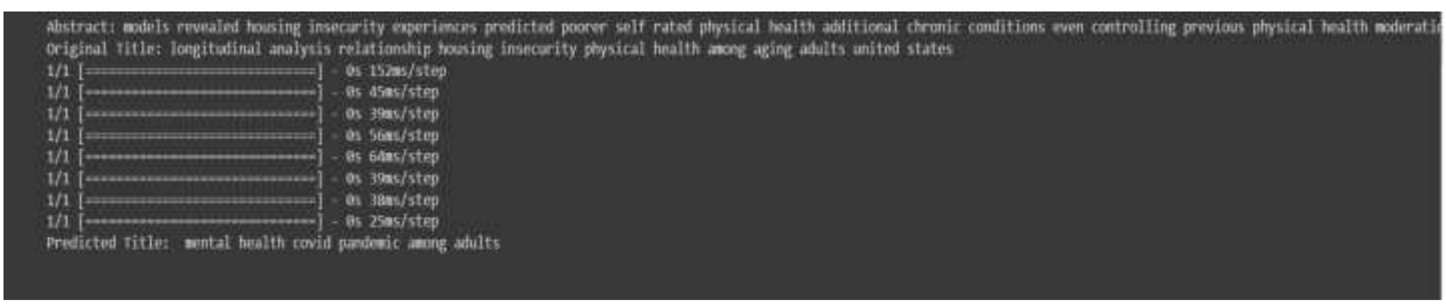

Fig 15: Abstractive Summary of title

For the evaluation of our model, we calculate the rouge score which is shown below in figure 16. The rouge-1 is overlap of unigram between original title and get predicted title and get recall 0.23, precision 0.103 and F1 score 0.13. Rouge-2 reflects the bigram of original and predicted title and for this we get recall 0.09, precision 0.036 and F1 score 0.049. Rouge-L calculate the longest matching sequence between both training and validation data. We get the F1 score high for the rouge-1 which means our model makes better prediction during unigram and worse prediction during bigram which shows in rouge-2. Our model makes more false negative prediction for rouge-1 which is 0.23 and highest false positive prediction for rouge-1 which is 0.103.

```
[60] rouge = Rouge()
     rouge.get_scores(title, predicted,avg=True, ignore_empty=True)

{'rouge-1': {'r': 0.23024999999999995,
  'p': 0.10339693639693637,
  'f': 0.1381351230556726},
 'rouge-2': {'r': 0.09092857142857143,
  'p': 0.036539682539682546,
  'f': 0.04997186876638159},
 'rouge-l': {'r': 0.21791666666666662,
  'p': 0.097043401043401,
  'f': 0.12980358531237018}}
```

Fig 16: Rouge score

At Last, we save the prediction of our model in csv file to utilize for question answering using the Bert model in the future.

Limitations and Conclusion

There were many challenges that we faced are

i. Initially, our plan was to Integrate Question Answering model with Text summarization using BERT and Distil BERT.
ii. We implemented BERT model as part of our Increment -1 and then we tried implementing Distil BERT, but the results were almost the same. Hence, we dropped it and implemented LSTM to see how well the model performs.
iii. But the thing with LSTM is it cannot be used for Question Answering; hence we did Text summarization using LSTM.

Implementation Status Report

Work Completed

We have created a Text summarizing using the LSTM model.

- Harsha – Dataset, data pre-processing
- Roshan – Model implementation and Results

We both worked equally on report writing and presentation slide.

References:

[1] Mahsa Afsharizadeh, KOMLEH HOSSEIN EBRAHIMPOUR, and Ayoub Bagheri. Automatic text summarization of covid-19 research articles using recurrent neural networks and coreference resolution. 2020.

[2] Abdullah Javaid Chaudhry, Shehryar Hanif, and Muhammad Ali. An exploration in extractive text summarization and sentence vectors with specific reference to covid-19 medicinal articles.

[3] Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wen-peng Yin, Dragomir Radev, and Richard Socher. Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. NPJ digital medicine, 4(1):1–9, 2021.

[4] Nur Hayatin, Kharisma Muzaki Ghufron, and Galih Wasis Wicaksono. Summarization of covid-19 news documents deep learning-based using transformer architecture. TELKOMNIKA (Telecommunication Computing Electronics and Control), 19(3):754–761, 2021.

[5] Bhrugesh Joshi, Vishvajit Bakarola, Parth Shah, and Ramar Krishnamurthy. deepmine-natural language processing based automatic literature mining and research summarization for early-stage comprehension in pandemic situations specifically for covid-19. bioRxiv, 2020.

[6] Ammar Kadhim. An evaluation of preprocessing techniques for text classification. International Journal of Computer Science and Information Security,, 16:22–32, 06 2018.

[7] Virapat Kieuvongngam, Bowen Tan, and Yiming Niu. Automatic text summarization of covid-19 medical research articles using bert and gpt-2. arXiv preprint arXiv:2006.01997, 2020.

[8] Chatchawarn Limploypipat and Nuttanart Facundes. Abstractive text summarization for covid-19 news with data augmentation. In 2022 International Conference on Digital Government Technology and Innovation (DGTi-CON), pages 56–59, 2022.

[9] Milad Moradi, Georg Dorffner, and Matthias Samwald. Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. Computer methods and programs in biomedicine, 184:105117, 2020.

[10] Afsaneh Rezaei, Sina Dami, and Parisa Daneshjoo. Multi-document extractive text summarization via deep learning approach. In 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), pages 680–685, 2019.

[11] Deepika S, Lakshmi Krishna N, and Shridevi S. Extractive text summarization for covid-19 medical records. In 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), pages 1–5, 2021.

[12] Shengli Song, Haitao Huang, and Tongxiao Ruan. Abstractive text summarization using lstm-cnn based deep learning. Multimedia Tools and Applications, 78(1):857–875, 2019.

[13] Dan Su, Yan Xu, Tiezheng Yu, Farhad Bin Siddique, Elham J Barezi, and Pascale Fung. Caire-covid: A question answering and query-focused multi-document summarization system for covid-19 scholarly information management. arXiv preprint arXiv:2005.03975, 2020.

[14] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Michael Kinney, Ziyang Liu, William Cooper Merrill, Paul Mooney, Dewey A. Murdick, Devvret

Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D Wade, Kuansan Wang, Christopher Wilhelm, Boya Xie, Douglas A. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset. ArXiv, 2020.

[15] Awane Widad, Ben Lahmar El Habib, and El Falaki Ayoub. Bert for question answering applied on covid-19. Procedia Computer Science, 198:379–384, 2022. 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks/11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare.

[16]http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[17] https://intellipaat.com/blog/what-is-lstm/

Video Link: https://youtu.be/tTu6W6gOrf0

GitHub Link - https://github.com/ErRsah/NLP