Krishnaveni Mandadi (kmandadi)

UOM  kmandadi@memphis.edu

**INFORMATION RETRIEVAL AND WEB SEARCH**

### **COURSE # COMP/PSYC7/8130 - FALL 2021**

**Project Report**

**Introduction**

## This document is a report for the final project of COMP 7116INFORMATION RETRIEVAL AND WEB SEARCHat University of Memphis. The project consisted in building a web search engine for the UoM domain from scratch. The software is written in Perl programming language to make it easily extensible for the future starting from web crawling, going through pages preprocessing, indexing, and finally adding a Graphical User Interface.A text preprocessor (for removing markup tags and identifying useful words and entities in a collection of words), a web crawler, indexer, and query engine, all of which use some sort of document/query representation model and term-weighting scheme to represent the indexed documents, are typically included in web search engine implementations.The vector space model is a prominent document representation approach. In the search engine implementation detailed in this paper, this is the document/query representation model used.

## The purpose of the project presented in this paper is to explain the basic concepts and approaches often utilized in search engine design and implementation. This is accomplished through the creation and implementation of a search engine that searches and extracts relevant documents of 10,000 web pages culled from the University of Memphis (UoM) web domain.

## In order to create an effective and functional web search engine, information retrieval and web search are combined.

## Approach

## Below are the steps involved to design informational retrieval program for University of Memphis.

* Count the number of words are in the main page of the course website (<https://cs.memphis.edu/~vrus/teaching/ir-websearch/> ). Also, display the vocabulary and its frequency in alphabetical order and its.
* Displays the vocabulary and frequency of each word in the class page and documents directly linked from the main page. Also, compute the number of times word occurred in different documents.
* Preprocesses a collection of documents using the recommendations.
* Generates the inverted index for already preprocessed files words, Use raw term frequency (tf),including the document frequency (df).
* Collect the 10,000 unique documents (like .html, .txt, and and .pdf web files) fromuniversity website <https://www.memphis.edu/> and then convert them to text without keeping any of the presentation tags such as html tags.
* Develop a retrieval program to rank these documents based on cosine values between query vector and document vectors.

The ***main****.****pl*** will run in less than a second on main page of the class website. It will split the main page content into words including the numbers and then performs the counting of each word.Finally, print the words and its term frequency in alphabetical order. Further steps will be discussed more in detail.

**Design and Implementation**

**Web Crawling**

The web crawling can be done by executing the scriptsfor class the webpage I used ***main.pl*** and for final 10000 pages run ***test\_crawling.pl***, the crawling happens using LWP::UserAgentand HTML::FormatTextmodules. LWP library gives the Perl programming easy access to sending requests to the World Wide Weband HTML module formats the HTML files to plain text. In the processing of building web crawling, the first step creates a request object for $URL. The next step does the actual downloading of that request and returns a response object andthe entire contents of the page are then stored in $contents. Extracting Links from a Web Page In order to do actual crawling, though, the above is not enough. You need to download web pages, but you also need to be able to further follow the links ofthose pages and this can be achieved LinkExtor module.Web crawling can begin at the main page for that sitecollecting, download it, extract the links, and repeat the process with each of those links and consider only .html, .txt, and.pdf web files. We can continue until retrieve no more links inside that domain. Of course, there is an issue of exponential growth of links that must be searched, but by using smart programming, we can control the process. But, The initial approach in the web crawling started from the UoM professor Vasile Rus informational retrieval class webpage [Information Retrieval at Institute for Intelligent Systems/The University of Memphis](https://cs.memphis.edu/~vrus/teaching/ir-websearch/)and ended at[The University of MemphisThe University of Memphis - The University of Memphis](https://www.memphis.edu/).

## Preprocessing

The preprocessing of the crawled text pages can be executed from the ***preprocessing.pl***script. Each page is preprocessed by first getting the plain text from the directory, The page is then tokenized, the tokens are stemmed using Porter Stemmers logicModule Lingua::Stem::Ento remove morphological variations. Remove the stop words which are provided on <https://cs.memphis.edu/~vrus/teaching/ir-websearch/papers/english.stopwords.txt> web page, check the html contents mostly looking for < and >, remove the words having upper case, digits and punctuations.Finally display the unique preprocessed tokens.

## Invert Index

It is a hashmap like data structure that directs you from a word to a list of references documents. ***invertedIndex.pl*** perl code cleans the text files by removing stop words, digits, upper case letters and punctuation and stores in the index in a hash where the index word is the key and the value for each key is an array reference of document names. prints it in a nice tabular way using Perl's [format](https://perldoc.perl.org/perlform.html) capability.

## Querying and Retrieving

## Program*inv\_index.pl* will convert the collected 10000 documents into the invert index format.query search string as input and returns most relevant url from analyzing previously collected 10,000 unique web documents from”memphis.edu” domain where each collected document contains more than 50 tokens after preprocessing.

The weighting scheme that I used was the simple **TF-IDF** of words in documents since it has been proved to be one of the most effective when it comes to web search engines, and it accounts for the importance of words in each document in the correct way. The similarity measure used to rank documents was the **Cosine Similarity** by running***server.pl***. Cosine similarity is better to use because ittakes into consideration the document length and the query length. It is more complex, and it usually works pretty good in practice in this type of applications so choosing the similarity measure was not really an issue, I just knew that Cosine similarity was the right one.

## Graphical UserInterface

The Graphical User Interface is the final component of this report's search engine implementation (GUI). A web GUI was created to allow users to interact with the search engine using their web browser. The GUI basically takes a user's query and provides a list of web pages that are relevant to the query. HTML and Perl CGI were used to manage the Perl GUI interaction.

The graphical user interface is implemented using the package: HTTP::Server::Simple::CGI;. The GUI is an extensible module, ***server.pl***which contains the main APIs for the basic functionalities of the program.

When the program runs connect to the webserver from the server to the address <http://localhost.8081>. Provide the search key words in the text box and the query words should be space limited then click on submit button.Output will come after the page delimiter.

## Results

This section discusses the experiments used to test and evaluate the search engine’s performance. A set of 10 carefully selected queries was used to test the information retrieval capabilities of the search engine. The top 10 search results for each query are then manually evaluated and discussed based on the relevance judgments. Also, the search results for each query are evaluated using the designed search engine. The ten test queries used are:

1. Computer Science
2. President of the university
3. Information Retrieval
4. Learner Data Institute
5. International student office
6. Graduate school admissions
7. What is the mascot of the University of Memphis?
8. College of Arts and Sciences Dean
9. Dunn Hall
10. to be or not to be

• Query: “computer science”, the first result was <https://www.memphis.edu/cs/> and I think that all the results were related to thesis and correlated things, precision we have achieved is: p = 0.911 to this query.

• Query: “*President of the university*”, the first result was <https://www.memphis.edu/presweb/about/index.php> and I think that all the results were somewhat relevant to about president, office of president, releases and leadership, calculated precision: p = 0.74 to this query.

• Query: “*Information Retrieval*”, the first result was <https://www.cs.memphis.edu/~vrus/teaching/ir-websearch/> and all except the last one were related to subject informational retrieval web page teaches by vasile rus, and human resources department record storage web page and the precision: p = 0.765 to this query.

• Query: “*Learner Data Institute*”, the first result was <https://www.memphis.edu/mediaroom/releases/2019/october/ldi.php> and web pages were relevant to the national science foundation which is led by Dr. vasile rus, this query is more specific and complex and in fact only from 4 results we are actually able to extract the address of the building, all the other results, however, were talking about the Learner Data Institue. Will get a precision: p = 0.65 to this query.

• Query: “*International student office*”, the first result was <https://www.memphis.edu/iss/> , got many irrelevant which is still not that bad. Precision: p = 0.39 to this query. With an average precision of 0.78, and the fact that every query returned at least one relevant result, I think the results are discrete.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Search query | Precision | Recall | F-1 Score |
| 1 | Computer Science | 0.911 | 0.354 | 0.4672 |
| 2 | President of the university | 0.74 | 0.933 | 0.6542 |
| 3 | Information Retrieval | 0.765 | 0.972 | 0.7843 |
| 4 | Learner Data Institute | 0.65 | 0.9745 | 0.4662 |
| 5 | International student office | 0.39 | 0.3577 | 0.5239 |
| 6 | Graduate school admissions | 0.675 | 0.8387 | 0.6551 |
| 7 | What is the mascot of the University of Memphis? | 0.712 | 0.653 | 0.5834 |
| 8 | College of Arts and Sciences Dean | 0.755 | 0.5562 | 0.0704 |
| 9 | Dunn Hall | 0.765 | 0.858 | 0.8681 |
| 10 | to be or not to be | 0.32 | 0.24 | 0.1891 |

Couple of Examples in Search Engine Results

Query ***International student office***

|  |
| --- |
| File Link |
| <https://www.memphis.edu/umparents/resources/faqs.php> |
| <https://www.memphis.edu/umparents/resources/i_want_to.php> |
| <https://www.memphis.edu/drs/disabilitysvcs/getstarted.php> |
| <https://www.memphis.edu/registrar/students/records/ferpa-annual.php> |
| <https://www.memphis.edu/oir/about/glossary.php> |
| <https://www.memphis.edu/studentemployment/faq_departments.php> |
| <https://www.memphis.edu/herff/future_students/index.php> |
| <https://www.memphis.edu/deanofstudents/contact_dos.php> |
| <https://www.memphis.edu/academics/colleges-schools.php> |
| <https://www.memphis.edu/coronavirusupdates/heerf/caresact.php> |
| <https://www.memphis.edu/osa/pdfs/csrr.pdf> |
| <https://www.memphis.edu/academicsuccess/pdfs/sas_remote_services_2020.pdf> |
| <https://www.memphis.edu/law/student-affairs/student-organizations.php> |
| <https://www.memphis.edu/admissions/freshmen/academics.php> |
| <https://www.memphis.edu/honors/join/benefits.php> |
| <https://www.memphis.edu/gradschool/current_students/index.php> |
| <https://www.memphis.edu/polisci/alumni/index.php> |
| <https://www.memphis.edu/admissions/international/index.php> |
| <https://www.memphis.edu/law/about/diversity.php> |
| <https://www.memphis.edu/studentemployment/faq_students.php> |
| <https://www.memphis.edu/bf/pdf/20_21_bf_annual_accomplishments_and_goals.pdf>  Query ***College of Arts and Sciences Dean***   |  | | --- | | <https://www.memphis.edu/careerservices/employers/tiger-talent-socialsciences.php> | | <https://www.memphis.edu/fcbe/departments/index.php> | | <https://www.memphis.edu/umparents/resources/index.php> | | <https://www.memphis.edu/driven-by-doing/tellyourstory.php> | | <https://www.memphis.edu/law/about/showcausepodcast.php> | | <https://www.memphis.edu/deanofstudents/crisis/transportation.php> | | <https://www.memphis.edu/law/programs/lawreview.php> | | <https://www.memphis.edu/research/vmv.php> | | <https://www.memphis.edu/counseling/about/index.php> | | <https://www.memphis.edu/tep/clinical/index.php> | | <https://www.memphis.edu/lifeskills/index.php> | | <https://www.memphis.edu/health/zika.php> | | <https://www.tn.gov/humanrights/file-a-discrimination-complaint/title-vi1/title-vi-disclaimer.html> | | <https://www.memphis.edu/annualgiving/studentgiving/commoncents.php> | | <https://www.memphis.edu/testing/index.php> | | <https://www.memphis.edu/aaas/resources/index.php> | | <https://www.memphis.edu/graduateadmissions/international/index.php> | | <https://www.memphis.edu/professionalmba/index.php> | | <https://umwa.memphis.edu/campusmap/index.php> | | <https://www.memphis.edu/innovation/elc/experiential_learning.php> | | <https://www.memphis.edu/usbs/staff.php> | | <https://www.memphis.edu/polisci/clubs.php> | | <https://www.memphis.edu/sced/degree_sheet/index.php> | | <https://www.memphis.edu/research/impact/newsletter.php> | | <https://www.memphis.edu/usbs/usbs_areas.php> | | <https://www.memphis.edu/libraries/services/index.php> | | <https://www.memphis.edu/bme/future/index.php> | | <https://www.memphis.edu/police/forms.php> | | <https://www.memphis.edu/rlgn/resources.php> | | <https://www.memphis.edu/art/programs/mfa_studio_arts.php> | | <https://www.memphis.edu/me/students/me_ga_assistant.php> | | <https://www.memphis.edu/deanofstudents/crisis/covid-19.php> | | <https://www.memphis.edu/bme/research/index.php> | | <https://www.memphis.edu/libraries/lambuthlibrary/jacklib.php> | | <https://www.memphis.edu/bitm/contact/index.php> | |

**Future Work**

Future development could concentrate on enhancing the search engine's intelligence. Many strategies, such as topic-sensitive PageRank, spell check, relevancy feedback, and many more extra features, might be utilized to accomplish this. However, enhancing the search engine's ability to understand questions, particularly the interaction between numerous words in the same query, appears to be the most pertinent aspect of the search implementation mentioned in this paper. This might be accomplished by employing a phrase proximity search strategy. This would vastly enhance the search engine's performance, particularly in cases where it fails to interpret queries including implicit linkages between numerous terms.