UNIVERSITY OF NORTH TEXAS

COMPUTER SCIENCE DEPARTMENT

CSCE 5200 Information Retrieval and Web Search (Fall 2022)

Project Group Number – **25**

UNT – Search Engine

Team Members:

1) Sai Manoj Malepati
2) Sai Meghana Gushidi
3) Goutham Pallapothu
4) Abhay Arora
5) Sai Anjani Pinreddy
6) Sri Sai Charan Yerramsetty

Source Code

- .Zip Submitted in Canvas
- Git Repo: https://github.com/abhayarora23UNT/UNT_IRS_Project

Steps to Run the Project :

1. Download the zip file of the project.
2. Extract all files from the zip.
3. Open the command prompt and change the directory to the project file.
4. Run the command flask run.
    a. To change the default IP, run this command flask run --host=0.0.0.0
5. Open this URL in the browser: http://127.0.0.1:5000/

Technologies/Framework/Language

- Languages: Python, HTML
- Libraries: Beautiful soup, urllib, pandas, nltk, sklearn

Implementation Details

1) In the first phase, the application crawls the unt.edu website using python libraries: Beautiful soup, requests, urllib, and collections.
2) The crawler returned more than 2000 URLs.
3) After getting all the links, we parsed the HTML content, removed all tags, cleaned the text,
4) Then we removed all the characters from the text, saved the link, and cleaned the text in the corpus file.
5) After that, we query a search keyword using a vector space model, check document similarity using cosine similarity, and finally sort the result set.
6) The top 11 search results will be displayed on the UI
7) For routing, flask decorators are used.

Crawling via BeautifulSoup

```
soup = BeautifulSoup(response.text, "lxml")

    for link in soup.find_all('a'):

        anchor = link.attrs["href"] if "href" in link.attrs else ''



        if anchor.startswith('/'):

            local_link = base_url + anchor

            local_urls.add(local_link)

        elif strip_base in anchor:

            local_urls.add(anchor)

        elif not anchor.startswith('http'):

            local_link = path + anchor

            local_urls.add(local_link)

        else:

            foreign_urls.add(anchor)



        for i in local_urls:

            if not i in new_urls and not i in processed_urls:

                new_urls.append(i)
```

VectorSpace Model using TfidfVectorizer

```
for data in corpusData:
        tokenData = toGenerateTokens(data)
        LinkData = toReadStopWords(tokenData)
        LinkData = wordStemmer(LinkData)
        LinkData = ' '.join(LinkData)
        ModifiedCorpus.append(LinkData) # This contains data after preprocessing
stage

    vectorizerX = TfidfVectorizer(stop_words='english')
    documentVector = vectorizerX.fit_transform(ModifiedCorpus)
    dataFrame = pd.DataFrame(documentVector.toarray(),
                        columns=vectorizerX.get_feature_names_out())

    query = toGenerateTokens(query)
    query = toReadStopWords(query)
    queryData = []
    for word in wordStemmer(query):
        queryData.append(word)
    queryData = ' '.join(queryData)
    queryVector = vectorizerX.transform([queryData])
    #Using cosineSimilarities to find similarity between query and existing link
and content
    cosineSimilarityValues = cosine_similarity(documentVector,
queryVector).flatten()
    #Top 11 documents will be displayed on the top
    topScoredDocuments = cosineSimilarityValues.argsort()[:-12:-1]
```

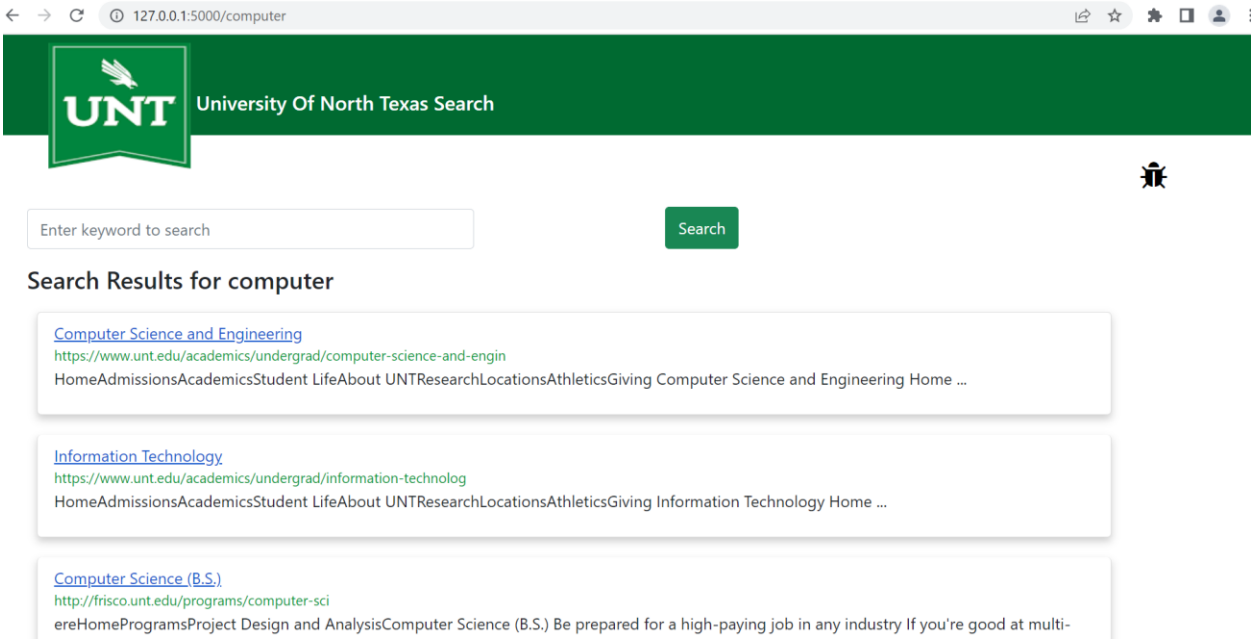
Outputs

- The home screen will show a search box.
- The user will enter some word in the search box and click the Search Button.
- The system will retrieve the top results 11 using the vector space model

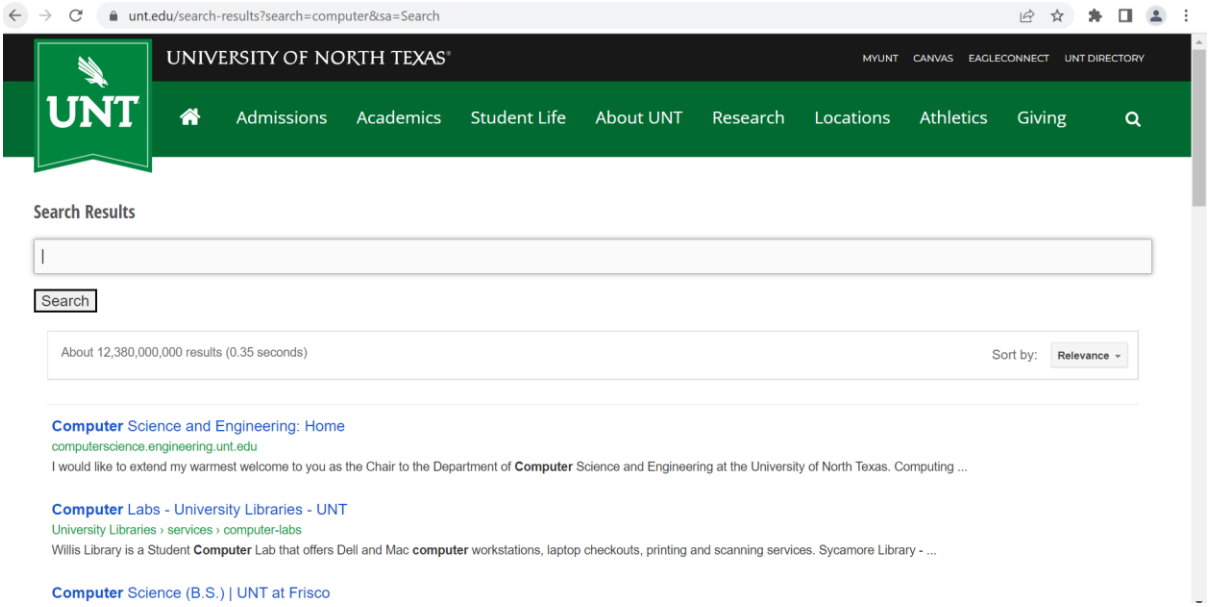
Sample Queries

Input 1: Computer

- Results from our application

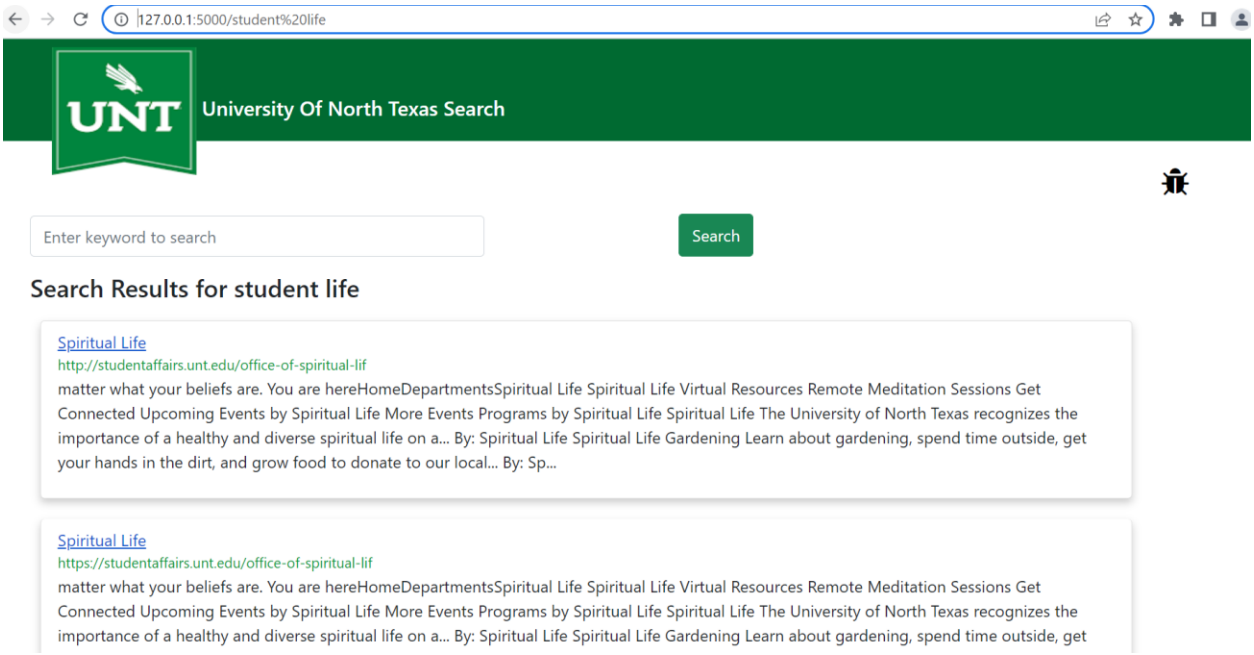


- Results from original Unt.edu :

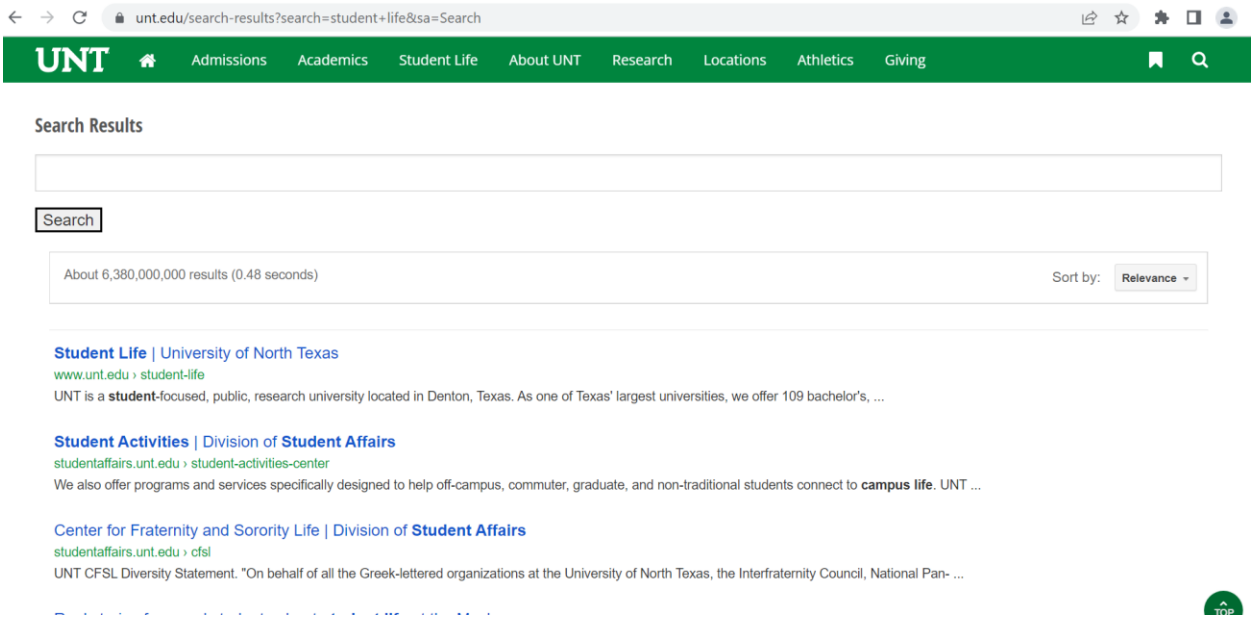


Input 2: Student Life

- Results from our application

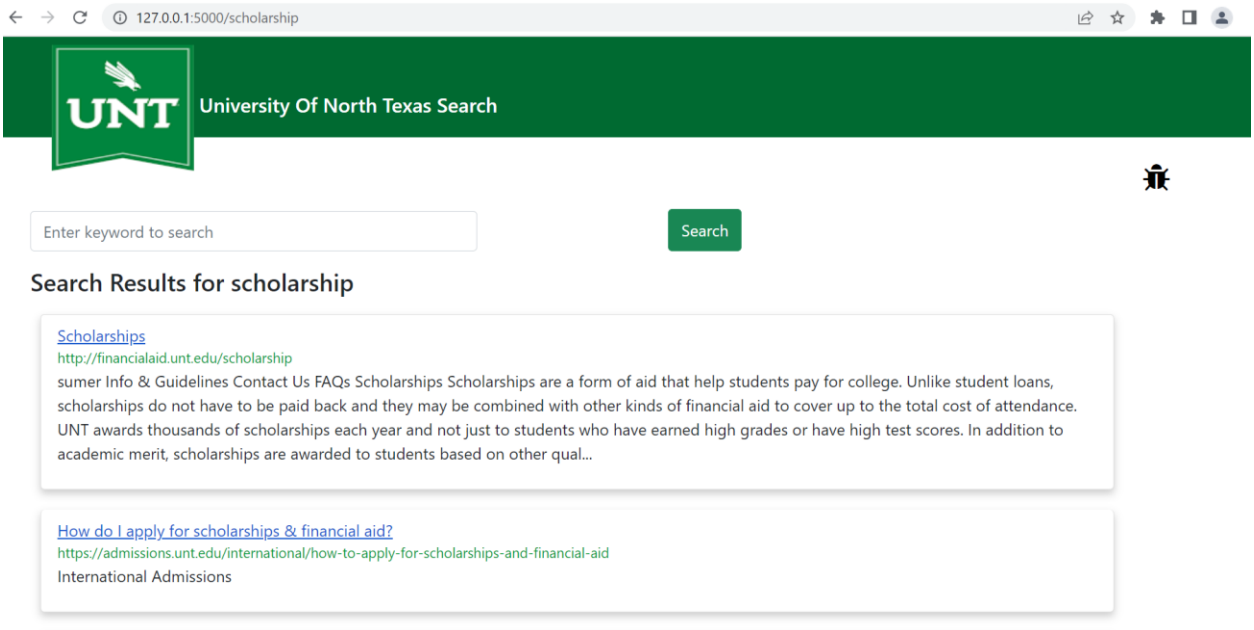


- Results from original Unt.edu :

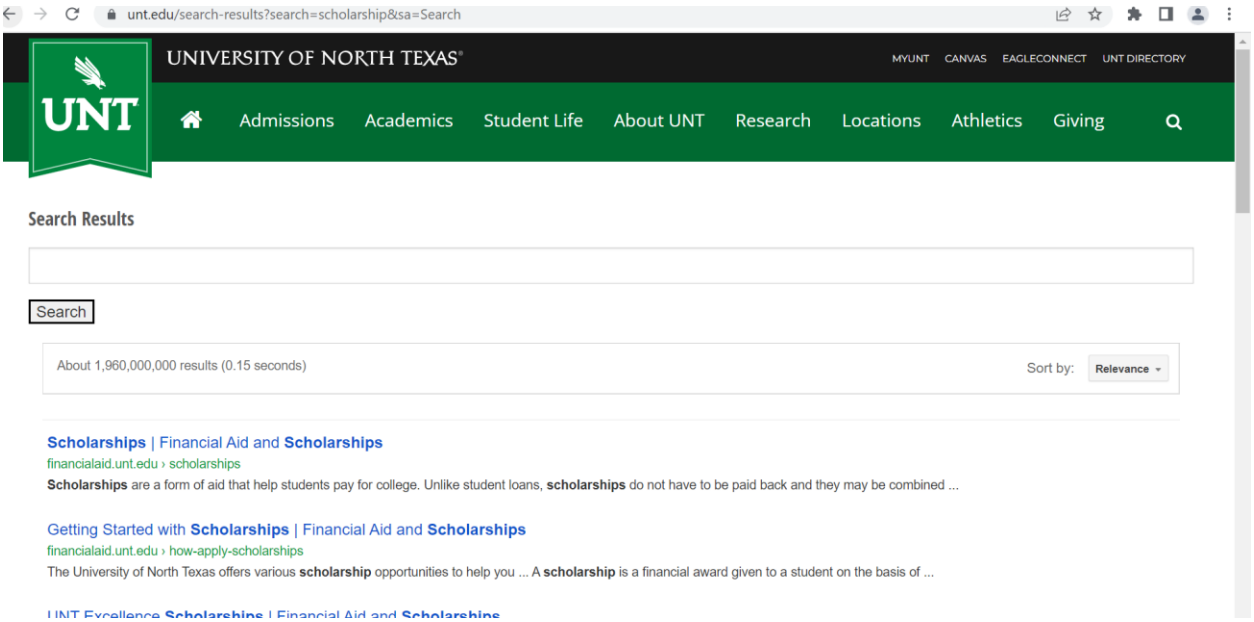


Input3: Scholarship

- Results from our application



- Results from original Unt.edu :



References

1. http://www.learningaboutelectronics.com/Articles/How-to-find-all-hyperlinks-on-a-web-page-in-Python-using-BeautifulSoup.php
2. https://www.kaggle.com/code/adepvenugopal/nlp-text-similarity-using-cosine-count-vectorizer
3. https://towardsdatascience.com/vector-space-models-48b42a15d86d
4. https://www.python.org/
5. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
6. https://www.w3schools.com/html/