Design and Implementation Paper: Health Information Search

Toghrul 2020280126
abbaslit10@mails.tsinghua.edu.cn

June 2021

Abstract

When search engines are used for complex and important health-related topics, people face unverified claims and disinformation. There is more open room for that when it comes to COVID-19. Existing works propose the usage of scientific articles and papers. In this project, I have implemented a search engine for pandemics-related topics based on the CORD-19 corpus. Title and abstract base Lucene index was build using the Pyserini library. BM25 ranking model is used for document retrieval. As a result, retrieved titles and abstracts matched the query. Evidence sentences are highlighted in the abstract using the SciBERT Pre-trained Language Model, a BERT PLM pre-trained on scientific articles. The proposed method can be useful for claim verification, but tracking social network discussion forums can be more effective against disinformation. Today's expert communities are more engaged in social networks, so searching discussion forums can be efficient for the real-world claim assessment. Classifying search results according to stance is the open new research question. With enough performance, such a search engine may be useful for both the research community and daily life.

1 Background and motivation

Today, the usage of search engines became a crucial part of our daily life. People are searching about a variety of topics that are also of paramount importance and even critical for life. One of the striking examples of such topics is public health information. In 2015, one in every 20 searches was about health on Google search engine[1]. Search Engines can describe the primary informational needs of the Online Health Information Seekers regarding well-established disease symptoms and remedies[1].However, there is a specific query type called Multi-Perspective Consumer Health Information in which there are multiple supporting and opposing views where there is a need for balance. For example, given the query "Can Sun exposure lead to skin cancer?" there is no single correct answer. In addition, search engines face the filter bubble effect which means they can be biased to the party with more views of a specific stance[2]. Consequently, search engine users can face lots of claims and the spread of disinformation. However, information retrieval methods with collective intelligence

can address the issues. The typical fact-checking pipeline should consider real-world claims, select evidence sentences that can support or refute the claim and predict the veracity based on the evidence. Also, such a pipeline should not be bounded to the collection with already available personal validations such as Wikipedia. A dataset for fack-checking was proposed for NLP tasks which contains claims from the digital journalism project[3]. LSTM and CNN-based methods were proposed to automatically identify debates from posts about complementary and alternative medicine (CAM) in the online health community[4]. But they focused only on a specific community and controversial topics were biased by the prevalence of the therapies in the population and amongst forum members. The reproducibility of existing stance detection methods on health-related online news articles was analyzed in [5]. The authors also used BERT pre-trained language model[6].As disinformation is even more observant around the recent COVID-19 pandemic, I decided to build a search engine that can be used for claim verification by extracting evidence sentences from existing scientific articles and papers. I got motivation from a need for a search engine based on controversial public healthcare topics by a close family friend who survived cancer. The proposed search engine implements abstract and title-based search by highlighting evidence sentences.

2 Related work

The effectiveness of abstract and title-only searching methods is analyzed in [7]. Although users searching full text are more likely to find relevant articles than searching only abstracts, treating an entire article as an indexing unit does not consistently yield higher effectiveness compared to an abstract-only search. Not only, full-text may require much more computational resources, but also search engines could be biased towards significantly longer articles. Some works implement fact-checking by using Wikipedia data[8][9], however, those works do not consider real-world claims. Another line of work that includes Multi-FC[10] provides real-world claims collected from fact-checking websites and evidence documents and other meta information, but it does not extract evidence sentences. It is also notable that, CORD-19 corpus is proposed by [11]. Allen Institute for AI has partnered with leading research groups to provide CORD-19, a free resource of more than 280,000 scholarly articles about the novel coronavirus for use by the global research community. The idea for the proposed search engine is inspired by [12]. The authors provided the dataset which classifies the search results given the Multi-Perspective Consumer Health Information(MPCHI) queries. Also, they proposed stance vectors considering biomedical semantic relations alongside standard BoW features. However, the authors only consider 5 MPCHI queries. Another notable work is about generating a summary of controversial topics considering the stance[2]. The authors work on retrieving summary tweets of controversial topics considering stance-indication using hashtags, articulation, and topic relevance. Their work is also limited to specific topics.

3 System Design overview

Overall system design is given in Figure 1. Articles and papers are retrieved from the CORD-19 corpus[11], as it has several novel scientific works and is common for research. Pyserini library[13] is used to build the Lucene index[14] on the CORD-19 corpus. It is a document structure that is optimized for the BM25[15] ranking model. Flask backend server is interacting with index to get relevant documents. Flask is python based server-side programming language[16]. Also, evidence sentences in the abstracts are highlighted using SciBERT pre-trained language model[17] by the backend server. Finally, React-based client-side code sends requests to the backend server as the end-user interacts with the User Interface(UI). React[18] is a JavaScript-based client-side framework that is optimized for building fast Single-Page Application(SPA).

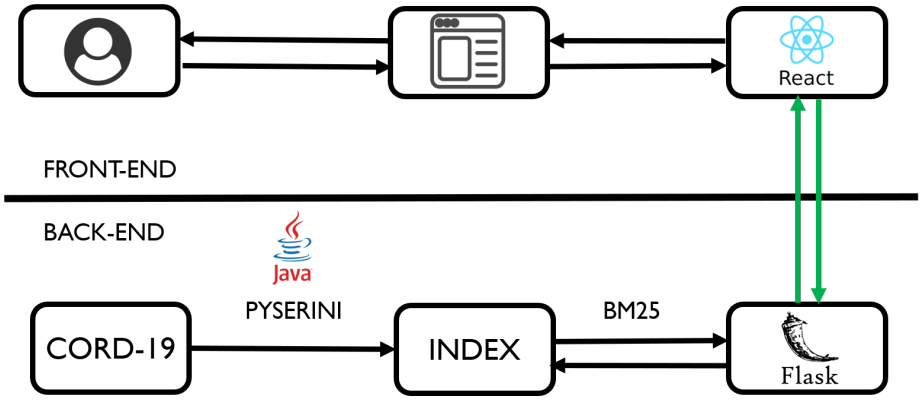

Figure 1: Overview of System Design

4 Technical details

As noted before CORD-19 corpus[11] is used for retrieving articles and papers. Corpus with recent articles is used for the project. Then, the pyserini library is used to generate lucene index[13]. Lucene is a Java programming library that generates indexes with programming-friendly document structures. The Lucene document structure is optimized for fast retrieval with the BM25 ranking model. With such a structure, certain fields of a retrieved document can be accessed easily. Pyserini library enables the usage of this structure with Python programming language, by wrapping around Java JDK. Now, it is possible to use Machine Learning tools and IR tools easily with Python. BM25 is a strong baseline method and many dense models underperform it for zero-shot evaluation[19]. In the Flask backend, only one route is responsible to get query and return search results to the React frontend. JSON[20] format is used to transfer data between the frontend and backend server. For the demo search engine, only the first 10 results for the BM25 search are returned. After that, the SciBERT[17] pre-trained language model is used to assess the semantic relation between query and retrieved document by highlighting the evidence sentences in the abstract. SciBERT is BERT PLM which is pretrained on scientific data. The BERT

model architecture [6] is based on a multilayer bidirectional Transformer [21]. Transformers are novel Seq2Seq models which utilize Attention mechanism and Positional Encodings instead of Recurrent Neural Networks(RNNs). Instead of the traditional left-to-right language modeling objective, BERT is trained on two tasks: predicting randomly masked tokens and predicting whether two sentences follow each other. The model is trained on a corpus with the size of 1.14M papers from *semanticscholars.org*. The average paper length is 154 sentences (2,769 tokens) resulting in a corpus size of 3.17B tokens. We can get the contextualized vectors of the query or abstract:

$$q_1, ..., q_T = SciBERT(query)$$

Given the embeddings of query and abstract we then compute the cosine similarity matrix between the query and each paragraph:

$$A = [a_{ij}] \in R^{|query|x|abstract|}$$

where

$$a_{ij} = \frac{q_i^T \ abstract_j}{||q_i|| \ ||abstract_j||}$$

Finally, we pick the two highest-scoring words and highlight them with the max window of 10.

In the client part, React JS is used to build a single-page application in which only one HTML file represents the whole website. Javascript allows the transition between pages and browser local storage is used to store query to send request again. React JS is also helpful for rendering the HTML which was sent from a backend server. Also, the state hook of React makes the webpage dynamic to changes. When the search results are shown, the user can see the BM25 score, Title, Journal, Source, authors, and evidence sentences extracted from the abstract. After double-clicking the search result, it is possible to see the expanded version where evidence sentences are highlighted inside the abstract. After clicking the title, it is possible to land on the source website.

5 Conclusion and future work

In this project, I developed a search engine for claim verification about Covid-19 by searching novel scientific research articles and analyzing evidence sentences from Abstract. Although the demo application was based on the title and abstract of the paper, the search engine was able to get related articles given the query. Abstract highlightings were also informative and gave the main point about the search result.

As a future work, I would like to continue to work on performance as there are lots of computations during the search. Being inspired by the [12], a search

engine should also classify the search results whether they refute or support the query. Existing works are slow in terms of performance so it is a potential open research challenge. Especially, classifying according to stance is also a complex open problem as there are multiple targets[22]. Each query should be treated as a new target and classification should consider the contents of both the query and the article. The main goal is to build a search engine for overall public healthcare search. Only searching articles is not sufficient for preventing disinformation so there is a need to track the health-related discussions in social networks such as Reddit. Such applications are analyzed in [23]. As these social networks provide more real-world related topics and views of the community, I would like to build a search engine on healthcare subreddits of Reddit which could be used as a collective social search for overcoming disinformation. Also, zero-shot ranking methods could address performance issues[24].

References

[1] P. Ramaswami, "A remedy for your health-related questions: health info in the knowledge graph," Feb 2015. [Online]. Available: https://blog.google/products/search/health-info-knowledge-graph/

[2] M. Jang and J. Allan, "Explaining controversy on social media via stance summarization," in *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1221–1224. [Online]. Available: https://doi.org/10.1145/3209978.3210143

[3] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1163–1168. [Online]. Available: https://www.aclweb.org/anthology/N16-1138

[4] S. Zhang, L. Qiu, F. Chen, W. Zhang, Y. Yu, and N. Elhadad, "We make choices we think are going to save us: Debate and stance identification for online breast cancer cam discussions," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 1073–1081. [Online]. Available: https://doi.org/10.1145/3041021.3055134

[5] S. Ghosh, P. Singhania, S. Singh, K. Rudra, and S. Ghosh, "Stance detection in web and social media: A comparative study," *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, p. 75–87, 2019. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-28577-7_4

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[7] J. Lin, "Is searching full text more effective than searching abstracts?" *BMC bioinformatics*, vol. 10, p. 46, 03 2009.

[8] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: a large-scale dataset for fact extraction and VERification," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 809–819. [Online]. Available: https://www.aclweb.org/anthology/N18-1074

[9] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The FEVER2.0 shared task," in *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1–6. [Online]. Available: https://www.aclweb.org/anthology/D19-6601

[10] I. Augenstein, C. Lioma, D. Wang, L. Chaves Lima, C. Hansen, C. Hansen, and J. G. Simonsen, "MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4685–4697. [Online]. Available: https://www.aclweb.org/anthology/D19-1475

[11] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, K. Funk, R. M. Kinney, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. B. S. Stilson, A. D. Wade, K. Wang, C. Wilhelm, B. Xie, D. A. Raymond, D. S. Weld, O. Etzioni, and S. Kohlmeier, "Cord-19: The covid-19 open research dataset," *ArXiv*, 2020.

[12] A. Sen, M. Sinha, S. Mannarswamy, and S. Roy, "Stance classification of multi-perspective consumer health information," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, ser. CoDS-COMAD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 273–281. [Online]. Available: https://doi.org/10.1145/3152494.3152518

[13] P. Yang, H. Fang, and J. Lin, "Anserini: Reproducible ranking baselines using lucene," *J. Data and Information Quality*, vol. 10, no. 4, Oct. 2018. [Online]. Available: https://doi.org/10.1145/3239571

[14] A. Lucene, "Lucenetutorial.com." [Online]. Available: http://www.lucenetutorial.com/basic-concepts.html

[15] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 120–127. [Online]. Available: https://doi.org/10.1145/383952.383972

[16] "Welcome to flask." [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/

[17] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," 2019.

[18] "React – a javascript library for building user interfaces." [Online]. Available: https://reactjs.org/

[19] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models," 2021.

[20] "Introducing json." [Online]. Available: https://www.json.org/json-en.html

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[22] D. Küçük and F. Can, "Stance detection: A survey," *ACM Comput. Surv.*, vol. 53, no. 1, Feb. 2020. [Online]. Available: https://doi.org/10.1145/3369026

[23] A. Park and M. Conway, "Tracking health related discussions on reddit for public health applications," *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2017, pp. 1362–1371, 04 2018.

[24] S. Sun, Y. Qian, Z. Liu, C. Xiong, K. Zhang, J. Bao, Z. Liu, and P. Bennett, "Few-shot text ranking with meta adapted synthetic weak supervision," 2021.