Intelligent Browsing Search
mcroos2,miguelf4, rgyanm2, ak85
Team Captain: miguelf4

Overview

The objective of this project is to develop a Chrome extension for intelligent browsing. Specifically, we developed a Chrome extension that indexes the current page to allow users to search over the page using the OkapiBM25 ranking function from the metapy package. In addition, we implemented a frontend UI that the users can use to input a search string, and the top 5 results are displayed back to the user.
The text preprocessing, cleaning, indexing, and ranking functionalities are handled in the backend flask-api using the metapy package.

Implementation

The software is implemented in two components:

(1) Frontend – Chrome extension

The core UI is implemented in popout.html and popout.css, consisting of the UI code for search and display results. The extension is located in the chrome_intelli_search folder. backend.js file contains the event listeners to handle the search submission on click by extracting the text from the currently active tab. We utilize the chrome.scripting.executeScript to execute the text scraper functionality on the active window, POST and fetch data from the backend flask-api using the javascript 'fetch' method and dynamically render the resulting json results. We utilize basic HTML, CSS, Javascript, and default Chrome extension APIs to accomplish the frontend-backend communication and UI rendering.

(2) Backend - Flask API server

The backend server is designed with Flask-API, and the text search indexing/ranking functionality is developed using the MetaPy packages. All the required packages to run the server are in the requirements.txt file. The server is run locally for the initial implementation is per user. The flask-API server is located in the flaskapi folder.

The Flask-API implements a /search route that expects at least two parameters, the raw text from the HTML page(corpus) and the search string. Optionally, a ranker option can be passed to test out other available rankers. However, for this project, we implement OkapiBM25 as the default ranker.

The search engine pipeline is a three-stage pipeline with Preprocessing, tokenization/stemming, and indexing/ranking stages.

Preprocessing Pipeline:

1. Split lines
2. Get rid of lines with one word or less
3. Create data set each sentence in a line
4. Write to dataset file from the configuration file

Tokenization/Stop Word removal/Stemming/N-Grams

We utilize the Metapy's rich API to perform further preprocessing of the documents as follows:

1. ICUTokenizer to tokenize raw text and provide an Iterable that spits out the individual text tokens.
2. Lowercase to convert all tokens to lowercase
3. Porter2 Stemmer to stemmer, or lemmatize the tokens by reducing the words from its inflections.
4. Ngram-word-1 to bag of words" representation or "unigram word counts".

Indexing and Ranking

An inverted index is created using the metapy's make_inverted_index function and BM25 ranker is instantiated with the parameters k1=2.4, b=0.75 and k3=400. A query document is created using the search string and the ranker is queried to score and return the top-5 results.

Project Environments:

1. Python 3.7, Metapy, Flask-API
2. Chrome

Testing the Backend :

Sample Server Request/Response

```
$:~/code/admin/ill/CS410Text/CS410-CourseProject-Team-WXYZ/flaskapi

$ curl -X POST -H "Content-Type: application/json" -d '{"corpus":"Hello Inteli Searcher",
"search": "searcher", "ranker": "test"}' -i http://127.0.0.1:5000/search

HTTP/1.0 200 OK

Content-Type: application/json

Content-Length: 45

Access-Control-Allow-Origin: *

Access-Control-Allow-Headers: Content-Type,Authorization

Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS

Access-Control-Allow-Credentials: true

Server: Werkzeug/1.0.0 Python/3.7.6

Date: Fri, 09 Dec 2022 04:30:45 GMT


{"search_results":["Hello Inteli Searcher"]}
```

Usage:

Requirements: Python Version 3.7

1. **Run Flask-API server**

   $cd /flaskapi

   Optional:

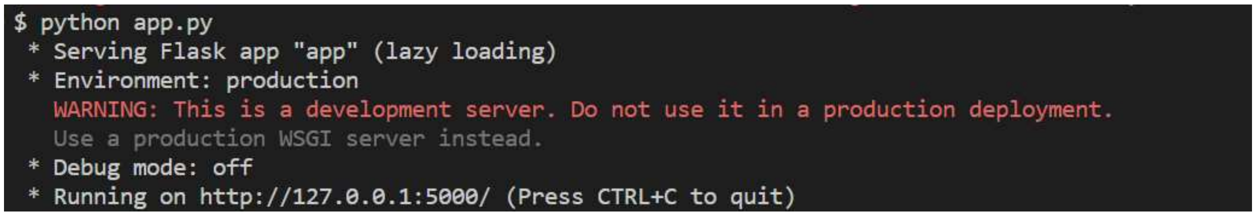
   $ conda create -n testenv python=3.7

   $ conda activate testenv

   **This project was tested on python version 3.7. Please use a Python version 3.7 environment.

   $ pip install -r requirements.txt

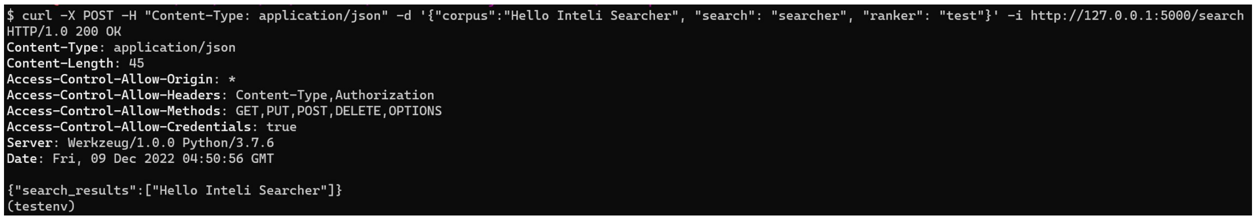
   $ python app.py



```
$ python app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

   Testing the API:

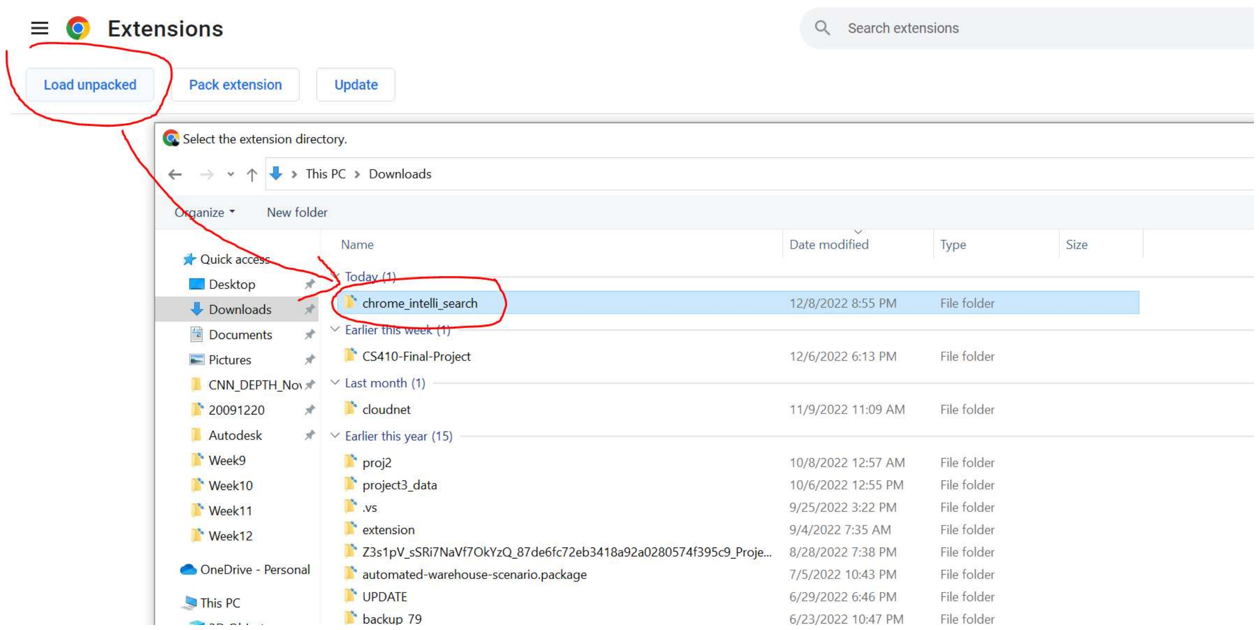
   $ curl -X POST -H "Content-Type: application/json" -d '{"corpus":"Hello Inteli Searcher", "search": "searcher", "ranker": "test"}' -i http://127.0.0.1:5000/search



```
$ curl -X POST -H "Content-Type: application/json" -d '{"corpus":"Hello Inteli Searcher", "search": "searcher", "ranker": "test"}' -i http://127.0.0.1:5000/search
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 45
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type,Authorization
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Allow-Credentials: true
Server: Werkzeug/1.0.0 Python/3.7.6
Date: Fri, 09 Dec 2022 04:50:56 GMT

{"search_results":["Hello Inteli Searcher"]}
(testenv)
```

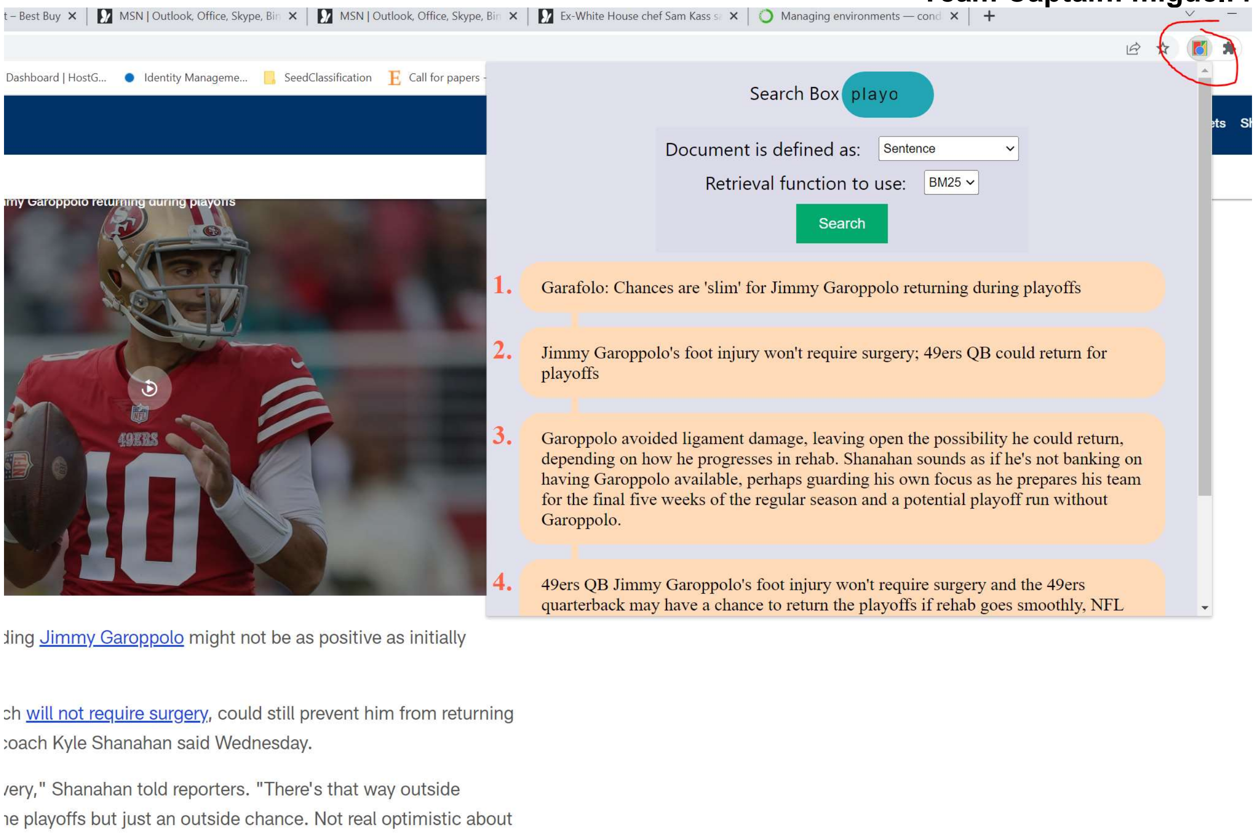
2. Install Chrome Extension



Name	Date modified	Type	Size
Today (1)			
chrome_intelli_search	12/8/2022 8:55 PM	File folder	
Earlier this week (1)			
CS410-Final-Project	12/6/2022 6:13 PM	File folder	
Last month (1)			
cloudnet	11/9/2022 11:09 AM	File folder	
Earlier this year (15)			
proj2	10/8/2022 12:57 AM	File folder	
project3_data	10/6/2022 12:55 PM	File folder	
.vs	9/25/2022 3:22 PM	File folder	
extension	9/4/2022 7:35 AM	File folder	
Z3s1pV_sSRi7NaVf7OkYzQ_87de6fc72eb3418a92a0280574f395c9_Proje...	8/28/2022 7:38 PM	File folder	
automated-warehouse-scenario.package	7/5/2022 10:43 PM	File folder	
UPDATE	6/29/2022 6:46 PM	File folder	
backup 79	6/23/2022 10:47 PM	File folder	

3. Browse & Search

Intelligent Browsing Search
mcroos2,miguelf4, rgyanm2, ak85
Team Captain: miguelf4



t – Best Buy
MSN | Outlook, Office, Skype, Bin
MSN | Outlook, Office, Skype, Bin
Ex-White House chef Sam Kass s
Managing environments — cond
Dashboard | HostG...
Identity Manageme...
SeedClassification
Call for papers
Search

Search Box playo

Document is defined as: Sentence

Retrieval function to use: BM25

1. Garafolo: Chances are 'slim' for Jimmy Garoppolo returning during playoffs

2. Jimmy Garoppolo's foot injury won't require surgery; 49ers QB could return for playoffs

3. Garoppolo avoided ligament damage, leaving open the possibility he could return, depending on how he progresses in rehab. Shanahan sounds as if he's not banking on having Garoppolo available, perhaps guarding his own focus as he prepares his team for the final five weeks of the regular season and a potential playoff run without Garoppolo.

4. 49ers QB Jimmy Garoppolo's foot injury won't require surgery and the 49ers quarterback may have a chance to return the playoffs if rehab goes smoothly, NFL

ımy Garoppolo returning during playoffs

ding Jimmy Garoppolo might not be as positive as initially

ch will not require surgery, could still prevent him from returning
coach Kyle Shanahan said Wednesday.

very," Shanahan told reporters. "There's that way outside
he playoffs but just an outside chance. Not real optimistic about

Contribution of Team:

All team members participated in all the backend and frontend work. Lot of hours were spent on learning the technology and getting up to speed on tasks such as developing Chrome extensions and developing in Flask-API.