COMP 479 Project 2

Web Crawling and Document Clustering Analysis of Concordia's Spectrum

Ghali Oudghiri

40202095

Concordia University

Words:1193

Pages:5

Introduction

Concordia University's Spectrum platform is an extremely accessible repository, containing over 20000 academic documents in its diverse collection. The documents are categorized into different types, such as an 'Article', or a 'Book Section', or 'Book', or others. The primary aim of this project is to develop a program that is able to web crawl through Concordia's repository for Master's and PhD theses. Text processing will then allow us to have their contents retrieved into an inverted index, resulting in document analysis through k-means clustering.

This report will go through the methods of procedure, the effectiveness of clustering, scores of individual clusters, setting the stage for all the documents analyses.

Procedures & Protocol

Web Crawling and Scraping

The web crawling began at the spectrum library, more specifically at: **'https://spectrum.library.concordia.ca/view/year/{year}.type.html'** where {year} is iterated from 2024 by the amount inputted when the program first runs, as shown in **Figure 1** found in Demo_P2.pdf.

Several external packages are used for the web crawling and scraping process. **'Scrapy'** is a robust web crawling framework that's used throughout to handle requests and manage data. **'BeautifulSoup'** is next, which helps us parse and navigate through the HTML structure of the repository, allowing us to identify all of the pdf links. These links were always found in the same citation block **"ep_document_citation"**, always located in the **"group_thesis"** class. Within the citation block, return the links connected to a **"ep_document_link"** class. In order to exclude links that do not immediately open the pdf, either redirecting to a login page, any pdf link that also contained the phrase **"Restricted to Repository staff"** within the same citation block, was skipped. The logs found in **Figure 2** help visualize the process.

In order to disallow using redirects to identify whether a pdf url is usable, a robots.txt file is implemented in which **'Scrapy'** follows any rules set by it as seen in **Figure 4**. In addition, download delays were added to ensure there are no server issues.

Text & Document Processing

For text and document processing, **'PyPDF2'** is used allowing us to read and extract text from pdf files. The pdfs are downloaded one at a time, in which during its stay in my directory, **'PyPDF2'** extracts all the text from it. By using RegexpTokenizer from NLTK, we are able to remove tokens with numbers or punctuation, as well as lowercasing all of the tokens. This helps with decreasing the amount of non-word tokens, despite there still being a lot of them remaining. These tokens are then placed into an inverted index built using defaultdict, storing the token and its respective document IDs, as shown in **Figure 3**. Once all of the pdf is extracted, it is removed off of the directory before moving on to the next one.

Clustering

The document clustering process was implemented by using **'Scikit-Learn'**. Multiple different features were obtained through **'sklearns'** such as **'KMeans'** to perform the clustering, **'TfidfVectorizer'** to transform tokenized text into numerical vectors capable of representing significance of the token, **'TruncatedSVD'** to reduce dimensionality for LSA transformations and a few others.

K-means clustering iteratively assigns documents to clusters while minimizing the variance within each cluster to make sure that documents are more similar to those in the same cluster than to those in separate clusters.

All the while Tf-Idf is a common evaluation technique that takes into account the term frequency multiplied by the inverse of document frequency of a specific term. These terms are then all ranked by significance, from 1-20 or 1-50, in their respective clusters.

The number of clusters (k) is a critical parameter that massively influences the clustering outcome. For this project, the number of clusters was determined based on the number of departments and

faculties, in order to accurately reflect the academic documents found in Concordia's Spectrum repository. In addition, the number of clusters could be directly affected by the number of documents (n), as we noticed that there must be at least n-1 clusters for each clustering. This deduction was made due to the consistently useless results whenever k = n, as each document gets its own cluster and no progress is made. Even then, if k = n - 1 it will still produce nothing-results because a cluster will at most have 2 documents which is usually not enough for a respectable result.

Clusters will be evaluated by using the **Silhouette Score**, which goes through the clusters and measures how similar, on average, are the documents to their own cluster compared to other clusters. This will result in a score of between -1 and 1, with a higher score indicating well-separated clusters and therefore a better-defined cluster. These scores will be further visualized by using **'Matplotlib'** in order to help compare the different clusters in each clustering as shown in **Graph 1** and **Graph 2**.

Discussion

Clustering Results & Analysis

As stated previously, we were expected to run 2 different clustering runs each with a number of clusters dependent on the amount of departments and faculties in Concordia University. The number of faculties was simple to determine being only 4, "**John Molson School of Business"**, **"Gina Cody School of Engineering and CompSci"**, **"Fine Arts",** and **"Arts and Science"**. The number of departments required some post-filtering, as over 90 were found in the divisions tab of the repository's website, however many of these divisions were not a part of any faculty mentioned above, or did not contain any or a limited amount of theses during the last several years. After deciding to exclude all departments that did not have at least 3 documents in each of the last several years, the total number of departments was 38.

After running the program with 2 clusterings, 4 and 38 clusters respectively, I decided to process up to 151 documents in the year of 2024, which as shown in the .txt results files **Figure 5 and 7** turned into 120 total documents.

1. The clustering with k=4, reflecting the faculties, obtained an average silhouette score of 0.565, which on a scale from -1 to 1 is a respectable result. The 4 clusters are all individually scored as well:

   Cluster 0 Score: 0.686
   Cluster 1 Score: 0.571
   Cluster 2 Score: 0.357
   Cluster 3 Score: 0.904

   as visualized in **Graph 1**. The 4th cluster, Cluster_3, was undoubtedly the best ranked out of the 4. We can see this cluster's top 50 vocabulary ranked by tf-idf in **Figure 6**, which helps to deduce that the 6 documents in cluster_3 are french-based texts as in token rankings, a majority of the vocab is in french,reflecting its extremely high silhouette score as these words will most likely not appear whatsoever in an english text**.** While the lowest ranked is Cluster_2 with a positive score of 0.357. The number of documents within the cluster is commonly associated with scores or ranking, however although Cluster_3 did have only 6 documents, while the other 3 had between 29-47 documents, Cluster_2 did not have the largest amount, meaning that the amount of documents in the cluster is not enough to determine its performance.

2. The clustering with k=38, representing departments, resulted in an average Silhouette score of 0.144, much lower than the previous clustering. Looking at **Figure 7**, and especially **Graph 2**, we are able to see the massive variation in scores when implementing a large number of clusters. Scores vary from up to 0.629 all the way down to -0.053. **Figure 8** displays the top 50 vocabulary ranked by tf-idf for Cluster_30, which is the lowest ranked out of all the clusters in both clusters. A negative silhouette score mainly represents a misclassification of documents. This means that some of the documents are not categorized in their right, in this scenario, "department". 4 different clusters are facing this issue, while an additional 6 clusters only contain 1 document and is therefore a useless cluster with a score of 0.00 as seen in **Graph 2**. This issue likely came up from the method of choosing k, as well as due to the limited dataset size. Using only 120

documents from 2024 worked well when categorizing into 4 faculties, but separating into 38 departments that may not have even been represented yet due to insufficient documents could lead to poorly defined clusters. While increasing the dataset size could potentially improve upon any issues, it will also significantly increase process times as the original attempt of 151 documents already faced heavy time constraints.

Looking at **Figure 10** and **11** helps us understand roughly what ends up happening with the categorization of the clusters. We mentioned previously how in k=4 clustering, due to some of the documents being in French, cluster_3 ended up with a ridiculous 0.9 score. With an increase of clusters, the magnitude of this separation between languages has decreased but can still be identified as seen in the 5th highest ranked cluster in **Figure 10.** While many of the high rated clusters in k=38 are only 2-3 documents large, they at least remain accurate as the 2-3 documents are commonly from the same department, or at worst the same faculty. Same can be said for k=4 clustering, as despite the french documents skewing the results for 1 cluster, the other 3 clusters contain well categorized documents, with the 'worst' one having to deal with the consequences of french being classified as its own faculty.

Conclusion

As we've gone through the report, we've detailed the development and analysis process of a web crawler designed to obtain, processed, index, cluster, and rank the Master's and PhD theses from Concordia University's Spectrum repository through the use of several different python libraries and external packages. Number of clusters per clustering was deduced to be k=38 and k=4, based on the number of viable departments and faculties, and after running the program, k=4 clustering aligned really well with faculty distinctions, despite the slight hurdle due to the French documents. Conversely, the k=38 clustering faced multiple challenges potentially due to limited data and misclassification, yet its higher scored clusters demonstrated considerable accuracy when categorizing documents. Overall despite encountering issues, particularly with larger cluster counts, the results were still satisfactory.