Question-Answering (QA) Tool for COVID-19

BT 5153
Group Project

24/04/2022

Xhoni Shollaj A0231930N | Philippine Gallot A0231973B | Remy Masbatin A0231979N | Sahil Sharma A0232063U | Isaac Sadikin A0163058E

SECTION 1: Problem Statement

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Although infected people will have mild flu-like symptoms, COVID-19 can expose older and more vulnerable populations to greater risks. As of April 2022, globally, over 400 million people were infected and over 6 million people died of the disease.

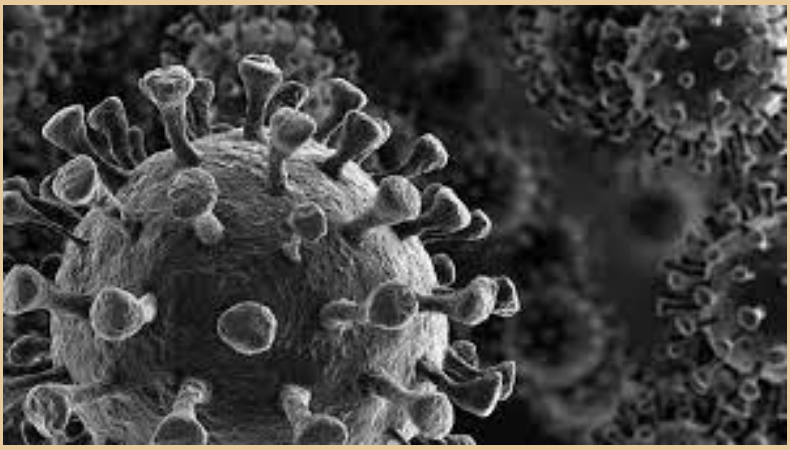
Given the new nature of this coronavirus, people have been asking many questions surrounding COVID-19. Searching for answers online is the most natural next step, yet sources may not be reliable and contradict each other.

On the one hand, for governments, making sure people quickly get access to the latest, most reliable and most relevant information to answer their interrogations is crucial to the management of the crisis.

For example, in Singapore, people infected with COVID-19 used to have to be quarantined in dedicated facilities, even though they were asymptomatic. Now, most infected people can stay at home to recover for a few days. For the Singaporean government, making sure people know this is crucial to prevent hospitals from being overburdened.

On the other hand, for the general public, searching accurate COVID-19-related information can be time-consuming and cumbersome with various online platforms that offer multiple information. There may not always be a research tool and on some websites, it may be necessary to go through a lot of questions before finding an appropriate answer. The risk is that people would directly type the question in their search engine, as the first answers may not be up-to-date or accurate.

Therefore, there is a need to create a Question-and-Answer (QA) system whereby the accurate COVID-19 information can be made swiftly and easily available to many people, thereby reducing the amount of time in searching in open source. Delivering accurate responses could in turn help fight misinformation and allow the general public to act responsibly and knowingly.



SECTION 2: Data Source, Data Preprocessing, and EDA

The COVID-19 Open Research Dataset (CORD-19) is a growing resource of scientific papers on COVID-19 and related historical coronavirus research.

We used the historical releases of CORD-19 from Allen Institute, which is prepared by a coalition of leading research groups using over 500,000 scholarly articles. While the dataset is updated on weekly basis, we used the version that was released on November 15, 2021 (file: cord-19_2021-11-15).

Every version release of the corpus is tagged with a date stamp and has the following files:

- **changelog: A text file summarizing changes between this and the previous version.**
- **cord_19_embeddings.tar.gz: A collection of precomputed SPECTER document embeddings for each CORD-19 paper.**
- **document_parses.tar.gz: A collection of JSON files that contain full text parses of a subset of CORD-19 papers**
- **metadata.csv: Metadata for all CORD-19 papers.**

We used the metadata of the articles which include features such as:

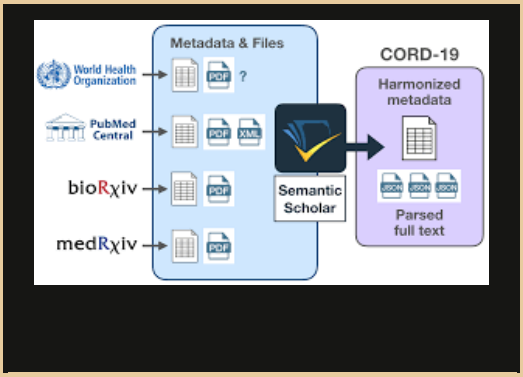
- **'sha' (hash value of encryption algorithm paper),**
- **'title' (Paper Title)**
- **'publish_time' (Date of Publishing of Paper)**
- **'authors' (Name of Paper Authors)**
- **'Abstract' (Paper Abstract')**
- **'url' (the site from which paper is fetched),**

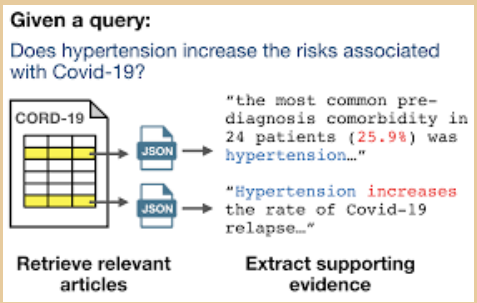
etc and used pickle library to serialize the structure in a dataframe. A brief overview of the metadata features can be found in Appendix 1.

We used the pickle library to serialize the PDF documents and the JSON files from the cord_19_embeddings.tar.gz document, which contained a collection of precomputed SPECTER

(Document-level Representation Learning using Citation-informed Transformers)

document embeddings for each CORD-19 paper for which the features included 'question' (question asked by user), 'answer' (answer for the query), 'score' (assigned score to the question-answer), 'probability' (confidence level for the answer), 'context' (text containing the context around the question), 'offset_start' (starting index of the answer), 'offset_end' (end index of the answer) and 'document_id' (ID od document containing the answer).





"High level of how the data set is created and how the query would be processed against the research papers document store we would create"

Further, the serialized JSON of PDF and PMC files were merged to include ‘body_text’ (entire text of the abovementioned files) with the main metadata dataframe.

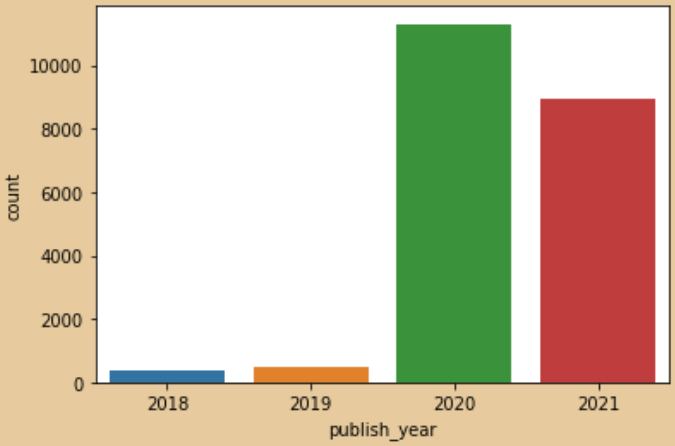
Basic sanity of the data was carried out by removing stopwords, lowercasing text, null values and we then analyzed the ‘abstract’ and ‘body_text’ features and retained the more accurate and complete versions in the main dataframes. We included 25000 samples from the data .

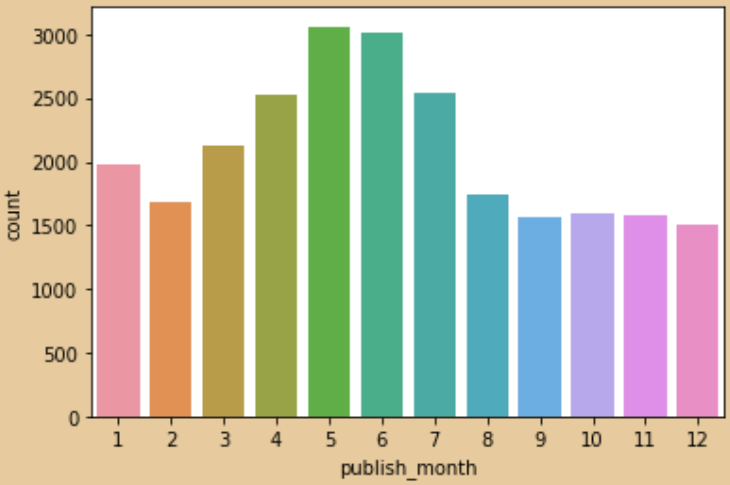
The knowledge and relevant information from CORD-19 dataset were retrieved to illustrate extracting valuable information from scientific papers. The results suggest that papers include a wide variety of the many entity types that are engineered, and that assertion status detection is a useful filter on these entities.

First, let’s explore the date of publication of documents in our dataset.

On the chart below, it appears that the papers were published between 2018 and 2021, with most papers being published in 2020 and 2021. It is surprising that some papers were published in 2018, but these papers must be centered around precedent coronaviruses.

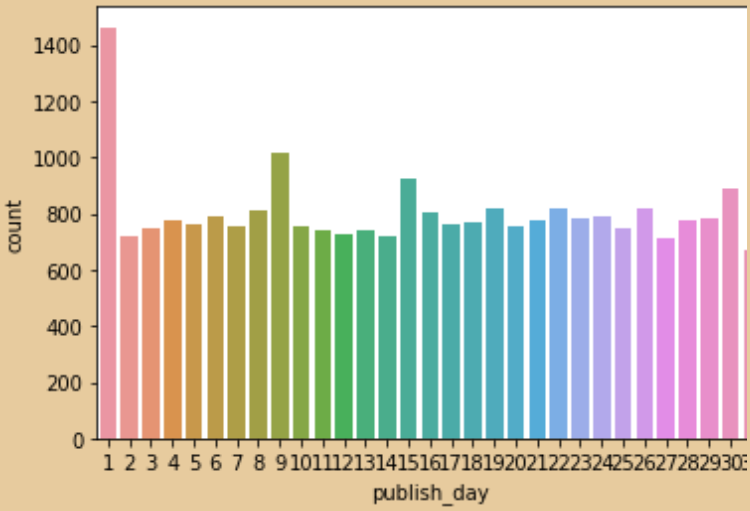
Interestingly, the chart on the right shows that most papers were published around the spring and summer, with a peak in May and June. A smaller number of papers were published during the end of the year (September to December).





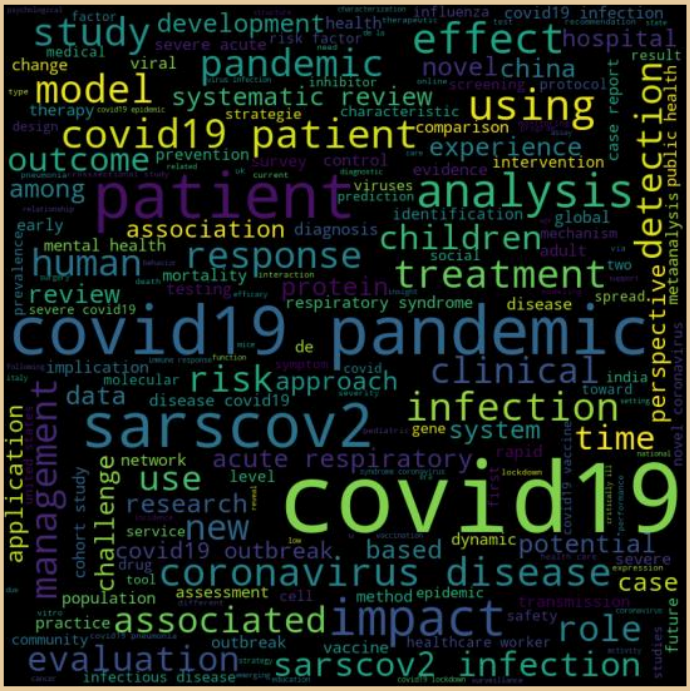
As for the day of publication, the distribution of days is almost uniform with a peak on the first day of the month.

It can either be a practice to publish a paper on the first day of the month, or it is also possible that when the day of the month is unknown, it is set to the first day of the month by default.



What’s more, the most frequent phrases from the selected entity types can be found in the wordcloud below.

Unsurprisingly, the word ‘covid19’ appears twice, and other words among the most important words include ‘pandemic’, ‘sarscov2’, and ‘patient’.



Finally, the two word clouds below show the content of the two main topics extracted from the data. The method used to extract these two topics is Latent Dirichlet Allocation (LDA). LDA is a popular topic modeling technique to extract topics from a text corpus. Latent means hidden or concealed, conveying the idea that the topics extracted from the data are hidden topics.

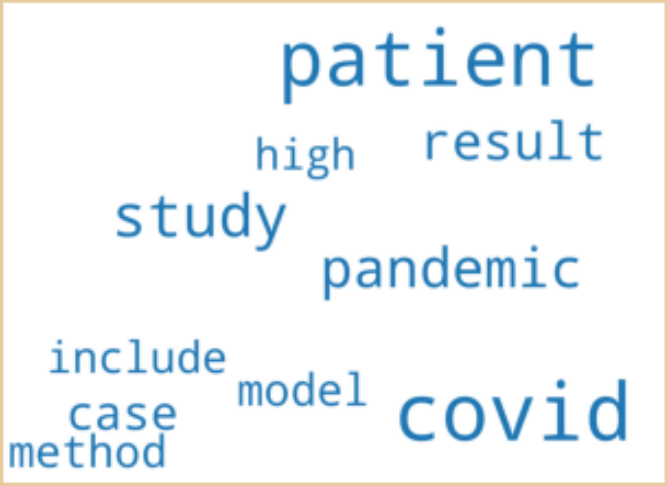
LDA is built on the idea that each document can be described with a distribution of topics and each topic can be described with a distribution of words.

These topics are precisely the latent or hidden layer since they are not observed.

Notably, the 'Topic 0' cites words in the lexical fields of epidemiology and medical studies such as 'study', 'method' and 'model'.

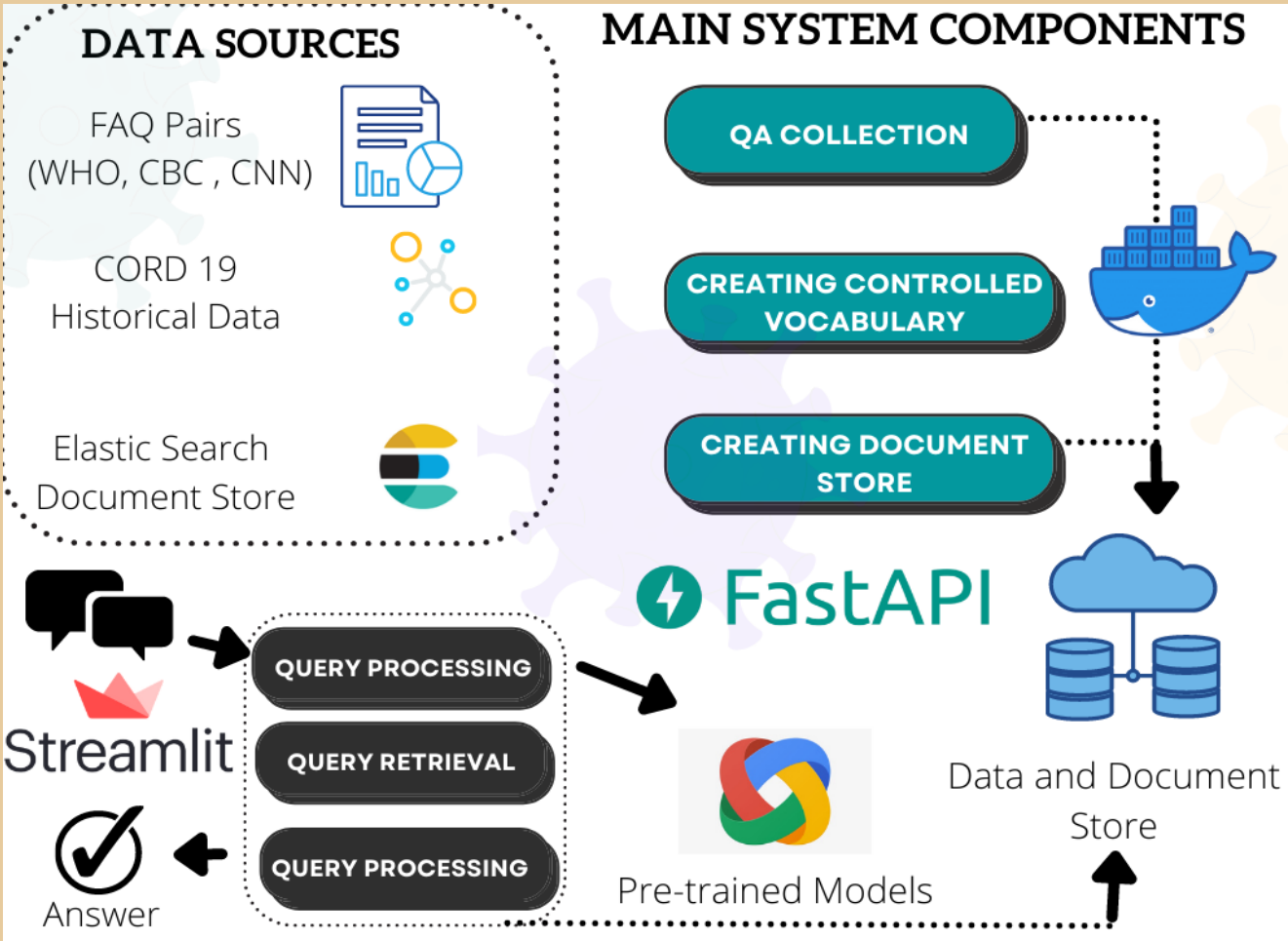
On the other hand, the 'Topic 1' gathers words around the topic of illnesses such as 'virus', 'infection', 'disease', and biological words including 'viral', 'protein', 'cell'

Topic 0



Topic 1

SECTION 3: Methodology



To build our Extractive QA engine, we used Docker container images to host the application, used Haystack library for developing an end-to-end question answering systems along with elasticsearch retriever model to fetch the data received, set up the backend of the tool through FastAPI. and built the UI for user to see question answers in Streamlit.

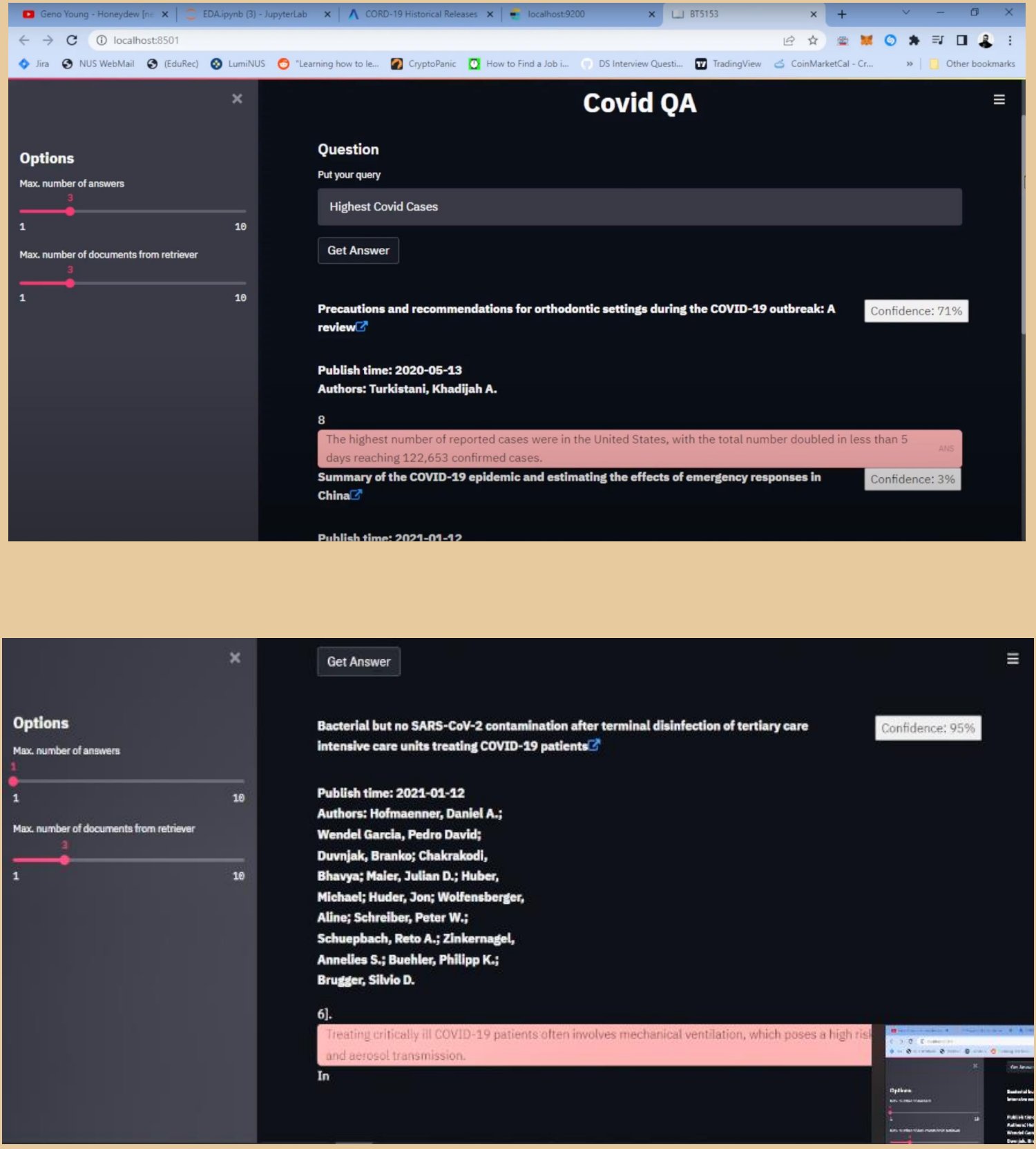
We started by pulling elasticsearch image from the docker hub, ran elasticsearch docker and mapped local server (port 9200) with the host server.
We ran a docker container in single node discovery type to setup our application.

Further to run the retriever model, we populated documents in the document store by converting the data to dictionary key value pairs to make it compatible with the elastic search format and used elastic search to access the document store for data by running it on Docker Image.

After setting the retriever model, we initiated a query reader pretrained model "deepset/roberta-base-squad2-covid" from huggingface.co, a model trained specifically with large number of QA pairs from articles written by scholars on COVID-19 , and then setup Extractive QA Pipeline.

We setup the core component of the tool- the QA engine- which was built on Docker. It receives user request through its' API endpoint, reads the data from knowledge base, runs the QA pipeline and sends the answer back to the user.

In the frontend, we let the user select top best results from up to 10 number of documents.



Sample Overview of the Streamlit Web App

SECTION 4: Conclusion

In a nutshell, this project aimed to create an Extractive QA engine whereby information on COVID-19 could be made easily available to the general public.

Not only is this QA system beneficial to the general public, but also to governments and healthcare institutions, as it makes it possible to diffuse centralized, complete and accurate information about COVID-19.

The main steps involved in this project started with the gathering of research papers on COVID-19 and their cleaning and integration to a database.

Then, thanks to state-of-the-art Docker, Haystack, Elasticsearch and Streamlit, the outcome of this project is a user-friendly interface where the user can type a question and choose both the maximum number of answers and the maximum number of documents from the retriever.

The current prototype of the QA system is confined till the general knowledge of COVID- 19.

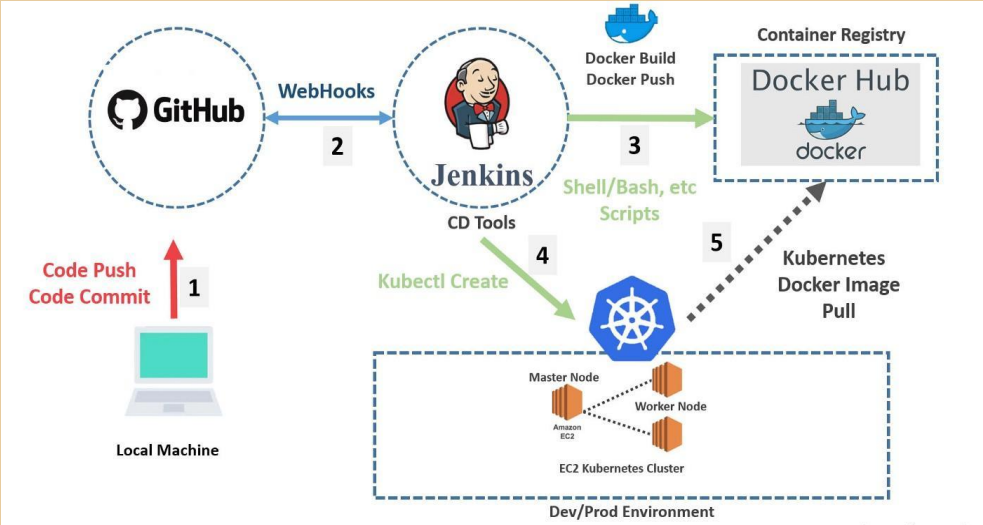
Different countries have different sets of measures in dealing with COVID-19. There are details of approaches how to handle COVID-19 by each country.

Further development can be done by including several countries approaches and training even more data to test out the accuracy of the model.

Gathering more data and adding more processing data as well as hyperparameter optimization will assist in improving the accuracy of QA system.

Dockerize the whole webapp and deploy to the cloud for faster outcome of answer to the queries as well as bigger datasets.

As the topic of the matter is delicate as it relates to public health matter, a robust evaluation of the system must be carried out to prevent further misinformation being trickled down to the public which cause unnecessary additional burden to the healthcare system.



High lebel next steps for the Web App Deployment And Continuous Integration Flow

SECTION 6: Appendix

Appendix 1: Metadata overview of CORD-19 dataset

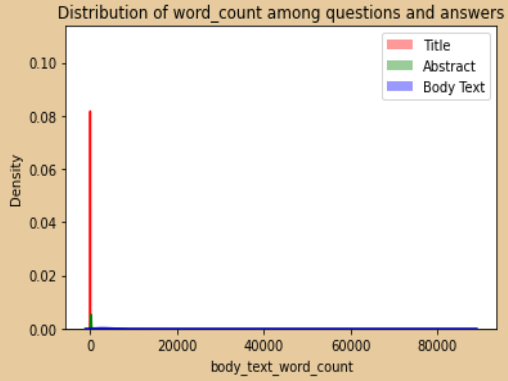
Feature	Data type	Description	Comments
cord_uid	str	Unique identifier to each CORD-19 paper	Not necessarily unique per row.As same paper can come from different sources.
sha	List[str]	Secured Hash Algorithm(SHA1) Id's of all the pdf's associated with CORD-19 dataset.	Mostly papers will have either zero or single value here .Some papers will have multiple indicating more than one pdf associated with single paper.
source_x	List[str]	Sources of paper	Single paper can have multiple sources
title	str	Paper title	
doi	str	Digital object identifier (DOI) of the paper	
pmcid	str	Paper's Id on PubMed Central	Begins with pmc followed by an integer
pubmed_id	Int	Paper's Id on PubMed	
license	str	License associated with the paper	
abstract	str	Paper's abstract	
publish_time	Str	Published date of the paper	In yyyy-mm-dd format

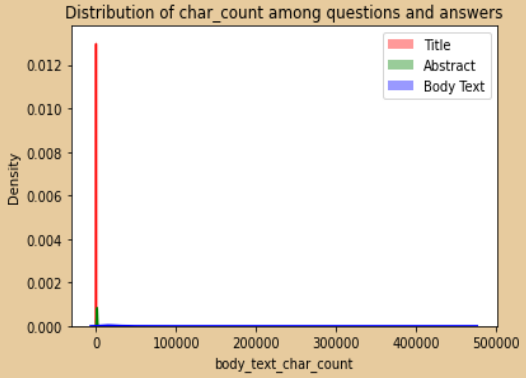
authors	List[str]	Authors of the paper	Each author name is in Last, First Middle format and semicolon-separated
Journal	Str	Paper journal	Strings are not normalized
mag_id	Int	Value field for the paper as represented in Microsoft Academic graph	Deprecated
who_covidence_id	Str	ID assigned by the WHO for this paper	
arxiv_id	Str	arXiv ID of the paper	
pdf_json_files	List[str]	Paths from the root of the current data dump version to the parses of the paper PDF's in Json format	Multiple paths are semicolon-separated
pmc_json_files	List[str]	Same as above but corresponds to the full text XML files taken from PMC,parsed in to the same Json files as above	Multiple paths are semicolon-separated
url	List[str]	All URL's associated with the paper	Semicolon-separated
s2_id		Semantic Scholar ID for this paper	Can be used with the Semantic Scholar API (e.g. s2_id= 1234 corresponds to http://api.semanticscholar.org/corpusid:1234)

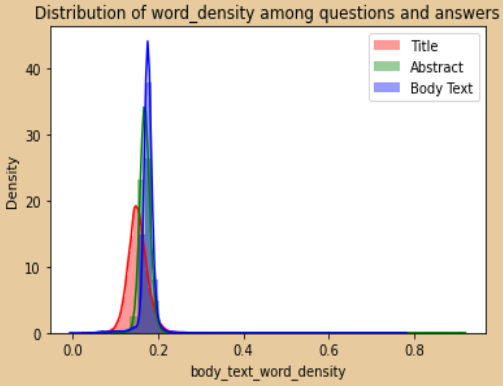
Appendix 2: Basic EDA on the text corpus

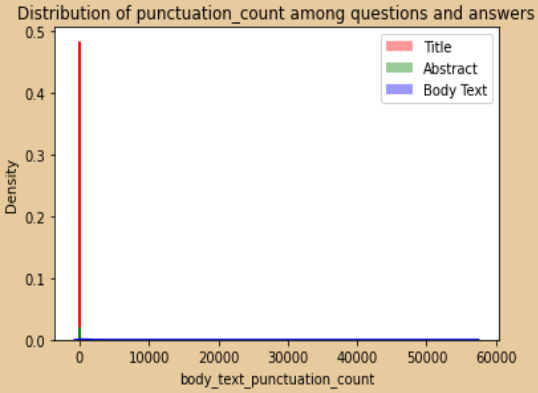
Some basic statistics on word counts, character counts, word density and punctuation count confirm that titles and abstracts contain less words and therefore characters and punctuation, and they have a

smaller word density. On the other hand, the length of papers measured in ‘body_text’ varies a lot and the word density is slightly higher.

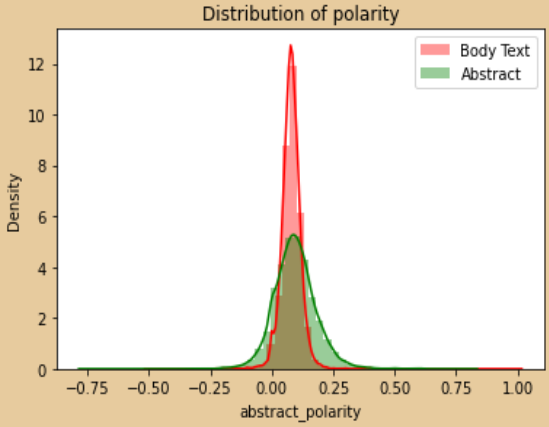


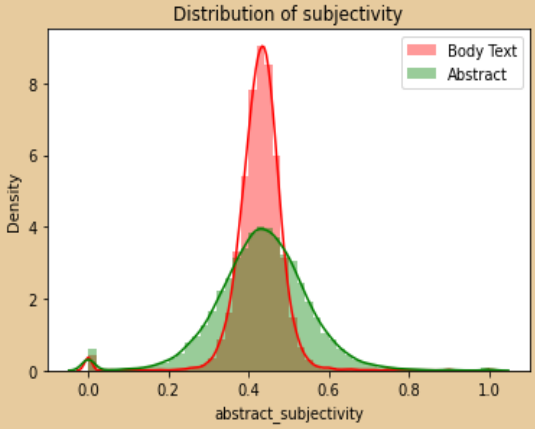






Let’s have a look at polarity and subjectivity. Polarity analysis takes into account the amount of positive or negative terms that appear in a given sentence. Subjectivity quantifies the amount of personal opinion and factual information contained in the text. We observe that the average polarity and subjectivity is slightly above zero for both abstracts and body texts.





SECTION 6: References

1. CORD-19: The COVID-19 Open Research Dataset
2. Allen Institute Datasets
3. Roberta-base-squad2-covid Model
4. https://github.com/deepset-ai/COVID-QA/blob/master/README.md
5. https://github.com/allenai/cord19
6. https://arxiv.org/abs/2004.07180
7. https://www.udemy.com/course/deep-learning-nlp-build-and-deploy-bert-covid-qa-system/
8. https://huggingface.co/deepset/roberta-base-squad2-covid