ECS736P - Search Engine Design

Group 22

1. Problem Statement and Approach

In this era of rapid development of Internet technology, the number of technical papers is also growing rapidly. There are a huge number of technical papers in the databases of the entire Internet. Due to the limited field of technical papers, there are many repeated words in different articles, so there will be limitations in querying the relevance of documents and query alone. Inspired by the PageRank algorithm, we believe that citations between papers can also be regarded as links to web pages, so we introduced page rank to our paper search engine. We used the Citation Network Dataset from Kaggle as our dataset.

Minimal Viable Product (MVP)

Experiment of paper retrieval model based on Citation Network Dataset from kaggle using Elastic Search (ES) on top of Lucene. Indexing model will be using stemming, lemmatization, stop words remove and tf-idf. We also will use PageRank to calculate a score for each paper. We will use BM25 as our retrieval method and cache the query result.

The objective is to learn how indexing, ranking work, what feature define the best model, is PageRank suitable for using on paper and why. We will evaluate our model in batch mode. We will manually collect the retrieval results from well-known paper retrieval websites such as arxiv to construct a dataset to evaluate the results of our model. We need our model to be a model usable in real life.

Enhancement time permitting

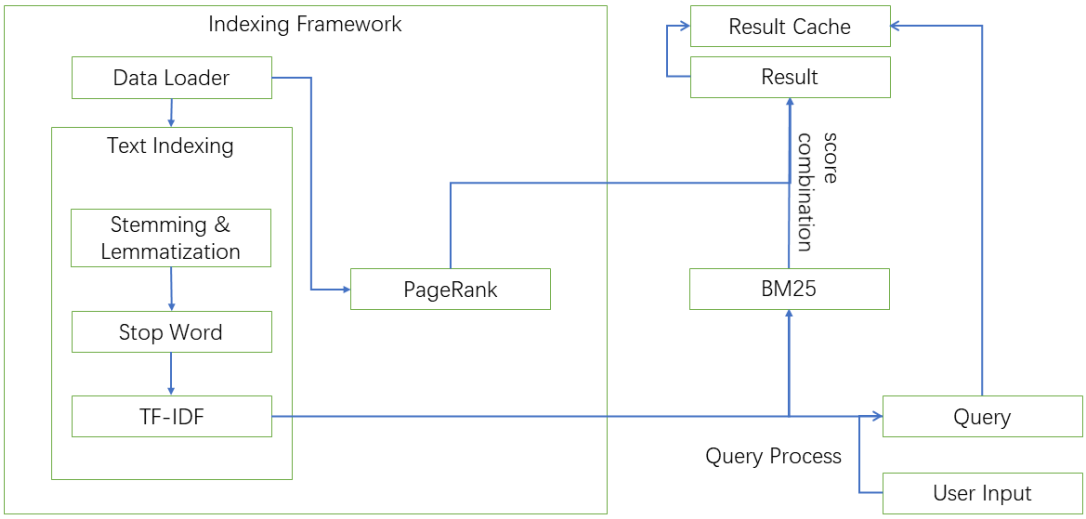
Use more complex algorithms and combinations of algorithms to optimize retrieval algorithms.

2. Datasets

We will apply Citation Network Dataset on Kaggle as the ground truth for search engine design and test.[1] The data set is designed for research purpose only. The citation data is extracted from DBLP, ACM, MAG (Microsoft Academic Graph), and other sources, which has 629,814 papers and 632,752 references. Each of them has relationship with their abstracts, authors, location, and titles.

3. Architecture

The diagram below illustrates the principle that the technical papers search engine will be developed.



The components are briefly described:

- **Data Loader** will process the Citation Network Dataset into a workable structure.
- **Text Indexing** will pre-process the data and extract the useful information for each term.
  - **Stemming & Lemmatization** will reduce inflectional forms and derivationally related forms of a word to a common base form.
  - **Stop Word** Removal will remove the words that have no significance.
  - **TF-IDF** will make aware the importance of each word in the document.
- **PageRank** will compute the rank of the papers.
- **Query Process** the **User Input** into **Query**.
- **BM25** will get the relevance for the Query and the paper.
- Both PageRank score and BM25 will be considered to get the Result by **score combination**.
- The **Result Cache** will store the query result so that future requests will be served faster.

4. Framework/Tools Employed

Python would be the first choice to develop APIs and infrastructure. We may choose Elasticsearch package as the elastic search tool. BM25 and NetworkX python package are introduced to implement algorithm.

Tool	Description
Python	Python will be used to develop search engine components.
Elasticsearch	Python Elasticsearch Client is a low-level version for python. It is very extendable.
BM25	BM25 is a collection of search and query algorithms, which is common in building search engine.
NetworkX	NetworkX is a python library for studying graphs and networks that includes

	PageRank.
GitHub	GitHub will be used for teamwork and version control.

5. Product Development Backlog

Product feature and technical backlog are as follow:

- Research technical paper relevance/precise benchmarks for given queries for Citation Network Dataset
- Initial domain model design (iterate through the product development processes)
- Configure paper-search project, library dependencies and build script for Citation Network Dataset
- Data loader to process the Citation Network Dataset into a workable structure
- Elastic Search feasibility analysis
- Text indexing for data pre-process: Stemming & Lemmatization, Stop Word and TF-IDF
- PageRank to calculate rank of the papers
- Query process: transform User input into Query
- BM25 Retrieval model
- Score combination: PageRank and BM25
- Result Cache
- Result analysis and presentation

6. Roles & Responsibilities

All team members are involved in development and analysis. Pair programming is adopted in this project.

Role descriptions are as follow:

Team member	Roles
Yinqing Gou	Project structure, data loader
Yingzhe Wan	Text indexing including stemming, Lemmatization, stop word and TF-IDF
Zengze Wang	PageRank and BM25 calculation, query Process
Yansen Xue	Score combination, result cache, result analysis

Reference

[1] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. ArnetMiner: Extraction and Mining of Academic Social Networks. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (SIGKDD'2008). pp.990-998. [PDF] [Slides] [System] [API]