Web Search Engine for UIC Domain

Snehal Tikare
Computer Science
IL, USA
stikar2@uic.edu

ABSTRACT

This document is report for final project under CS 582 Information Retrieval course. It describes the process involved in building a search engine, the methods that were used, drawbacks and future scope. In this project, I represent Scrapy – a python library for crawling and scraping the website,a vector space model to compute weighted document vectors using TF-IDF, cosine Similarity to retrieve the most similar document to query and Pagerank algorithm to rank the crawled links. The scraped documents are processed,indexed and saved.A graphical user interface to read the user input i.e. a query and display the ranked links using the above described methods. An intelligent component similar to Google's "People also search for" is implemented. using TF-IDF and similarity score.

CCS CONCEPTS

• **Information systems → Information retrieval**; **Retrieval models and ranking**; **Similarity measures**; **Top-k retrieval in databases**;

KEYWORDS

Crawling,Vector Space Model, Cosine Similarity,PageRank, Keyphrase extraction

1 INTRODUCTION

The project built using python3 and makes use of various inbuilt libraries. The process starts with crawling the websites under UIC domain starting with https://www.cs.uic.edu.Information is scraped from each crawled link. After crawling and scraping the information, next step is to preprocess the scraped information to extract only relevant data.Details of the process are explained in subsection below.A vector space model is implemented to represent the document as vector using TF-IDF.The page rank algorithm is used to rank all the crawled links.Cosine similarity is used return the top 10 links answering the query.

2 SOFTWARE COMPONENTS

This section explains in-depth pipeline of process followed to obtain the desired results

2.1 Web Crawler

The web crawler is used to crawl and scrape information from all the websites under UIC domain.It starts with cs.uic.edu as seed site and crawls all the websites following the links in a breadth first search manner. The information scraped from each website consist of title, content - body of html page, and outlink - all the links going out from a web page.The html page is scrapped off it's tags and only the data is saved to a JSON file. The library used is "Scrapy"- An open source and collaborative framework for extracting the data from websites.The reason Scrapy is chosen because it's provides in-built multi-threading feature which makes crawling faster and easier.
The following configurations are followed while crawling the links

- The links are canonicalized, removing same page links.
- Only links in the domain "uic.edu" are crawled and rest are discarded.
- Links leading to gif,pdf,zip are discarded as well.

The configurations used:

- `Number of links crawled`: 4000.
- `Crawling technique`: Breadth First Search(BFS).
- `Approach`: A multi-thread crawling

2.2 Preprocessing

The raw text retrieved from the website needs to be processed before building the model.The preprocessing steps involves the following steps.

- The first step is used to remove all the special characters,non alphabets,convert text to lower case and tokenize each document
- The second step is to remove the stopwords.To achieve this, NLTK's stopword list is used.
- The third step is to stem the tokens to their root forms. PorterStemmer is used to do this task.
- Finally, words with length less than 3 are removed from the document.

2.3 Model Building

This section explains various models and algorithms implemented.

2.3.1 Vector Space Model and Weighing Scheme

Once the documents are preprocessed, the next step is to vectorize them.We make use of Term Frequency and Inverse Document Frequency as the weights.
Term frequency is number of times the words occur in the document.It measures how frequently a term appears in a document.Term frequency is normalized and calculated using the formula.

$$tf_{ij} = \frac{f_{ij}}{max(f_{ij})} \tag{1}$$

Inverse Document frequency computes how important a term is.While calculating TF all words are considered equal. However,it is known that certain terms, such as "is", "of", and "that",appear more frequently but are not important. Thus to dampen this effect and increase the weights of rare terms we use IDF which calculated using the given formula:

$$idf_{ij} = log_e(\frac{N}{d_i}) \tag{2}$$

where N - Total number of documents,
d_i - Number of documents containing the term i.

. The collections of documents are represented in a vector space model by a term document matrix given by weights which is calculated as below

$$w_{ij} = tf_{i,d} \cdot idf_i \tag{3}$$

These weights are saved in a dictionary which is then to pickle file for external use.

2.3.2 PageRank

One of the intelligent component implemented is the pagerank algorithm. The Page Rank algorithm evaluates the quality and quantity of links to a webpage to determine a relative score of that page's importance and authority on a 0 to 10 scale.
Python's library - Networkx is used build to the graph.All the crawled links are fed to a directed graphs as nodes.A directed edge is added between two nodes A and B, if A directs to B. A score is given to every link crawled, the link with highest score is the most important one.This score is combined with Cosine Similarity to get the most relevant document.A weighted score with different weights associated with scores from cosine similarity and page rank is evaluated to get a final score and returned as a ranked list of links most relevant to the query.

2.3.3 Keyphrase extraction

A method I tried for implementing the "Also search for feature". Keyphrases provide a high level description of document.The idea beneath the implementation is that if a document retrieved is relevant then the top keyphrases associated with the document should be suggestive and similar to the user query.The initial step involves extracting keyphrases of all the documents using ngrams and page rank score.
Next,extract the top 5 keyphrases from each of the top 10 retrieved document relevant of the query.

The result of this implementation was not imperative and was not suggestive of the query.Possible explanation might be that the documents had more raw content not actually relevant to the context of the document.And page rank extracts phrases using textually similar neighbors.Thus,the keyphrases extracted weren't completed descriptive of the document and the query.

2.4 Query Processing, Retrieval and Ranking

The user enter the query that he/she wants to search.The query is preprocessed in the same way as document.The stop words are removed from the query, converted it to lower case and stemmed the remaining words and tokenized.Query vector is formed using the TF of query and IDF of document. Cosine similarity is computed between query and documents that contain atleast one of the query terms using the below formula.

$$CosSim(d_j, q) = \frac{\sum_{i=1}^{t} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}} \tag{4}$$

where w - weights are calculated using the formula in (3)

This score is added with page rank score to get a final score.The documents are then sorted in decreasing order.Top 10 links are returned to the user.

3 INTELLIGENT COMPONENT - ALSO SEARCH FOR

The intelligent component implemented is similar to Google's "People also search for".In addition to links provided to the user, the component suggests a set of additional queries similar in context to the input that the user might be interested in. We begin with the information intrinsic to the initially retrieved document relevant to query.To gain context about these documents, the words with highest TF-IDF score wrt the documents are considered. The process followed to extract keywords is follows:

- Top 10 words with highest TF-IDF from each of the top 10 relevant documents are extracted.
- For each of the token, sum the score of that token in every document.
- Next step is to rank these tokens based their scores.

- Finally, compute the similarity between the tokens and the query and return the tokens whose similarity is greater than 0.5 and tokens that are not in the query itself.

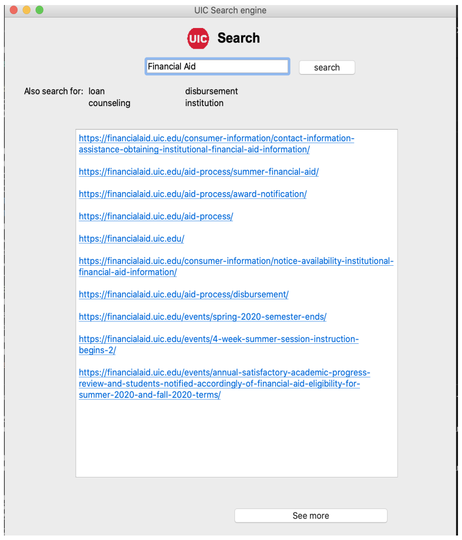
4 MAIN CHALLENGES

The first challenge faced was to understand the structure of webpages and how extract the relevant information.After careful observation and experimentation, the right tags were extracted to get the most out of the webpages.Also, an added advantage was that most of the pages under UIC domain have similar structure.Thus, a single blueprint for extracting information was sufficient to scrape all the websites.
Second challenge was tuning the hyperparameters of page rank like number of iteration, ϵ value.
Also, there were challenges while implementing the keyphrase extraction as explained in section 2.3.3

5 RESULTS AND EVALUATION

The result and analysis of five sample queries are as below:

(1) Results for query "Financial Aid"



Financial Aid

Also search for: loan disbursement counseling institution

https://financialaid.uic.edu/consumer-information/contact-information-assistance-obtaining-institutional-financial-aid-information/

https://financialaid.uic.edu/aid-process/summer-financial-aid/

https://financialaid.uic.edu/aid-process/award-notification/

https://financialaid.uic.edu/aid-process/

https://financialaid.uic.edu/

https://financialaid.uic.edu/consumer-information/notice-availability-institutional-financial-aid-information/

https://financialaid.uic.edu/aid-process/disbursement/

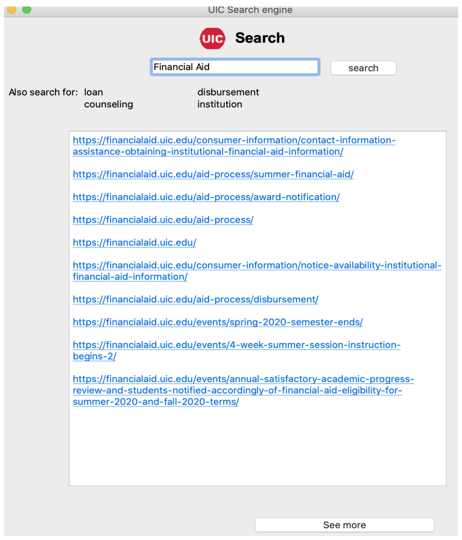
https://financialaid.uic.edu/events/spring-2020-semester-ends/

https://financialaid.uic.edu/events/4-week-summer-session-instruction-begins-2/

https://financialaid.uic.edu/events/annual-satisfactory-academic-progress-review-and-students-notified-accordingly-of-financial-aid-eligibility-for-summer-2020-and-fall-2020-terms/

Figure 1: Result of query for using both Cosine Similarity and Page Rank.

We see that the results for the query using a combined score of cosine similarity and page rank and that of only cosine similarity has no difference.Both result are relevant to the query.We see that the links returned have term "Financial" in it.And so does these web pages where the term is repeated a lot than in other documents.



Financial Aid

Also search for: loan disbursement counseling institution

https://financialaid.uic.edu/consumer-information/contact-information-assistance-obtaining-institutional-financial-aid-information/

https://financialaid.uic.edu/aid-process/summer-financial-aid/

https://financialaid.uic.edu/aid-process/award-notification/

https://financialaid.uic.edu/aid-process/

https://financialaid.uic.edu/

https://financialaid.uic.edu/consumer-information/notice-availability-institutional-financial-aid-information/

https://financialaid.uic.edu/aid-process/disbursement/

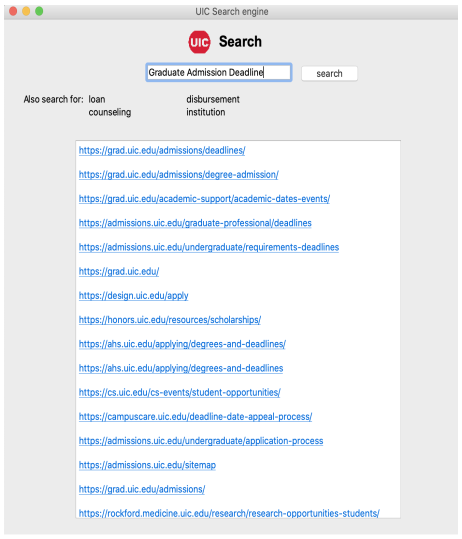
https://financialaid.uic.edu/events/spring-2020-semester-ends/

https://financialaid.uic.edu/events/4-week-summer-session-instruction-begins-2/

https://financialaid.uic.edu/events/annual-satisfactory-academic-progress-review-and-students-notified-accordingly-of-financial-aid-eligibility-for-summer-2020-and-fall-2020-terms/

Figure 2: Result of query for using only Cosine Similarity.

(2) Results for query "Graduate Admission Deadline"



Graduate Admission Deadline

Also search for: loan disbursement counseling institution

https://grad.uic.edu/admissions/deadlines/

https://grad.uic.edu/admissions/degree-admission/

https://grad.uic.edu/academic-support/academic-dates-events/

https://admissions.uic.edu/graduate-professional/deadlines

https://admissions.uic.edu/undergraduate/requirements-deadlines

https://grad.uic.edu/

https://design.uic.edu/apply

https://honors.uic.edu/resources/scholarships/

https://ahs.uic.edu/applying/degrees-and-deadlines/

https://ahs.uic.edu/applying/degrees-and-deadlines

https://cs.uic.edu/cs-events/student-opportunities/

https://campuscare.uic.edu/deadline-date-appeal-process/

https://admissions.uic.edu/undergraduate/application-process

https://admissions.uic.edu/sitemap

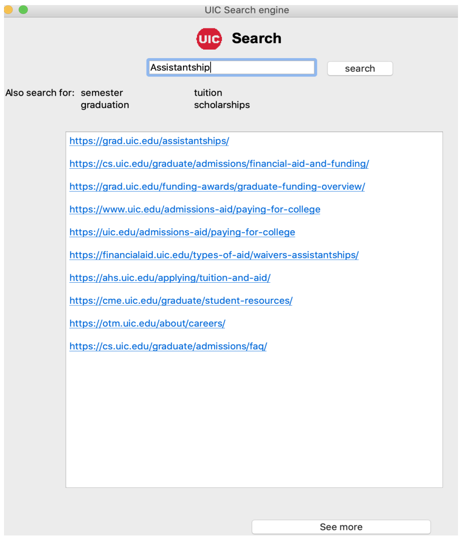
https://grad.uic.edu/admissions/

https://rockford.medicine.uic.edu/research/research-opportunities-students/

Figure 3: Result of query for using TF-IDF for suggesting additional queries.

The two results for the query shows the difference in the suggestions given to the user. We see the results obtained from keyphrases are not quite relevant to the query and possible reason is explained in section 2.3.3 whereas the suggestions obtained from considering TF-IDF are quite synonymous and similar in context with the query.The reason for this is same as that explained in section 3.

(3) Results for query "Assistantship"

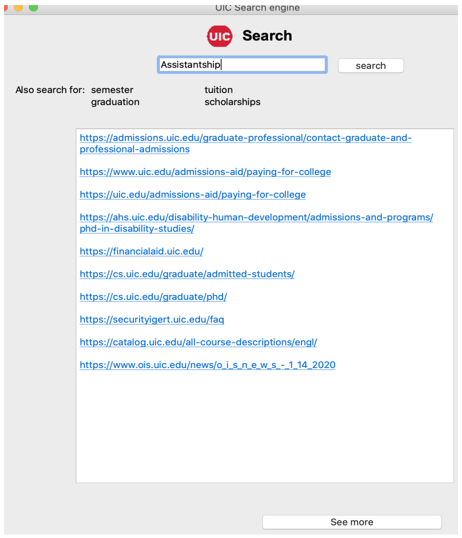


Assistantship

Also search for: semester tuition graduation scholarships

https://grad.uic.edu/assistantships/
https://cs.uic.edu/graduate/admissions/financial-aid-and-funding/
https://grad.uic.edu/funding-awards/graduate-funding-overview/
https://www.uic.edu/admissions-aid/paying-for-college
https://uic.edu/admissions-aid/paying-for-college
https://financialaid.uic.edu/types-of-aid/waivers-assistantships/
https://ahs.uic.edu/applying/tuition-and-aid/
https://cme.uic.edu/graduate/student-resources/
https://otm.uic.edu/about/careers/
https://cs.uic.edu/graduate/admissions/faq/

Figure 4: Result of query for using both cosine similarity and page rank.



Assistantship

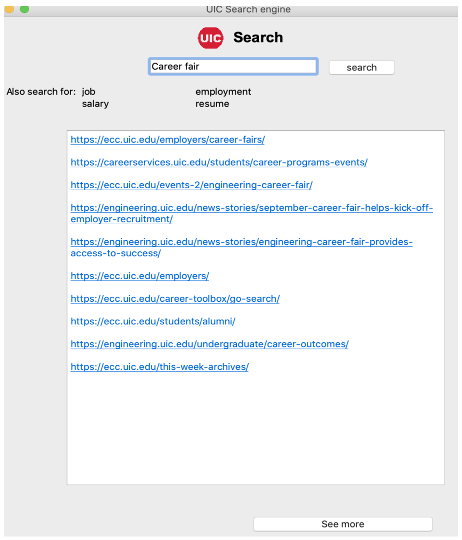
Also search for: semester tuition graduation scholarships

https://admissions.uic.edu/graduate-professional/contact-graduate-and-professional-admissions
https://www.uic.edu/admissions-aid/paying-for-college
https://uic.edu/admissions-aid/paying-for-college
https://ahs.uic.edu/disability-human-development/admissions-and-programs/phd-in-disability-studies/
https://financialaid.uic.edu/
https://cs.uic.edu/graduate/admitted-students/
https://cs.uic.edu/graduate/phd/
https://securityigert.uic.edu/faq
https://catalog.uic.edu/all-course-descriptions/engl/
https://www.ois.uic.edu/news/o_i_s_n_e_w_s_-_1_14_2020

Figure 5: Result of query for using only page rank.

The results for query "Assistantship" in fig 4 shows the links retrieved using a addition of cosine similarity and page rank score to decide the final ranking of the webpages.These pages have the term "Assistantship" frequently appear in their website thus giving them higher weightage. Figure 5 shows results obtained by using only page rank for deciding the order of the documents to be returned to the user.The results returned consists more of authorative links that are in general have higher importance and may not be completely relevant to the query or may not have the exact context as those returned by using both the techniques. Thus, a combination of scores is better than using just the page rank.

(4) Results for query "Career Fair"

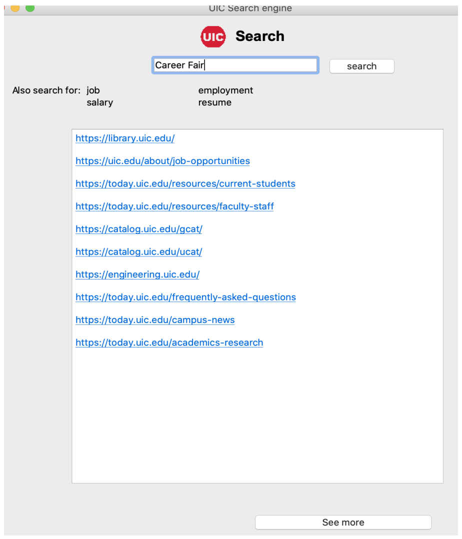


Career fair

Also search for: job employment salary resume

https://ecc.uic.edu/employers/career-fairs/
https://careerservices.uic.edu/students/career-programs-events/
https://ecc.uic.edu/events-2/engineering-career-fair/
https://engineering.uic.edu/news-stories/september-career-fair-helps-kick-off-employer-recruitment/
https://engineering.uic.edu/news-stories/engineering-career-fair-provides-access-to-success/
https://ecc.uic.edu/employers/
https://ecc.uic.edu/career-toolbox/go-search/
https://ecc.uic.edu/students/alumni/
https://engineering.uic.edu/undergraduate/career-outcomes/
https://ecc.uic.edu/this-week-archives/

Figure 6: Result of query for using only cosine similarity.



Career Fair

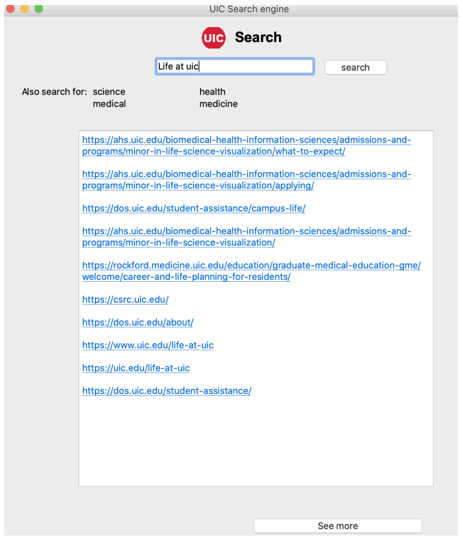
Also search for: job employment salary resume

https://library.uic.edu/
https://uic.edu/about/job-opportunities
https://today.uic.edu/resources/current-students
https://today.uic.edu/resources/faculty-staff
https://catalog.uic.edu/gcat/
https://catalog.uic.edu/ucat/
https://engineering.uic.edu/
https://today.uic.edu/frequently-asked-questions
https://today.uic.edu/campus-news
https://today.uic.edu/academics-research

Figure 7: Result of query for using only page rank.

The next comparison we are looking at is using either cosine similarity or page rank.For the query "Career fair" the links returned using Cosine similarity are more informative and have "Career Fair" as a term mentioned in their document where the links returned by Cosine Similarity are ranked based on their importance with respect to all the links.Mainly, page rank returns authoritative links which may or may not be useful.

(5) Results for query "Life at uic"



Life at uic

Also search for: science medical health medicine

https://ahs.uic.edu/biomedical-health-information-sciences/admissions-and-programs/minor-in-life-science-visualization/what-to-expect/

https://ahs.uic.edu/biomedical-health-information-sciences/admissions-and-programs/minor-in-life-science-visualization/applying/

https://dos.uic.edu/student-assistance/campus-life/

https://ahs.uic.edu/biomedical-health-information-sciences/admissions-and-programs/minor-in-life-science-visualization/

https://rockford.medicine.uic.edu/education/graduate-medical-education-gme/welcome/career-and-life-planning-for-residents/

https://csrc.uic.edu/

https://dos.uic.edu/about/

https://www.uic.edu/life-at-uic

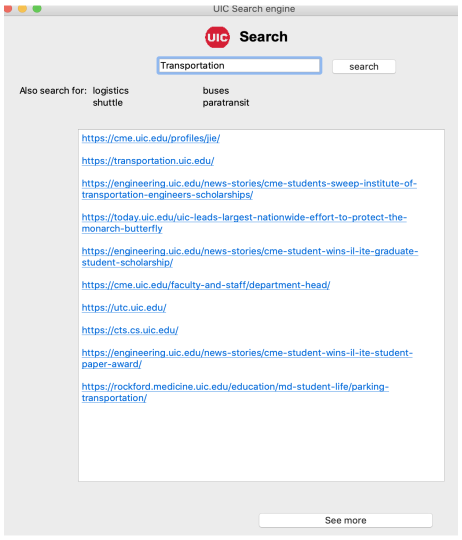
https://uic.edu/life-at-uic

https://dos.uic.edu/student-assistance/

Figure 8: Result of query for using both the measures.

The result of query "Life at UIC" shows the relevancy of the retrieval model.We see that model retrieves document that consists these terms.This shows the problem of ambiguity that comes with short queries. We would like to see more of contextual results which shows documents related to student life.Query expansion that takes into account feedback from user might help curb this problem.

(6) Results for query "Transportation"



Transportation

Also search for: logistics shuttle buses paratransit

https://cme.uic.edu/profiles/jie/

https://transportation.uic.edu/

https://engineering.uic.edu/news-stories/cme-students-sweep-institute-of-transportation-engineers-scholarships/

https://today.uic.edu/uic-leads-largest-nationwide-effort-to-protect-the-monarch-butterfly

https://engineering.uic.edu/news-stories/cme-student-wins-il-ite-graduate-student-scholarship/

https://cme.uic.edu/faculty-and-staff/department-head/

https://utc.uic.edu/

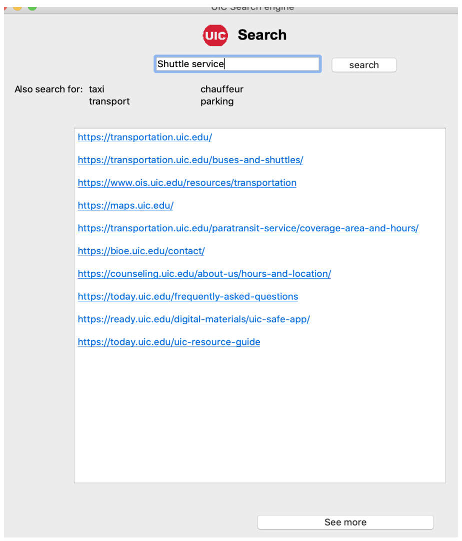
https://cts.cs.uic.edu/

https://engineering.uic.edu/news-stories/cme-student-wins-il-ite-student-paper-award/

https://rockford.medicine.uic.edu/education/md-student-life/parking-transportation/

Figure 9: Result of query for using both the measures.

(7) Results for query "Shuttle service"



Shuttle service

Also search for: taxi transport chauffeur parking

https://transportation.uic.edu/

https://transportation.uic.edu/buses-and-shuttles/

https://www.ois.uic.edu/resources/transportation

https://maps.uic.edu/

https://transportation.uic.edu/paratransit-service/coverage-area-and-hours/

https://bioe.uic.edu/contact/

https://counseling.uic.edu/about-us/hours-and-location/

https://today.uic.edu/frequently-asked-questions

https://ready.uic.edu/digital-materials/uic-safe-app/

https://today.uic.edu/uic-resource-guide

Figure 10: Result of query for using both the measures.

Figures 9 and 10 shows difference between results for two similar queries.The results returned for transportation consist of links that have descriptions for research paper related to transportation.Thus, there is difference in the context of the result.The results returned for query "Shuttle" seem more appropriate to what user wants.This shows how keywords affects the results.

6 RELATED WORK

There are various other methods for implementing the methodologies described here to enhance the retrieval process.Query expansion using Rocchio algorithm and Pseudo relevance feedback is one such method that expands the user's query intrinsically using the initially retrieved relevant documents.Using inner product to calculate the similarity instead of Cosine Similarity. Probabilistic language model in-place of page rank algorithm.

7 FUTURE SCOPE AND CONCLUSIONS

The results were quite promising.The links were relevant to the user query. The future scope would be to implement a auto suggestion and auto correction module.Also, use a trained model integrating neural network to achieve more accurate result. And even would like to extend to domains other than UIC.Use of spacy to find similarity has slowed the retrieval, I would like to improvise the perfromance of the GUI.

REFERENCES

[1] Min Song ,Il-Yeol Song, Robert B. Allen and Zoran Obradovic. *Keyphrase Extraction-Based Query Expansion in Digital Libraries*.

[2] Stack Over flow website `https://stackoverflow.com/questions/14940569/combining - tf-idf-cosine-similaritywithpagerank`