N_104-Projet-Informatique: Search engine for Covid-19

Jessica López Espejel

1 Introduction

Over the years, health issues have been studied by doctors and researchers. However, due to the pandemic situation, thousands of institutions and scientists are mobilized to tackle the Coronavirus (Covid-19). In consequence, the number of medical articles has been increasing exponentially. The following list points out some examples to show the growth of scientific articles in the last few years.

1. In 2020, the number of submissions to Elsevier's journals increased by 58% between February and May compared to the same period in 2019 [6].

2. The number of health articles increased by 92% in 2020, where scientists published more than 100,000 articles about the Covid-19 [4].

3. According to the EMC Digital Universe with Research and Analysis by IDC[1], there was an enormous growth in the global healthcare data between 2013 and 2020.

With the continuous increase in the amount of digital documents in the web, it is challenging for researchers and scientists to find articles related to the domain of interest. However, having an adequate tool can save time considerably. For instance, a search engine is a good choice to find the most related articles to the keywords one is searching for. Inspired by [7] we propose you in the following sections to build your own search engine for Covid-19.

[1] https://www.idc.com/

2 Implementation

2.1 Tools

In the following list, you find the tools that we strongly recommend you in order to develop your search engine.

- Python $\geq$ 3.6
- Pandas
- Tokenizers such as scispacy, spacy, nltk, etc.
- Sklearn
- Whoosh as a search server.

Note: If you want to use extra tools, you are free to do it.

2.2 Dataset

COVID-19 Open Research Dataset (CORD-19) [1] was proposed by the White House and research groups. It consists of 33.31 GB of information that includes target tablets, ready-to-use embeddings, and json articles. To develop your search engine, you will use a subset of this dataset. You can download it from this **link**.

3 Classes and methods

Before starting coding, it is highly recommended that you analyze the classes and methods you need to solve the problem (the semantic part of the solution). For instance, you can start by writing the skeleton of the code and gradually developing it. The advantage of working progressively in an organized way is to detect troubles with your conception and code. Thus, it is easier to rectify the mistakes and ask for help on-time.

Hereafter, we provide you with three classes that you can consider to start your project:

1. **Main class -** In this class, you can instantiate objects and call methods to run the search engine.
2. **Pre-processing class -** This class contains the different methods you will use to clean your data.

3. **Index class -** I will give you the implementation of this class. It is not mandatory to use it.

4. **Graphical user interface**.It is not mandatory but considered as a bonus.

Note that the classes provided above are recommended but not mandatory. You are free to make the conception you think is suitable for your solution. The code skeleton can be found in the Github repository.

4 Evaluation

4.1 Main criteria

You will implement a search engine to get the most related n articles with input keywords. The goal of your search engine is to list the titles of recommended articles containing relevant information the user is searching for. The main tasks we expect from you are as follows:

- Implement a class containing methods to clean your data.
- Use a tokenizer to separate the text into tokens. You have to explain in your final report how the tokenizer you select works. Nowadays, there are many kind of tokenizers. However, we recommend you to use scispacy [5]; which was designed for biomedical data, or others such as nltk [3], or spacy [2].
- Implement an optimal index using Whoosh search server (you can use the one in my Github, but you have to understand carefully how it works)
- All the names of your attributes, methods and classes should follow the PEP-257 standard to document your software.
- The number of code lines does not define the quality of your code. You may write a short, clear and effective code. Avoid repetitive code.
- The Github activity should be frequent.
- I will give you some unit tests to validate the performance of your search engine. However, you can write your own unit tests to help yourself to detect some drawbacks in your system.
- **You should understand the functionality of each line in your code**.

4.2 Final report

Your final report should contain the following items:

- Link to your Github repository.
- Project introduction
- Explain how you and your partner share the tasks.
- Explain the libraries you used to resolve the problem. In Section 2.1, we mention the tools we strongly recommend for your implementation. However, if you use other tools, you should mention which are those libraries and the reason you used them.
- Explain the main difficulties you found in the process to build the search engine. Moreover, explain how you resolved them.
- Project conclusion.

4.3 Evaluation percentages

Based on the above specifications, you find in the following list the percentage of mark used for each criterion:

- **Source code - 25%**. It is important the presentation of your code, because it shows the organization of your project.
- **Defense of your project - 20%**. In the presentation you should explain:
  - Introduction
  - Goal of your project
  - How did you resolve the problem (include the tools, code functionality, the main challenges you faced, and how did your solve them)
  - Conclusion
- **Progressive advance - 25%**. Each commit will let us know your advancement during the project.
- **Analysis - 30%**. In your final report (Section 4.2) the explanation of your solution is critical. You should know the reason and the functionality of each method and class.

** **Plagiarism is prohibited** ** **Contact**: jessicalopezespejel@gmail.com

References

[1] Covid-19 open research dataset challenge (cord-19). https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge. Accessed: 2021-03-26.

[2] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[3] E. Loper and S. Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.

[4] nature. How a torrent of covid science changed research publishing — in seven charts. https://www.nature.com/articles/d41586-020-03564-y, 2020.

[5] M. Neumann, D. King, I. Beltagy, and W. Ammar. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, Aug. 2019. Association for Computational Linguistics.

[6] F. Squazzoni, G. Bravo, F. Grimaldo, M. Garcia-Costa, Daniel Farjam, and B. Mehmani. Only second-class tickets for women in the covid-19 race. a study on manuscript submissions and reviews in 2329 elsevier journals. *Elsevier*, 2020.

[7] D. Wolffram. discovid.ai: a search and recommendation engine for cord-19. Online; accessed 27 March 2021.