SSW CS410 Project Proposal: Design a Results Cache for Popular Queries (System Extension)

Github Link: https://github.com/waelmb/Results-Caching-System

1. **What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.**
    a. Wael Mobeirek, wmobei2 - Captain
    b. Suchita Joshi, suchita3
    c. Shirley Mao, smao10
2. **What system have you chosen? Which subtopic(s) under the system? If it is not listed above, how is it related to the class?**
    The system we chose is 1-Search, which is a medical search engine that is customized based on medical speciality. We are going to build an extension to this system, in the form of a results cache for popular queries to improve the search efficiency. Currently, the search takes between 5-10 seconds for every query as it needs to crawl data sources in real-time since these data sources do not allow continuous crawling. We aim to improve the search performance by caching results that are previously crawled.
3. **Briefly describe any datasets, algorithms or techniques you plan to use.**
    a. Health forums and test queries: Dr. Karl & Sophia
    b. Keyword extraction approaches and tools: Yake
    c. Use topic modeling to get most popular searches/topics from forums (compare LSA, PLSA, LDA, and Ida2Vec and see what works best - can implement with scikit-learn and PyTorch; if we have time, we can also try using a transformer-based model like BERT)
    d. Use LRU cache to store and get most popular results on 1-Search
    e. Scraper: Selenium
    f. Data sources: PubMed, DynaMed, UpToDate, MayoClinic
4. **If you are adding a function, how will you demonstrate that it works as expected? If you are improving a function, how will you show your implementation actually works better?**
    We will be adding a function to allow caching of popular queries. There will be two parts: 1) identifying what the current popular medical query is and 2) caching the search results of that query. The function would work as expected if the query time for the current popular topic using our system is faster than 1-Search.
5. **How will your code communicate with or utilize the system? It is also fine to build your own systems, just please state your plan clearly.**
    We will design the cache as a separate microservice. It will run on the cloud separately in its own container. The service will cache results to an elasticsearch index, which can then be utilized by the search engine to fetch query results. The search engine will inform the service of the popular queries by REST API communication.
6. **Which programming language do you plan to use?**

Python for the backend. We will also utilize some web technologies for the frontend, including but not limited to: HTML/CSS/Django

7. **Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.**
    a. Identifying current popular queries from outside sources (web crawling) - 25 hrs
        i. Create crawler, crawl website, and parse relevant info (~10 hrs)
        ii. Identify the popular and relevant queries (> 15 hrs)
    b. Create and maintain the index for these popular queries on elastic search - 20 hrs
    c. Identifying current popular queries within 1-Search - 20 hrs
    d. Testing and cleaning up the service to make it more user friendly - 20 hrs