F033583 Introduction to Web Search & Mining

Group Project: Building A Search System

Final Report, Code and Demo Due: Saturday, June. 16, 2018

Specifications

In this project, you have three options in building a search system. The work includes crawling pages/media files, building the dataset, indexing data and creating a nice web interface for search.

For each option, there are some requirements. Please try you best to achieve these requirements. The more requirements you implement, the more scores you get. Also, if you design some other creative functions, you will get some bonus.

OPTION A

In this project, you are asked to crawl the text of books/literatures (both Chinese and English) from the web, including but not limit to full-length novels, novellas and short stories. You can find many websites which may contain those kinds of texts. There are some example sites which you can use or not:

Chinese:
http://book.chaoxing.com/
https://www.qidian.com/
https://www.duanwenxue.com/
http://www.aliwx.com.cn/

English
http://www.gutenberg.org/
http://novel.tingroom.com/
https://themillions.com/
http://therumpus.net/

Quantity target (for each of language):
Full-length novels: > 5000
Novellas: > 5000
Short stories: > 10000

You are required to build index and/or other data structures to support four kinds of requirements.

1. (about 50% scores) Free text queries, like search book's title and return a list of ranked books, search an author and return a list of books according to the year posted or other properties.
2. (about 30% scores) Manually design or automatic generate proper categories

and classify the crawled books into those categories.
3. (about 20% scores) Try to provide some comments or reviews for each book. Maybe, you can crawl these information from shopping websites such as Jindong, Amazon.
4. (Bonus) Design a beautify and practical book reading page. Users can click a book and go to the reading page.

OPTION B

OPTION B is very similar to OPTION A. Instead, you are asked to crawl at least three types of audio books from the web, such as full-length novels, novellas and short stories. There are some example websites which you can use or not:

Chinese
http://www.lrts.me/
http://www.tingbook.com/
http://www.ximalaya.com/explore/

English:
https://www.audible.com/
http://etc.usf.edu/lit2go/
http://esl-bits.net/

Quantity target (for each of language):
Full-length novels: > 3000
Novellas: > 3000
Short stories: > 6000

You are required to build index and/or other data structures to support four kinds of requirements. The first three requirements are the same as OPTIONA.

1. (about 50% scores) Free text queries, like search book's title and return a list of ranked books, search an author and return a list of books according to the year posted or other properties.
2. (about 30% scores) Manually design or automatic generate proper categories and classify the crawled books into those categories.
3. (about 20% scores) Try to provide some comments or reviews for each book. Maybe, you can crawl these information from shopping websites such as Jindong, Amazon.
4. (Bonus) Design a beautify and practical book listening page. Users can click a book and go to the listening page.

OPTION C

In this project, you are asked to build a simple financial news website. All your data should be crawled from the following websites. These are some Chinese financial news websites. Note that for each website we give, you need to crawl all the news starting

from year 2015 (including 2015) up to now.

Data Source (Chinese financial news):
http://www.eastmoney.com
http://finance.sina.com.cn
http://www.10jqka.com.cn
http://finance.qq.com
http://www.cnstock.com
http://business.sohu.com

Your data should be stored in plain text format. Each line should be a json of one document. For each document json should be like,
doc= {“content”: “xxx”, “source”: “xxx”, “time”: “xxx”, “title”: “xxx”, “url”: “xxx”}.

You are required to achieve following requirements:

1. (about 30% scores) Support several search methods. Firstly, user can just input some keywords, you need to search these keywords in the whole database and return a list of news. Secondly, user can choose search in the title or search in everywhere. Thirdly, users can limit which year of news they want when doing search.
2. (about 20% scores) Manually design or automatic generate proper categories and classify the crawled news into those categories.
3. (about 30% scores) Support news recommendation. After reading one or more piece of news, you need to find topics which are interesting for this user, and recommend some piece of news.
4. (about 20% scores) Your system should work in real-times. This means when one of the above websites publish new articles, you can catch them into your system with little delay.
5. (Bonus) Allow users to track some topics. When your system finds some latest articles which are related to one of these topics, you need to generate a new web page which arranges these related articles in a good manner. Then you need to send this generated web page to user’s email.

Deliverables

The final deliverables should include the following items:

- A well-written report to describe your ideas, design, implementation, example queries and results (with screenshots), conclusion, etc.
- A web demo deployed on any publicly accessible web server (in case you can’t find an accessible machine to host your code and data, you can deploy the server on your local computer and contact Yangyang or Jessie for a personal demo in her office, before the due date).
- Source code of the whole search system.
- Zipped archive of the entire crawled data.