Project 1: Search Engine

# Text Processing

## Stop words

The stop word list we chose for this project was from the Ranks.nl site used in the homework, and we decided to use the: Default English Stopword list, Long Stopword list, and the MySQL Stopword list. Additionally, we took the tokens that were found in 2000 wiki documents out of the top 10,000 documents in size, which increased our stopwords by around 220 words.

* <https://www.ranks.nl/stopwords> **Total words: 758**
  + Default Stopword list - Unique words: 174
  + Long Stopword list - Unique words: 527
  + MySql Stopword list - Unique words: 57
* Top 10,000 Wiki Docs - **Total words: 221**

**Combined: 979**

## Tokenizer / Lemmatizer

The tokenizer and lemmatizer we used for this project is Stanford’s Core NLP. Originally, we had a discussion because the processing time of SimpleCoreNLP was far too slow to be considered a viable option to use in this project. After a bit of research, we discovered that using the pipeline severely reduced the processing time and provided a lot of powerful tools to make processing a bit easier to manage.

* <https://stanfordnlp.github.io/CoreNLP/>

## Modifications

In addition to using Stanford’s CoreNLP and the stopword lists from Ranks.nl, we decided to use an English dictionary of terms to compare against our tokens to remove many of the unusual terms our index was saving from the documents. These terms often seemed to be in reference to the images stored in the documents, and therefore we were able to remove many of these terms with the dictionary. Implementation was such that the stopwords were removed first, then we checked the term against the dictionary to ensure it was a valid English word, if not we discarded the term.

The dictionary we chose to use was Keith Vertanen’s second biggest list of English words that he took from the intersection of 12 different popular word lists. Keith is an Associate Professor for the Department of Computer Science at Michigan Tech and put together this dictionary of terms for the use in his research Augmentative and Alternative Communication devices. After looking through this dictionary we decided that it should work well for our needs since his needs for speech and language processing seem to coincide well with this project.

* <https://www.keithv.com/software/wlist/>
  + Popular Corpus’s Used
    - Google corpus (top 400k words)
    - Wikipedia (top 400k words)
    - Gigaword newswire corpus (top 400k words)
    - Moby word list (335K words)
  + Total Size of Dictionary
    - **586,632 words**

## Final Statistics

* **Tokens found Preprocessing: 312,608,683**
* **Intersection Size: 61,443,567**
* **Total Tokens after Preprocessing: 61,443,567**
* **Index Size of Unique Tokens: 377,594**

# Index Creation

Creating an index was the more time-consuming part of this project for us, both in terms of actual processing time and decision making. Initially our idea was to use a database to store our index with three separate tables, which I will outline briefly as it pertains to the final implementation of our project. The first script was Documents which would store the document number, the highest frequency term (for easy use in our TF-IDF calculation), the title, and the filepath of the document. Second was our Tokens script, which would save unique tokens to a primary key and store a hash value for that token. Last was our Intersections script which would have a foreign key to a token, the document id, and a frequency. After a bit of testing and a lot of patience, we realized that we couldn’t really make the database work in a timely manner without doing much more research on optimization.

Instead of scrapping everything we had, we decided to approach the problem trying to use what we had already implemented; a multi-threaded approach that split the documents into three separate script files of much easier -to-process information. Using this approach, we were able to keep much of our implementation and extend it to maps and sets instead of the database. After doing this, we still ran into some issues with extending this to fit our needs but loading the index after its creation was severely reduced. This section is an area that we identify in our discussion as a focus for improvement on the next project.

## Final Statistics for Index Creation

* Index Creation Time: **2 hrs. 37 min**
* Index / AOL Log Load Time: **3 min 58 sec**
* Average Doc/Suggestion Retrieval **~15ms**

# Search Engine

The final piece of our system was the search engine, which points to the directories of our index script files and the AOL query logs provided that utilizes a command line interface to interact with the user. While it’s a bit primitive in this stage, it has functionality to return all the necessary requirements for this project and responds with relevant documents very quickly upon querying.

## Relevance Ranking

With some foresight in our design implementation for parsing and storing the document, our relevance ranking for documents was straightforward. We already saved the term-frequency as well as the term with the highest frequency in each document in our data structures, so calculation of the TF-IDF is just the process of pulling this information related to the query. To improve efficiency even more, as we add ranked documents to our data structure, we limit this to only the top five ranked documents at a time. This allows inserts and removal of the lowest ranked document from the ranked data to be very quick, which can be seen in the speed of our query results.

## Snippets

For snippets, we followed a very similar design premise as the rest of the project. First to improve efficiency we only generate the snippets for the top five documents after we have discovered this information. To add to the efficiency further, we take a threaded approach again to pass the speed to providing the user with information quickly. When it comes to the cosine similarity, we treated the document in question as the corpus and each individual sentence as the document to get the best ranking possible for the sentences. Finally, for better readability for the user, we ordered the top two ranked sentences by the order in which they were seen in the document instead of the overall cosine similarity score.

## Query Suggestions

Unlike the index, query suggestions are generated and loaded into memory at the same time we take our prebuilt index and load it into memory as it takes less than 30 seconds to completely parse and load into memory. To create the suggestions, we created both Query objects and Session objects, with the former having a variable linking it to the specific Session it belongs to as well as the position in the Session it was located. The Session would have an array list of all the unique queries (to that specific session) performed. Finally, we stored a map of each individual term (space separated) in the queries, that linked to all the full queries that contained that term. This allowed for us to quickly search for the individual terms of a user query, gather all queries in which that term existed, and cross reference it against the list for the rest of the terms in the user’s query. The design allowed us to quickly get all the sessions that had our query + n terms (n>=1), get the frequency in which that specific query occurs in the logs, and calculate the score to rank the suggestions. When storing the list to return the suggestions to the user, we employ a similar tactic of only storing the top five most relevant search terms with ties being sorted by the suggestions that are closer in length to the original query plus one term. Last bit of information related to suggestions is that we implemented a way for the user to quickly pick a suggestion as the next query by entering in its rank.

# Discussion

## Result Analysis

**Query 1: personal assistant**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Document ID** | **Snippet** | **Score** |
| 1 | 169490 | "EO Personal Communicator The EO is an early commercial tablet computer that was created by Eo, Inc. (later acquired by AT&T Corporation), and released in April 1993. Officially named the AT&T EO Personal Communicator, it was similar to a large personal digital assistant with wireless communications, and competed against the Apple Newton. | 7.438 |
| 2 | 12667 | "Clova (virtual assistant) Clova is an intelligent personal assistant for Android and iOS operating systems developed by Naver Corporation and Line Corporation (a subsidiary of Naver). Clova, short for ""cloud virtual assistant"", was officially introduced in May 2017. | 7.317 |
| 3 | 119308 | "Personal organizer A personal organizer, day planner, personal analog assistant, personal planner, year planner, or agenda (from Latin ""agenda"" - things to do), is a small book or binder that is designed to be portable. By the end of the 20th century, paper-and-binder personal organizers started to be replaced by electronic devices such as personal digital assistants (PDAs), personal information manager software, and online organizers. | 7.276 |
| **Ranking** | **Candidate Suggestion** | **Score** |
| 1 | Personal assistant jobs | 1 |
| 2 | Personal assistant websites | 1 |
| 3 | Personal assistant job sites in us | 1 |
| **Ranking** | **Wikipedia Document Title** | |
| 1 | Personal assistant | |
| 2 | Virtual assistant | |
| 3 | Personal digital assistant | |

**Query 2: zodiac**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Document ID** | **Snippet** | **Score** |
| 1 | 334927 | Graysmith was a cartoonist for the ""San Francisco Chronicle"" and later also wrote ""Zodiac Unmasked"". The film adaptation ""Zodiac"" (2007) is roughly based on Graysmith's books, ""Zodiac"" (1986) and ""Zodiac Unmasked"" (2002). | 11.480 |
| 2 | 950423 | "The Zodiac Killer (film) The Zodiac Killer is a 1971 film directed by Tom Hanson. The Zodiac Killer begs for his father to talk to him only to be rebuked. | 10.271 |
| 3 | 361460 | "Los Zodiac Los Zodiac (sometimes Los Zodiacs or Los Zodiac's) were a Peruvian rock music band which formed in the 1960s. Los Zodiac performed with singers: Frank Sinatra, Palito Ortega and Leo Dan. | 9.840 |
| **Ranking** | **Candidate Suggestion** | **Score** |
| 1 | chinese zodiac | 0.1818 |
| 2 | zodiac signs | 0.0909 |
| 3 | zodiac tattoos | 0.0909 |
| **Ranking** | **Wikipedia Document Title** | |
| 1 | Zodiac (film) | |
| 2 | Zodiac Killer | |
| 3 | Zodiac (cipher) | |

**Query 3: home design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Document ID** | **Snippet** | **Score** |
| 1 | 917137 | "Freddy Wittop Freddy Wittop (July 26, 1911 ~~~ February 2, 2001) was a costume designer. Moving to Paris in 1931, he designed for the Folies Berg~~re and other music halls, creating costumes for Mistinguett and Josephine Baker, among others. | 8.647 |
| 2 | 257133 | "Kara Saun Kara Saun is a German-born fashion and costume designer and CEO of Kara Saun LLC, an LA-based fashion and costume design house. She also costume designed Disney~~~s ~~~Make and ""Jabbawockeez MUS.I.C. | 7.206 |
| 3 | 780014 | He was invited to London by John Gielgud to design the sets and costumes for Shakespeare's ""Much Ado about Nothing"", in 1949. He also designed the sets and costumes for Hector Berlioz's ""Les Troyens"", directed by Gielgud at Covent Garden in 1957. | 6.918 |
| **Ranking** | **Candidate Suggestion** | **Score** |
| 1 | Home design cad | 0.333 |
| 2 | Home design alternatives | 0.333 |
| 3 | Home metal decorative holder the bullseye design | 0.333 |
| **Ranking** | **Wikipedia Document Title** | |
| 1 | Interior Design | |
| 2 | Design | |
| 3 | Research Design | |

**Query 4: malignant melanoma**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Document ID** | **Snippet** | **Score** |
| 1 | 71642 | "Ecromeximab Ecromeximab is a chimeric monoclonal antibody being developed for the treatment of malignant melanoma. The drug was developed by Kyowa Hakko Kogyo Co., Ltd. | 24.384 |
| 2 | 182182 | "Acral lentiginous melanoma Acral lentiginous melanoma is a kind of lentiginous skin melanoma. Acral lentiginous melanoma is due as a result of malignant melanocytes. | 14.452 |
| 3 | 248490 | "Superficial spreading melanoma Superficial spreading melanoma (also known as ""superficially spreading melanoma"") (SSM) is usually characterized as the most common form of cutaneous melanoma in Caucasians. As the risk of spread varies with the thickness, early SSM is more frequently cured than late nodular melanoma. | 12.797 |
| **Ranking** | **Candidate Suggestion** | **Score** |
| 1 | Moles and malignant melanoma | 0.5 |
| 2 | Malignant melanoma family history | 0.5 |
| 3 | None | N/A |
| **Ranking** | **Wikipedia Document Title** | |
| 1 | Melanoma | |
| 2 | Acral lentiginous melanoma | |
| 3 | Skin cancer (redirect from Non-melanoma skin cancer) | |

**Query 5: what is that disney movie with the rock**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ranking** | **Document ID** | **Snippet** | **Score** |
| 1 | 367108 | "Disney Girlz Rock Disney Girlz Rock is a compilation of songs that have been used in Disney Channel. The album contains mostly songs that have been used on Disney Channel movies and shows. | 11.754 |
| 2 | 346615 | Other major DCOM franchises include ""Camp Rock"", ""The Cheetah Girls"", ""Twitches"", ""Halloweentown"", the ""Teen Beach"" films, the """" trilogy and ""The Descendants"" series. Disney Channel broadcast all of the 99 Disney Channel Original Movies that came before it, beginning with the 51 most popular films airing over the four-day weekend, beginning May 27, 2016. | 10.878 |
| 3 | 738945 | Much of the movie was filmed on location at Central High School. It aired on the Disney Channel on January 17, 1993. | 9.202 |
| **Ranking** | **Candidate Suggestion** | **Score** |
| 1 | No suggestions found |  |
| 2 | No suggestions found |  |
| 3 | No suggestions found |  |
| **Ranking** | **Wikipedia Document Title** | |
| 1 | Lady in the Water | |
| 2 | Teen Beach Movie | |
| 3 | The Great Movie Ride | |

## Insights and Lessons Learned

For this project, we probably spent most of the time worrying about and planning on how to deal with such a large corpus efficiently. We spent almost an entire week on trying to get our project to work with a database, because we felt that this would allow us to get the information extremely quick for the user and satisfy their needs over the time it would take to push data into the database. After quite some time we realized that with our current skill set this wasn’t possible, so we moved on to storing the program in memory. Once we did this, we noticed that we were only using a small fraction of our memory and the project went together rather quickly following this point. For future implementations, it might be useful for us to do some research on how other search engines do this and as stated in class “Not Reinvent the Wheel”.

Another thing that we learned during this process that falls in the same category as above, is that on a shorter deadline we can’t complete everything ourselves. When reached the point of wanting to limit our index size, we really needed to find a dictionary of words we could compare against. Doing so though, means that we had to rely on information already available and couldn’t spend too much time pruning this information to fit our needs specifically. While we feel like we did a good job of picking a relevant dictionary, we will probably spend more time on our preprocessing next project to improve upon what we already have.

When it comes to our actual project, there is a lot we can still learn to optimize our program. While it was totally workable, our index creation does take some time as seen in the stats above, but what we’d really like to optimize is the time it takes to load our index into memory after it has been created. This time is only about 5 minutes, but we believe through research and some thought process we could probably really work this time down to be much more responsive.

Overall, we think our best insight from this project was that the ability to adapt when things don’t work out as planned is extremely valuable for a project of this size. We had multiple occasions where we had a great plan, tried to implement it, and ran into issues that pretty much put us at a standstill. Luckily, we both are very adaptable and didn’t get hung up on what we had already created. Additionally, we did a really good job of still taking the valuable parts of what we already had with us into the next adaption we made, so we were never at a total loss. We hope that insights we have gained from this project and this adaptability will carry us through adding the necessary improvements on the next iteration of this project.