F033583 Introduction to Web Search & Mining

Group Project

Final Report / Paper / Data / Code Due: June 19th

Demo scheduled in week 17~18

Introduction

This is a group-based project. Each group should contain maximum 3 students. In this project, there are four options. Each group must email TA (wsm2020spring@163.com) your choice. There should be **3 choices** in the email (First choice, Second choice and Third choice ranked by your preference). The topic of the email should be named as ***[Project Choice]+group leader's name+group leader's student id***. If a project idea is chosen by too many groups (each option can be selected about 1/4 groups), it will be allocated on a first-come, first-serve basis. The deadline of project selection is April 26th.

Option A

In this project, you are building a simple Chinese WestLaw system. You are **provided** with court records of legal cases in China in Chinese. Data are available at https://adapt.seiee.sjtu.edu.cn/wsm2020/wsm_proj1_data/. The data are in json format with detailed description. The json files store dictionaries where keys include person name, case code, city, etc. You are **not allowed** to crawl extra data from any sources.

You are required to build index and/or other data structures to support the following operations, and create a nice **web interface** for demo. You are **not allowed** to use any existing code from other people.

1. (about 20% scores) **Boolean search**. Users provide search keys and operations between keys. The system needs to return all the relevant documents or data. Types of search keys include but are not limited to a person name, a city, a case code, a court name etc. Operations include but are not limited to AND, OR, NOT, proximity, etc.

   Example user input:

   (1) 小明 (person name) AND 上海 (city). The system should return data containing both 小明 and 上海

   (2) 小明 (person name) AND NOT 上海 (city). The system should return data related to 小明 but not related to 上海.

2. (about 20% scores) **Tolerant (fuzzy) search**. While searching data with attributes including person name, city, case code, company name, etc., users may wrongly type some information and the system is required to do fuzzy search.

For example:

(1) Users search with court name "上海徐汇人民法院", the system should also return data containing "上海市徐汇区人民法院".

(2) Users search with case code "(2017)沪 0112 执 5984 号", the system should also return data containing "(2017)沪 0112 执 5983 号".

However, these fuzzy results should be ranked according to the relevance to users' search keys.

3. (about 20% scores) **Query**. Users are allowed to search for legal instruments (文书) with a query sentence. The instruments should be also returned according to the relevance to the query sentence.

Example user input: 该合同系双方真实意思表示，被告没有利用强势地位强行与原告签订合同。

4. (about 20% scores) **Ranking**. The system should provide options for users to sort the returned data or documents by amount of fines, registration dates, etc.
5. (about 20% scores) **Web interface** for demo.

Option B

In this project, you are asked to implement a distributed **crawler** that respect the politeness of the target servers (robots.txt). Then you are asked to use this crawler to crawl books and their metadata from the Internet and build a search engine that supports searching for **author, title, publish year, abstract (if available) and content**. The books you crawled must be the works in **English** by the winners of **Nobel Prize in Literature** and **Pulitzer Prize in Fiction and Non-fiction** (you need to collect a list of such authors first). You are **not allowed** to use any existing code from other people.

The final result will be considered from four aspects:
(about 80% scores total and 20% scores each)

1. **Quantity**: The number of authors and the number of books will be taken into consideration. You need to show how many results you have found in the search interface.
2. **Accuracy**: Only correct results will be rewarded. If the author of the book is not written by a Nobel Prize in Literature winner, this book will be punished. Also, if a book has obvious defects, such as lack of content, incorrect content, or incorrect language, it will be punished. At the time of submission, you will be asked to provide a list of all books, including title and author.
3. **Speed**: Given a query, you need to quickly return the ranked list. You should show the effectiveness and search time of each ranking algorithm you implement.
4. **Functionality:** You should build a full-featured search engine, including support for title, author, content search and an easy-to-use interface. The result provided by your search engine should be sorted by relevance.

Additional tasks
There will be additional rewards for the following tasks (about 20% scores total)

1. The search engine supports searching for **category** and **description** of book and can show the **picture cover** of the book when showing the results.
2. After click into one book, you can **divide the book into different chapters** and allow the user to select one chapter and **read it online**.
3. Crawl books from **more than one source.** (You need to include the source of the book in the final book list).
4. Your search engine can be accessed through the Internet, not just through localhost. (**If school doesn't reopen by the deadline of this project, we will need to check your homework remotely, and this will become a must instead of a reward**).

Option C

In this project, you are asked to build a search engine for Wikipedia with existing dumps. You can choose between Chinese and English, and process the Wikipedia dump into a clear and readable format.

Data Source (Choose one):
Wikipedia in English: https://dumps.wikimedia.org/enwiki/latest/
Wikipedia in Chinese: https://dumps.wikimedia.org/zhwiki/latest/

You are required to implement **ranking algorithms** to support 3 kinds of requirements.

1. (about 40% scores) **Support several ranking methods**. You need to implement at least five kinds of ranking algorithms. Note that you are not allowed to use the existing frameworks such as Lucene.
2. (about 30% scores) **Speed**. Given a query, you need to quickly return the ranked list. You should show the effectiveness and search time of each ranking algorithm you implement.
3. (about 30% scores) **Provide friendly search system GUI.** When user inputs a query, the query words should be highlighted in returned list. Users can click a result and go to the Wikipedia page in which you need to show the structured document.

Option D

This is a research project. In this project, you are asked to investigate and compare the searching algorithms of popular web search engines such as Google, Baidu and Bing. You need to write a research paper to show your findings.

You are required to meet 2 kinds of requirements.

1. (about 80% scores) **Content.** Your paper should be modeled and formatted after papers in SIGIR or WWW papers that evaluate the effectiveness of web search engines. You are required to ask at least **three** of the following research questions (you can ask additional research questions): 1) What are the main ranking algorithms used these three engines? 2) Are these engines' editorial content influenced by their advertisers? 3) How big are the web indexed by each of these engines? Can you give a breakdown on different types of web content: e.g., html files/images/videos? 4) How do these engines rank advertisements? 5) How do these engines defeat web spams? Your paper should clearly document your methodology or approaches to answer the above questions. Include your algorithms/model diagrams etc. to assist your narratives. Explain how you do the experiments and show evaluation results to support your findings.
2. (about 20% scores) **Format.** It should be written in English and follow the ACM format of research paper, and its length should be 10pt, double column, 9 pages + unlimited references. It must be organized with abstract, introduction with the motivations, method, evaluation, discussion and conclusion.

Deliverables

The final deliverables are different in specific options, please pay attention to what you need to submit for your option.

Item	Description	Option
Report	A well-written report to describe your ideas, design, implementation, example queries, results, conclusion, etc.	A, B and C
Research paper	A well-written research paper which follows the format mentioned in Option D.	D
Crawled data	Zipped archive of the entire crawled data.	B
Web demo	A web demo. (You need to display your demo to TAs before the due date.)	A, B and C
Source code	Source code of your whole implementation.	A, B, C and D