Designing a Search Engine for Covid-19 Publications

The current Covid-19 (SARS-CoV-2 virus) pandemic has highlighted how, in the modern digital age, readily available information from multiple sources can lead to entrenched, dichotomous and multipolar views held with conviction, and thus the importance of information retrieval (IR) systems that provide *reliable* information. As an exercise in understanding and appreciating the technical details of how search engines work, this miniproject aims to build a fully functional model search engine (small scale), and to evaluate its performance in terms of the validity (*relevance*) and also "trueness" (precision) of the information it returns to simple and complex queries.

Following extensive exploration for mutidisciplinary, reproducible peer-reviewed data sources, the dataset of choice now is the CORD-19 dataset (Lucy Lu Wang *et al*, 2020).

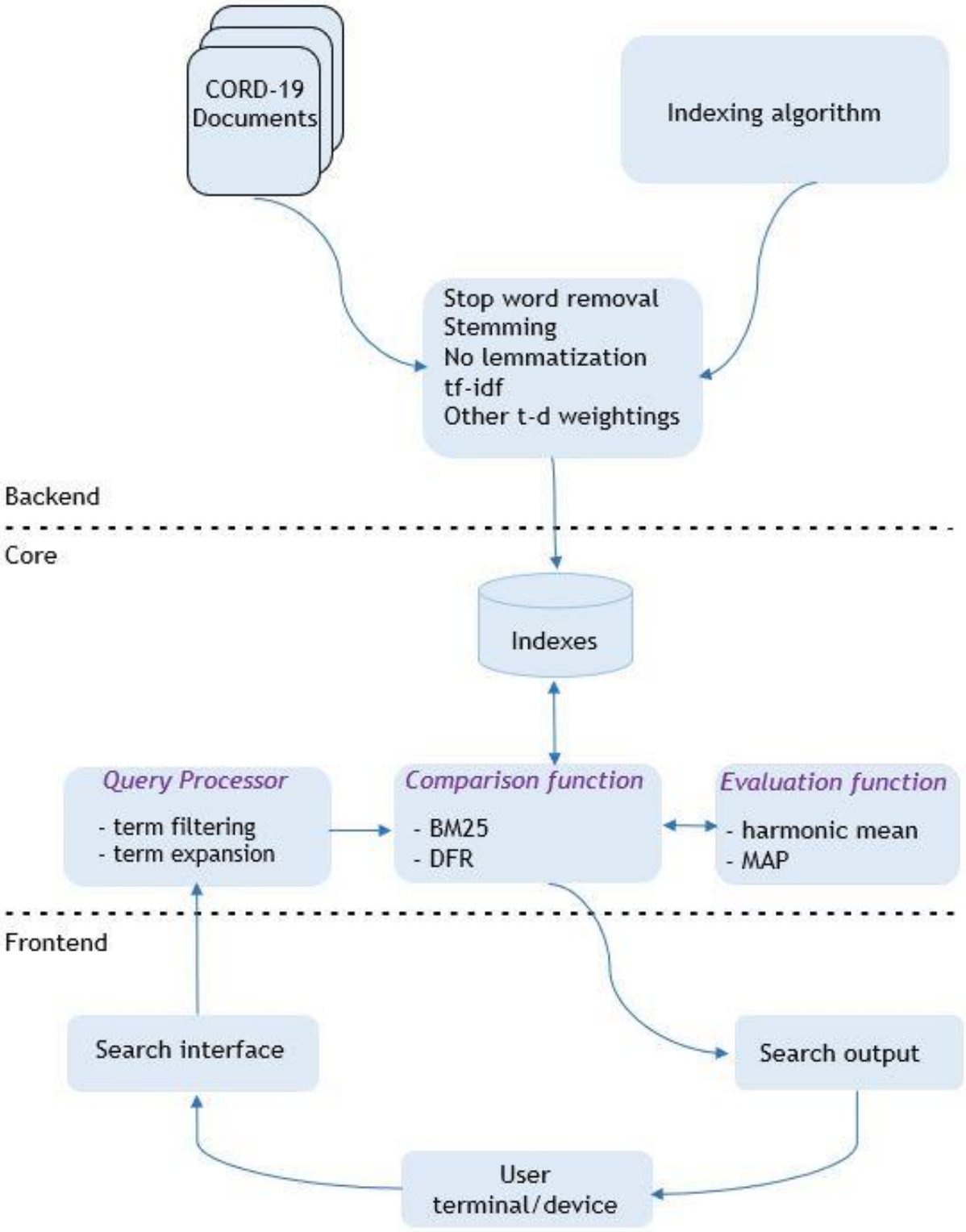
Dataset
The data feed for the search engine would be the "19th May 2020" version of CORD-19 dataset, which is also the version of the 3rd round of the TREC-COVID Information Retrieval Challenge on Kaggle (online reference 1, 2020)
Justification: On the topic of Covid-19 pandemic, this dataset consists of a collection of arguably the most rigorous scientific studies and authouritative findings on the subject matter, from virology and molecular biology through intensive care medicine and surgery to immunology and genetics. The dataset was designed and developed purposefully to facilicate text mining and information retrieval methodologies. It comprises harmonised metadata and parsed structured full text.

Crawler
This project would ***not*** require a crawler, as the world-wide web would not be crawled for indexing. The database would consist of the dowloaded dataset capable of being stored on a single local host, and does not require a distributed system either.
Justification: No justification required.

Indexer/Index
Stop word removal and stemming would be implemented using standard scikit-learn vectorizer modules, but no lemmatization would be carried out. This is because *context* in this subject matter is very important, and the judgement is that, other than the precision loss it brings, lemmatization would also affect context. Term and document weights including tf-idf scores would be obtained for all documents and indexed according to various criteria. Index would be stored locally or on cloud account.

Justification: Standard pandas, numpy and scikit-learn libraries adequetely compute term and document weightings as well as convert them to BM50 and DFR score quantifications. The goal is to keep this stage free from potential bugs.

Figure 1: Search Engine Architecture



Query Processor
Implicit recognition and separation of query terms from operators would be carried out using adapted python code.
Justification: Simplicity.

Comparison/Retrieval Function
Retrieval models to be utilised are the Best Match 25 (BM25) and Divergence from Randomness (DFR) probabilistic models, with BM25F (Robertson and Zaragoza, 2009) and Elasticsearch (Li and Wang, 2015) implementations as tentative models to be considered in conjunction with these if time permits.

If elasticsearch is implemented, its REST APIs will access the dataset directly at source via HTTP protocols and would therefore utilitise, among others, *urllib.request*, *BeautifulSoup*, and *time* packages.
Justification: The CORD-19 dataset consists of documents from specialised academic fields and would therefore comprise a large proportion of technical language. BM25's abundance of tuning constants could, if expertly used, enhance detecting and matching of nuances in technical language. It is reckoned, for the same reason of high proportion of technical language, that the binomial as well as the 2-Poisson distributions as randomness models in DFR would both be good retrieval model choices.

Elasticsearch 7.x, if implemented, would provide further experimentation and exercise in its unique set of advantages. This is however counterbalanced by the fact that many of elasticsearch's advantages – e.g. distributed systems, changes in data records, scalability – are not applicable to the search engine design in this miniproject.

Evaluation Function
Binary as well as multi-ranked relevance scores would be calculated using standard python code following the examples in Week 6 Lab 4, and from the lierature and GitHub. Effectiveness measures would be calculated as well as Harmonic Mean (F-score) (Lipton *et al*, 2014) and Mean Average Precision (MAP) (Sanderson and Zobel, 2005). Normalized Discounted Cumulative Gain (nDCG) (McSherry and Najork, 2008) would also be attempted, for which *ndcg_score* module from the *metrics* library of scikit-learn would be utilised.
Justification: F-score would enable combined term precision and recall measures whilst MAP and nDCG would enable document ranking ability of the engine to be measured.

Responsibilities
Being a single-member group (Group 68), I will be responsible for developing the entire engine.

Time Plan

- Week 9 : Wider reading and research.
- Week 10 : Dataset/feed setup, feed-indexing code development.
- Week 11 : Comparison and Evaluation functions platform development.
- Week 12 : Query processing interface development, search engine unit, component and integration testing.

Scenarios to be Investigated

A variety of scenarios would be investigated - from progression of disease severity, through types of treatment, to prognosis. Attempts would be made to log whether the correlations of these pandemic attributes to population demographics is apparent in the retrieved documents of appropriately worded query terms.

References

1. Lucy Lu Wang et al 2020, *CORD-19: The COVID-19 Open Research Dataset*, arXiv:2004.10706v4 [cs.DL], [online]: https://doi.org/10.48550/arXiv.2004.10706

2. Online reference 1 (2020), https://www.kaggle.com/c/trec-covid-information-retrieval/data?select=qrels.csv

3. Robertson, S. and Zaragoza, H., 2009, *The probabilistic relevance framework: BM25 and beyond*, Now Publishers Inc. [online]: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.437.660&rep=rep1&type=pdf

4. Li, X.M. and Wang, Y.Y., 2015, *Design and implementation of an indexing method based on fields for elasticsearch*, 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), IEEE, pp. 626-630. [online]: https://ieeexplore.ieee.org/abstract/document/7405916

5. Lipton, Z.C., Elkan, C. and Narayanaswamy, B., 2014, *Thresholding classifiers to maximize F1 score*, arXiv preprint arXiv: 1402.1892. [online]: https://arxiv.org/abs/1402.1892

6. Sanderson, M. and Zobel, J., 2005, *Information retrieval system evaluation: effort, sensitivity, and reliability*, Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 162-169. [online]: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.492.6958&rep=rep1&type=pdf

7. McSherry, F. and Najork, M., 2008, *Computing information retrieval performance measures efficiently in the presence of tied scores*, European Conference on Information Retrieval Springer, Berlin, pp. 414-421. [online]: https://marc.najork.org/pdfs/ecir2008.pdf