Group 20

CORD-19 based Question Answer System

Assignment 3: Natural Language Processing (Tri1 2023)

Project By:
Harpreet Kaur Hans | A1873328
Priyank Dave | A1843068
Chinmay Dharmik | A1855351

Executive Summary:

The COVID-19 pandemic has caused an unprecedented global crisis, affecting various aspects of human life. With the rapid spread of the virus, access to accurate and reliable information has become crucial. The internet has become the primary source of information, but with the massive amount of information available, it can be difficult to separate facts from fiction. In this regard, natural language processing (NLP) and question answering (QA) systems have emerged as promising solutions to help people access accurate information quickly and efficiently.

In this project, our team aimed to develop a COVID-19 QA system using state-of-the-art NLP models and techniques. We utilized parallel processing to optimize the performance of the system, ensuring fast and efficient responses to user queries. The system was trained on a COVID-19 dataset and was able to accurately retrieve answers to COVID-related questions. Additionally, we implemented a method to convert the system's output to natural language sentences, improving the user experience. Through this project, we gained valuable insights into the potential of NLP and QA systems in providing timely and accurate information to people during times of crisis.

Introduction:

The project aims to build a Question Answering system using the **roBERTa** and **spaCy** to retrieve relevant information from the vast **CORD-19** dataset of over 200,000 scientific articles related to COVID-19. **RoBERTa** is a powerful pre-trained transformer-based model that can be fine-tuned for various NLP tasks such as Question Answering. **SpaCy** is a natural language processing library that provides advanced entity recognition and parallel processing capabilities, which can significantly improve the speed and efficiency of information retrieval and Question Answering systems. Here, we are using parallel processing to speed up the inference time, allowing the model to retrieve entities more quickly and efficiently. It works by dividing a large task into smaller sub-tasks that can be processed simultaneously by multiple processors or threads. The project demonstrates the potential of natural language processing and machine learning in addressing the challenges of information retrieval from scientific literature.

Analysis of method suitability, approach and research:

To begin this assignment, we divided the task into various sections, each responsible for various functions. In contrast to Assignment 2, where we performed a paper-level search, here we conducted a paragraph-level search to improve precision. We imported all data so that a single paper corresponds to a list of paragraphs. We then performed basic pre-processing to make it easier for the model to extract entities.

One challenge that arose was how to find a paragraph that potentially had the answer to the query. We explored a few options, such as creating **paragraph-level embeddings** and using cosine similarity. Trail 1 involved using a pre trained **Sentence Transformer** model [1], which was discarded due to high computation time and inaccuracy. Trail 2 used **Doc2Vec** Library from **Gensim** [2], but it also had high computation time and was inaccurate. Idea 2 involved selecting paragraphs with common entities to the entities in the question. We used **SciSpaCy** [3], a version of **SpaCy**, to extract named entities. We implemented entity indexing to obtain a dictionary of entities and the indices of the paragraphs that had that particular entity. We used entity matching utility to get the indices of all paragraphs that have a subset of all the entities as the question and rank those paragraphs according to their similarity to the query using **SpaCy** similarity. This approach worked well, involving initial computation but with query handling time in the ballpark of a few minutes (<5), with a few sample paragraphs similar to the queried answer.

The next challenge was to implement a Question Answer System. We referred to a **pretrained transformer** capable of handling COVID-based questions [4]. We found that the **RoBERTa model** [5], fine-tuned on the **Squad Style annotated CORD19 dataset** [6], was well-suited for the task and accurately retrieved answers from queries.

The following challenge was to convert the output of the **Question Answer System** to a natural language sentence. We attempted to use **SpaCy** to extract entities and perform **POS tagging** to extract the semantic structure of the query and then create a template that would suffice for the grammatical requirements of the majority of the

answer requirements. However, due to the nature of the questions being more related to **biomedical** topics with entities that **SpaCy** or **SciSpaCy** could not identify accurately, this approach was not successful. Idea 2 involved using a **summarizer model**, where we masked the **wh-question words** to force the summarizer to rewrite the entire text and generate an answer statement. We compared Facebook's **bart-large-cnn model** [7] with **T5-Base_GNAD** model [8] and found that the **bart-large-cnn model** was more natural and faster.

The final challenge was to convert the answer sentence output to voice. We used the **pyttsx3** [9] Python library, which is a utility that converts text to voice in an instance.

In conclusion, we adopted an approach involving paragraph-level search and entity matching utility to improve the precision of the Question Answer System. We used a **pretrained RoBERTa** model and Facebook's **bart-large-cnn** model to accurately retrieve answers and convert the output to a natural language sentence. The **pyttsx3** library was used to convert the answer sentence output to voice.

Description of the dataset:

The dataset used in this assignment is the **COVID-19 Open Research Dataset** otherwise known as the **CORD-19** dataset. It is a "resource of over 1,000,000 scholarly articles, including over 400,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses" [10].

Description of implementation:

Reading The Dataset

The **CORD-19** dataset being a huge dataset with information spread across various articles and in different formats like PDF and PMC, makes it difficult to efficiently extract the required data. In order to address this issue and to make preprocessing easier, we selected only the relevant columns like the title of the articles, their IDs which would help us to uniquely identify the articles, the time the article was published, authors and DOI. Further, we limited our articles to being in PDF formats as PDFS are much more readable, more widely recognised, compatible, and easier to handle than PMC files. Any duplicate articles or **NaN** values in the dataset were dropped.

Preprocessing And Filtering Data

The preprocessing was performed on two levels namely article level and text level. On the article level, we did not consider articles not specific to COVID-19 or which were published before 2019. On exploring the dataset further, it turned out there were a reasonable number of articles which were not in the English language which for the purpose of this assignment, were eliminated. On the text level, which refers to the text in the articles, we removed emails, links, citations, reference numbers and

unnecessary spaces. It must be noted that quite a lot of preprocessing on this level was aimed at cleaning the text and eliminating the elements that appear frequently in research articles. Further, all stop words (the list of stop words was further extended to include words which are common in research articles, for example, “DOI”, “peer”, “reviewed”, “figure”, etc.) were eliminated and all the text was converted into lowercase.

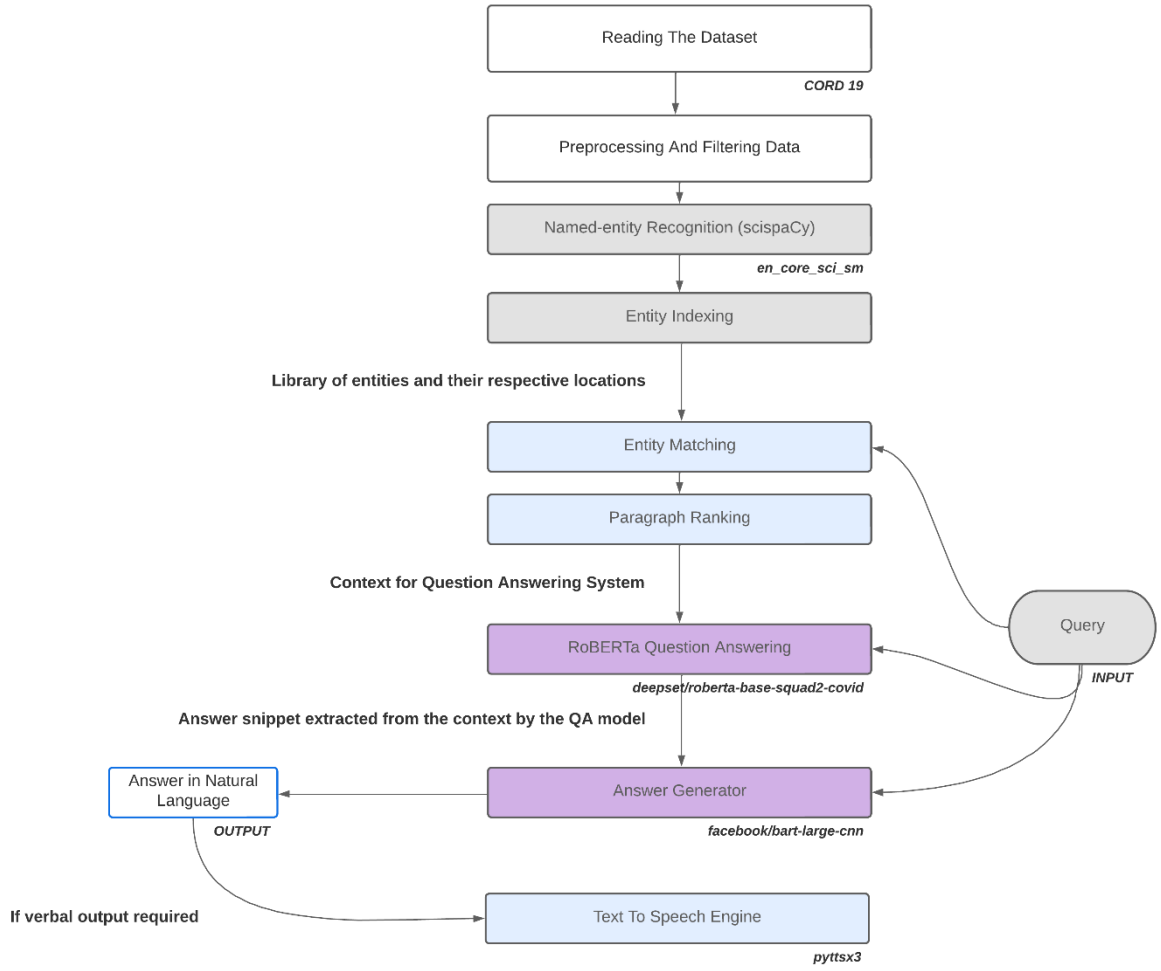


Figure 1 The above diagram summarizes the basic architecture of the system we implemented.

Named-entity Recognition

We extracted the entities from the preprocessed text by performing Named Entity Recognition. This was achieved by using the **scispaCy** library in Python.

Entity Indexing

Entity indexing is a technique used in the Question Answer System to create a dictionary of entities and a list of indices of paragraphs containing those entities. For our purposes, this was achieved by using the **scispaCy** library, a version of **spaCy** specifically designed for working with scientific and biomedical text. Named-entities were extracted from the text using **scispaCy**, and entity indexing was used to map each named-entity to the indices of the paragraphs in which it appeared. By doing so, the

team was able to identify relevant paragraphs more quickly and accurately, improving the efficiency and accuracy of the Question Answer System.

Entity Matching and Paragraph Ranking

Based on the indexes obtained in the previous step, we found the indexes of the paragraphs matching the entities of the query. This was done by using spaCy's similarity function, which in turn uses **cosine similarity**.

We then performed Paragraph ranking to obtain top 10 paragraphs and concatenated them. This was what formed the context for our **roBERTa** model.

RoBERTa Question Answering System

Our next challenge was to implement a Question Answer System that could accurately identify COVID-related terms and retrieve answers from the query. After researching several pre-trained transformer models capable of handling COVID-based questions, we found that the RoBERTa model fine-tuned on the Squad Style annotated Squad dataset was well-suited for the task and accurately retrieved answers from queries. With this model, we were able to generate answers to our questions with high accuracy and efficiency.

We used the **roBERTa** model which is a pre trained Question Answering model, adapted from **Hugging Face**. Given the question (or query) and the context, our model outputs the answer as a text snippet.

Answer Generation

In the next step, we had to tackle the challenge of converting the output generated by the Question Answer System to a natural language sentence. We tried using **spaCy** for entity extraction and **POS tagging**, but the nature of our queries, which were related to biomedical terms, made it difficult for **spaCy** to accurately identify the entities and perform POS tagging. As a result, we had to look for an alternative solution. We decided to use a summarizer model, where we masked the **Wh question words** and forced the **summarizer** to rewrite the entire text and generate an answer statement. We experimented with different models such as Facebook's **bart-large-cnn** and **T5-Base_GNAD**, and found that the **bart-large-cnn** model was more natural and faster than **T5-base_GNAD**.

Text to Speech Engine

We simply used the **pyttsx3** library to convert text into speech. The speech rate was set to **183** and the volume to **1** . This feature can be activated or deactivated by a YES or NO which is taken as an input from the user.

Results:

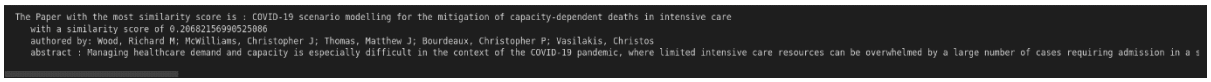
The results for this assignment were generated on a set of questions from the user. At each user prompt, the user was asked for the question and whether they wanted the answer to be dictated. If the user selected **y** which stood for “yes”, the answer to the query was first dictated and then printed. The output for this assignment consisted of answers in natural language both in the form of text as well as speech. The following are the results obtained after running the code for this assignment:

```
Searching for answer for the question: When was covid-19 outbreak declared as a global pandemic? in the list of paragraphs...
100%|██████████| 3/3 [00:00<00:00, 10.78it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: March 11
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 29. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=14)
Question: When was covid-19 outbreak declared as a global pandemic?
Answer: Was covid-19 outbreak declared as a global pandemic? [SEP] March 11.



Searching for answer for the question: What health measures should be taken for covid-19? in the list of paragraphs...
100%|██████████| 2/2 [00:00<00:00, 210.70it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: ."
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 26. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=13)
Question: What health measures should be taken for covid-19?
Answer: "What health measures should be taken for covid-19? [SEP] ."



Searching for answer for the question: How to control the spread of coronavirus? in the list of paragraphs...
100%|██████████| 2/2 [00:00<00:00,  7.60it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: total lockdown is in place in India from 24 th March 2020 for 21 days
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 38. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=19)
Question: How to control the spread of coronavirus?
Answer: Total lockdown is in place in India from 24 th March 2020 for 21 days. [CLS] to control the spread of coronavirus?



Searching for answer for the question: Where did coronavirus start? in the list of paragraphs...
100%|██████████| 1/1 [00:00<00:00,  5.54it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: Wuhan, Hubei Province, China
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 30. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=15)
Question: Where did coronavirus start?
Answer: Did coronavirus start in Wuhan, China?
```

```
Searching for answer for the question: What are the vaccines approved against covid-19? in the list of paragraphs...
100%|██████████| 2/2 [00:00<00:00, 56.59it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: 2021
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 25. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=12)
Question: What are the vaccines approved against covid-19?
Answer: Vaccines approved against covid-19 will be available in 2021.
```

```
Searching for answer for the question: How does covid-19 spread? in the list of paragraphs...
100%|| 2/2 [00:00<00:00, 33.74it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: tracing contacts within 3 days of cases being confirmed
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 30. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=15)
Question: How does covid-19 spread?
Answer: Covid-19 can spread within 3 days of cases being confirmed, according to the CDC.



Searching for answer for the question: What are the symptoms of coronavirus? in the list of paragraphs...
100%|| 2/2 [00:00<00:00,  9.44it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: cough, fever, and breathlessness
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 29. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=14)
Question: What are the symptoms of coronavirus?
Answer: What are the symptoms of coronavirus? cough, fever, and breathlessness.



Searching for answer for the question: Which country became the epicenter of the global pandemic in 2021? in the list of paragraphs...
100%|| 2/2 [00:00<00:00, 53.40it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: Saudi Arabia
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 29. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=14)
Question: Which country became the epicenter of the global pandemic in 2021?
Answer: Saudi Arabia could become the epicenter of the global pandemic in 2021.



Searching for answer for the question: What could be the origin of the coronavirus? in the list of paragraphs...
100%|| 2/2 [00:00<00:00, 73.59it/s]
Searching the answer in the top matching paragraphs using Roberta...
Answer Snippit: China
Generating answer using facebook/bart-large-cnn pretrained model...
Your max_length is set to 40, but you input_length is only 25. You might consider decreasing max_length manually, e.g. summarizer('...', max_length=12)
Question: What could be the origin of the coronavirus?
Answer: China could be the origin of the coronavirus?
```

On the other hand, the results from the Assignment 2 only provided the title of the article, its paper ID, authors and abstract. The following is a screenshot from the code results of Assignment 2:



This assignment clearly outperforms Assignment 2 both in terms of runtime and final output. The runtime for this assignment has significantly reduced due to usage of parallel processing. Also, from the perspective of a Question Answering system, this assignment provides better results as it answers the query of the user in natural language as opposed to text snippets or just the article titles.

Conclusion:

a) What did we learn from the assignment?

As a team, we have learned a great deal from working on this project. One of the most significant insights we gained was the importance of parallel processing in improving system performance. By implementing entity indexing and using multiple CPU cores, we were able to significantly speed up our query handling time, which made the system much more efficient and user-friendly.

Another important lesson we learned was the value of leveraging pre-trained models and libraries to optimize system performance. By using tools like `**RoBERTa**` and `**BART**`, we were able to improve our question-answering and natural language generation capabilities without having to start from scratch. This approach allowed us to focus our efforts on optimizing and integrating these models into our system, which ultimately saved us a lot of time and improved the accuracy of our results.

Overall, working on this project was a valuable learning experience for our team, and we believe that the insights and skills we gained will be useful in future projects as well.

b) Limitations of this assignment:

1. Uncertainty about the source: In some circumstances, identifying a specific source or providing citations that supports the response may be impossible. This could be due to a lack of knowledge, contradictory or incomplete sources, or other issues.

2. Legal or ethical limits: Depending on the project's setting, there may be legal or ethical constraints that prevent or limit the use of specific sources or techniques.

3. Inadequate training data: Because annotated data is utilised to train an NLP model, a lack of annotated data can limit the quantity of data available for training. As a result, the model may be less accurate or resilient since it has not been exposed to a diverse enough set of cases to learn from.

4. Biased training data: Annotated data might be biased if it does not represent the entire set of cases that the model may be expected to handle. For example, if the annotated data only contains questions and answers about one topic or domain, the model may be less effective in answering questions about other topics or domains.

5. Because the system is based on a literature-based dataset of `**CORD-19**`, the model's answerability on unstructured datasets remains an area of research that is not covered in the current study. Also, while we attempted to set a limit for the scores of answers, the transmission of scores varied greatly among different requests. The suggested model simply uses the generated embedding to capture context-sensitive properties. Our system has been pre-trained on a limited domain of COVID-19 material, all of which is in English. There is also the possibility of integrating the system with various trusted

c) Main Challenges:

- To begin with, it was difficult to process the `**CORD-19**` dataset's large amount of unstructured data, which included research papers, journal articles, and pre prints.

The sheer bulk of the dataset made it challenging to effectively extract the required information, which could have increased processing time and resource usage. .

- In supervised learning tasks, ground truth is a collection of information that has been explicitly labeled with the proper responses or labels. It is frequently used to train and assess artificial intelligence models. The `**CORD-19**` dataset, however, lacked a predetermined ground truth that could have been used for QA activities.

Future Implementation:

- The project can be connected to current search engines or chatbots, allowing people to access a large amount of scholarly information on COVID-19 in a more natural and straightforward manner [11].

- Using APIs, it is possible to retrieve crucial data for the Question Answering model and get appropriate material from `**CORD-19**`. Additionally, the model can be integrated with already-used platforms like chatbots and search engines. Users can now browse the scholarly literature on COVID-19 in a way that feels more natural and effective thanks to this. The development of a strong and effective Question Answering model for CORD-19 depends heavily on APIs [11].

- Spark NLP can be used in a `**PySpark**` application to clean up the `**CORD-19**` information set, train and fine-tune the Question Answering model using BERT or other pre-trained transformer models. The parallel processing capabilities of `**PySpark**` can reduce processing time and improve efficiency. It is a robust tool for creating an outstanding `**CORD-19**` Question Answering model [12].

References:

[1] Sentence Transformers, "All-MiniLM-L6-v2", https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, accessed on April 19, 2023.

[2] Doc2Vec Library, Gensim, https://radimrehurek.com/gensim/models/doc2vec.html, accessed on April 19, 2023.

[3] SciSpacy, https://allenai.github.io/scispacy/, accessed on April 19, 2023.

[4] RoBERTa model, Deepset, https://huggingface.co/deepset/roberta-base-squad2-covid, accessed on April 19, 2023.

[5] Squad Style annotated Squad dataset, COVID-QA, https://github.com/deepset-ai/COVID-QA/blob/master/data/question-answering/200423_covidQA.json, accessed on April 19, 2023.

[6] Facebook's bart-large-cnn model, Hugging Face, https://huggingface.co/facebook/bart-large-cnn, accessed on April 19, 2023.

[7] T5-Base_GNAD model, Hugging Face, https://huggingface.co/Einmalumdiewelt/T5-Base_GNAD, accessed on April 19, 2023.

[8] Spacy, https://spacy.io/models, accessed on April 19, 2023

[9] pyttsx3 , https://pypi.org/project/pyttsx3/ , accessed on April 19, 2023.

[10] https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge

[11] https://arxiv.org/abs/1910.10683 (Exploring the Limits of Transfer Learning)

[12] https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/