

CS410 Project Documentation

Team name: Night Crawlers

Team Members:

Jiawei Yuan (Captain)

jiaweiy3@illinois.edu

Ziqi Xu

ziqixu3@illinois.edu

Ziyao Zhang

zhang416@illinois.edu

How to Use

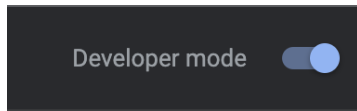
Demo Video

https://drive.google.com/file/d/1jqMBgYiXVSA6bAbrB5yDwbMnoRr2kC18/view?usp=share_link

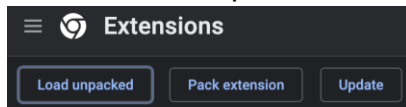
How to install the extension (Similar to steps in MP2.1):

Frontend:

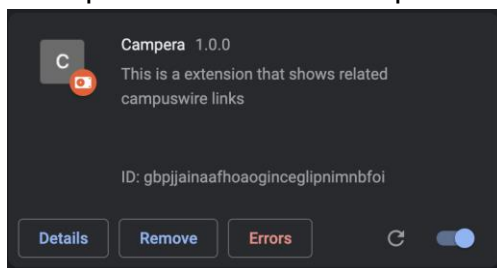
1. Download the extension folder from the github repository
2. Go to the “Manage Extensions” Page on chrome
3. Turn on developer mode



4. Select “Load unpacked” from the top left corner



5. Select the extension folder
6. “Campera” will then show up in the extension library



Backend:

1. Download the crawler folder from the github repository
2. cd to the crawler folder
3. pip install flask
4. run export FLASK_APP=main.py
5. flask run
6. The server side code will be running locally on <http://127.0.0.1:5000>. For testing purpose, you can add a query parameter to the url to see the matched post information. Upon starting up / when the API is hit for the first time, a crawler will scrap the most recent posts from campuswire.

How to use the extension:

1. Paste/Enter the keywords into the input box in the extension pop up
2. Click the search button

Related Campuswire Posts

Key Word:

3. Related posts on campuswire will show up as lists

Related Campuswire Posts

Key Word:

- Letter Grade
- Minimum grade on exam category to pass class?
- Grade for MP1 not updated
- Grade for MP 1
- Question about UIUC Grading

4. If the user click on one of the post titles, the post will be opened in a new tab

Implementation

- Frontend

A google extension was implemented as the front-end interface of the application. HTML, CSS and Javascript were used for the layouts and scripts.

The user can enter keywords into the input box in the popup to query related campuswire posts. The title of the related posts will be displayed as a list with hyperlinks. If the user clicks on the post title, the post will be opened in a new tab, and the user will be redirected to the campuswire post page.

- Backend

Main

main.py includes the steps for most of our backend execution logic. It was implemented on top of a Python Flask template and would interact with the frontend through HTTP request. Upon starting up it will call the crawler to crawl the campuswire posts. When a user inputs a query from the frontend, it will take in the keyword and fetch the top five related campuswire posts to be displayed.

Crawler

Crawler.py is a python crawler that can crawl the most recent post data from campuswire.com, it uses selenium as the main method to run a Microsoft Edge Driver with function of logging in and switching courses. The posts and relevant data is stored in cw.txt for further data processing.

Preprocessor

Preprocess.py is implemented for initial data cleaning and processing after the posts are crawled, including tokenization, stemming and removal of stop words and redundant urls.

Ranker

Rank.py is where we index each processed post, construct the inverted index and compute the similarity score between posts and the query that user inputs. We used BM25 as the ranking function