Report for Project 4 of COMP 479 by Silas Kalinowski ID: 40256077

The demo can be found in the file demo.pdf.

Design and summary of approach:

My project consists of 5 modules:

The first module named 'web_scraper.py' implements the web crawler based on the scrapy framework.

The second module named 'main.py' which calls the other functions and runs the project automatically.

The module 'clustering.py' contains the functions to implement the clustering. Also, the functions plot and save the results.

The module 'utils.py' contains a function to save the text from the downloaded web pages, a function to save the titles of a web page and a function to save a panda data frame to a txt file.

The module 'sentiment.py' contains the function to estimate the sentiment of the documents and plot, print and save the results.

For my design I ensured that every part of the project is implemented in separate modules. The modules then again consist of function that implement a specific part of the assignment.

I used panda data frames and matplotlib to visualize and give a good overview of the data obtained.

Also, the most important measures are saved as txt files. I also made sure that different methods like stemming or not stemming when vectorizing can be easily disabled.

Tools used for Web scrapping:

I used scrapy as my web crawler framework. To implement my web crawler I took the example spider on https://scrapy.org/ as a foundation. I extended it with the additional functionalities required for this project: I set a reasonable download delay of 0.5 and started crawling at https://www.concordia.ca/ginacody.html' and only allowed domains that start with www.concordia.ca. I also implemented an attribute num_pages that can be used to define the number of pages to be downloaded. The query is closed when that limit is reached. Additionally, I set the parameter ROBOTSTXT_OBEY to True so that if any website has a robots.txt it is obeyed. Since most of the pages that are crawled don't contain a robots.txt and rather have the rules for web crawler saved in a meta tag, I implemented a function to obey the meta tag too. The function is called 'checkforrobot' and uses BeautifulSoup to first find the robots meta tag and then read it and return the wanted result. Based on the return the crawler indexes the page and follows links on the page or not. Also, the bodies of the crawled pages are saved the Files directory. To follow the links on the websites to other websites all links that are provided are extracted and a new request with the link is requested.

Extracting the text from the web pages

To extract the text from the downloaded html files I used BeautifulSoup. Specifically, I used the get_text() from BeautifulSoup because it was simple and effective. This function returns all of the text in the page. I also tried different methods of extracting the text. For example, I tried only extracting the text in the contain-main elements in the files. But since the pages have many different structures and store the important text in different elements the get_text() function was the most appropriate. I also wrote a function that extracts the titles of the web pages to experiment with clustering only with the titles. (see in Demo file)

Different behaviour of the two clusterings

I used the random_state = 0 to make my results reproducable. For naming I tried basing my names on the most informative words and the titles of the documets. I only used non dicriminative methods, since discriminative methods were difficult to implement and the non discriminative methods gave me a valid names.

K=3

I noticed that most of the documents are in cluster 1. Only 6 documents are in cluster 0 and only 6 in cluster 2. The remaining 38 documents are all in cluster 1. The documents name with label and sentiment score can be found in the Project Directory named doc_name_sentiment_label_3.txt.

The cluster 0 contains mostly documents with a low sentiment score compared to the other documents. The only document in cluster 0 with a sentiment score over 0 is the document about traductologie. These score mostly seeam reasonable but all of the documents have a French url. Also when looking at the documents for all documents at least some parts are written in French. Additionally all the most informative words are all French. If I would have to name this cluster I would probably name it give I the title French student help and recruitment.

The second cluster (cluster1) contains the most documents with a high variance of sentiments and topics. It includes documents about events, the school in general, news, research, … . When looking at the 20 most informative words all the words are very general terms. Most terms are terms that are probably used in most of the other documents too. So it is difficult to give this cluster a name, but if I would give this cluster a name it would be: general information about Concordia.

The third cluster (cluster2) contains mostly documents that talk about mental health or counselling, or some other kind of service. One documents is about events. Three documents have an average sentiment and the other three have a very high sentiment. Overall the sentiment of the cluster is very positive. The cluster contains three of the top 5 documents with the highest sentiment. Also the 20 most informative words are all mostly about psychological help and mental health. If I would have to name this cluster I would name it: mental health services at concordia

Overall I found it interesting that documents that are written in French are assigned to one cluster. This shows one of the applications of clustering. Besides that I also found it interesting that documents with similar sentiment are more likely to be in the same cluster. Besides that the clustering with k=3 is not that viable because the clusters are so different in size. It would be better if cluster 1 would be divided into different clusters with different topics.

K=6

Cluster 0 only contains 1 document. Cluster 1 is the biggest cluster and contains 28 documents. Cluster 2 contains 5 documents. Cluster 3 contains 4 documents. Cluster 4 contains 8 documents. Cluster 5 contains 3 documents. The documents name with label and sentiment score can be found in the Project Directory named doc_name_sentiment_label_6.txt.

Cluster 0 only contains one of the French documents. Compared to k=3 this document now belongs to its own cluster. As expected are all the most informative words in French. I would name it: information about translation studies in French.

Cluster 1 still contains many documents but is does not contain as many documents as with k=3. As hoped in the analysis of the clustering with k=3 the prievous cluster1 is now split into different clusters. It is still contains a rather diverse set of documents with different topics. But most of the documents are about general information about Concordia and go into more detail about the academic and research at Concordia. I would name this cluster: general information about Concordia for Students.

Cluster 2 contains nearly the same documents as the ‘French’ cluster from k=3. It only misses the traductologie document. As expected all the most informative words are French. I would give this cluster the same name as before: French student help and recruitment.

Cluster 3 is very similar to the cluster 2 from k=3 and mostly contains very positive documens that talk about counselling and mental health. It only misses the eventy document. Now the cluster is even more homogenous. All the most informative terms are about mental health and counselling. The event document is now in cluster 5. I would give this cluster the same name as before with just one addition: mental health services and counselling at Concordia.

Cluster 4 contains part of the documents that were split off from the previous cluster1. It mostly consists of positive documents and talks about general infromation about the university, Montreal but also documents about methodology and the committee. The most informative words are mostly about the arts faculty and research and academic opportunities at Concordia. So I would name this cluster: The arts faculty and research at Concordia.

Lastly, Cluster 5 contains the events document that was previously part of the counselling, mental health cluster. Now Events, another events information and a deadlines document are in a separate cluster.Based on the 20 most informative words and the documents tiltles I is very clear that this cluster contains documents that contain important deadlines and also documents that announce events. This is the only cluster in which most of the 20 most informaive words are dates and times. I would call this cluster: Events and deadlines.

Assessment of the clusters

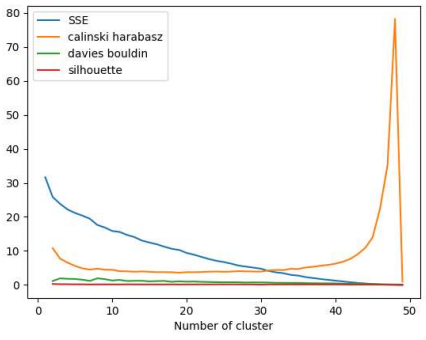
To convert my documents into vectors I used scikit tfidf and CountVectorizer. After some experiments the tfidf vectorizer yielded better results for me therefore I chose to use it from that point on. I experiemented with different compression techniques: stemming and removal of stop words. Especially the removal of stop words was very helpful for finding the most informative words. Stemming improved the clustering, but when trying to name the clusters the stemmed words were a little less helpful. That is why I used the clustering without stemming but with the removal of stopwords for the naming of the clusters.

To assess my clusters I used different measures provided by scikit learn. I used the slihouette socre, calinski harabasz score, the davies bouldin score and the SSE score. I decided to use these score because they were the most commonly used internal measures provided by sklearn. I could not use external mesures because we did not have target labels.

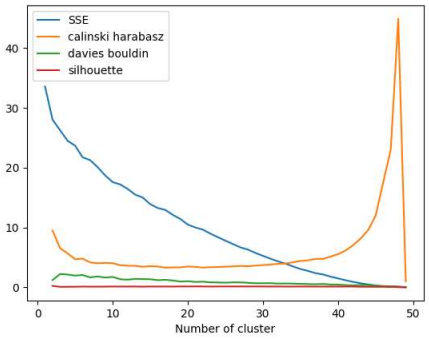
I plotted the results with the x-axis displaying the number of clusters and the y-axis the different score. When comparing the different scores from 3 clusters and 6 clusters with stemming I noticed that all measures improved. All the scores from the 6 clusters were lower. Without stemming only the SSE improved, the silhoutte, calinski harabasz and davies bouldin score decreased. One of the reasons might be that stemming reduces the dimensionality of the data and therefore makes the clusters more distinct.

I mostly plotted the graph for all possible number of clusters to get an estimate of which number of clusters would be ideal. I wanted to use the elbow method but the plot does not have an appropriate 'elbow point'.

With stemming



without stemming



Document sentiment

Top 5 documents with highest sentiment: 39_events.txt 212.0, 9_sara-weinberg-award-for, excellence-in-teaching.txt 204.0, 5_services.txt 173.0, 17_facilities.txt 150.0, 7_counsellors.txt 142.0

Top 5 documents with lowest sentiment: 4_confidentialite.txt -13.0, 1_comment-sollicitier.txt -11.0, 2_recrutement-etudiants.txt -10.0, 21_emplois.txt -7.0, 3_accessibilite.txt -7.0

The sentiment values were very useful. Especially the document with the highest sentiment and the document with the lowest sentiment was interesting. The document with the highest sentiment was the event web page. Advertising events and reporting and past events mostly consists of postive language and little negative language if any. The document with the lowest sentiment was the confidentialite web page. This webpage talks about Privacy and Data Protection. Most of the words have a negative connatation e.g. restriction and the web pages talks mostly about rules and laws.

Sometimes I felt like the Afinn score was not fully accurate because some of the documents in my test data had parts of it written in French. The Afinn package is only built for english or danish text.

What was also interesting when looking at the sentiment score of the documents in each cluster some cluster had a high variance in the sentiment of the documents and some had a very low variance. But for all clusters most documents where around the mean. One of the reason why similar documents (based on the cluster) are also similar in sentiment is that similar words are used which results in lower distance in vector space and similar sentiment score. Also as already mentioned above most of the documents with a high/low sentiment talk about a similar topic. Therefore they also are closer in vector space making it more likely that they are in the same cluster.

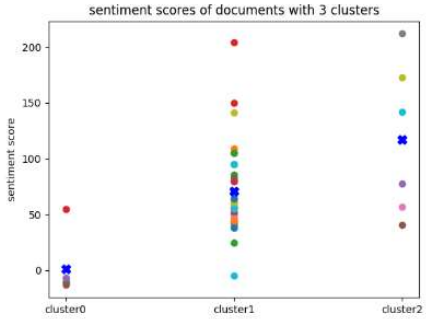
Different measures of Cluster sentiment scores

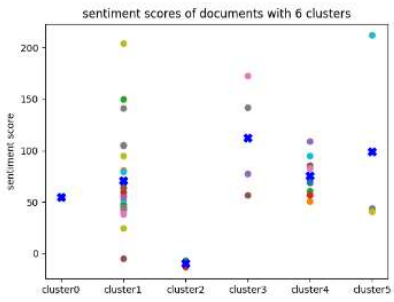
I used the Afinn script to evaluate the sentiment of every document seperately. Specifically, I used the Afinn.score() method to score each of the text that were extracted from the web pages. I tried using the table of words with assigned score from the Afinn website bu the scores were in my opinon less accurate than the scores from the score method.

```
   cluster   min    max  median   mean
0        0 -13.0   55.0    -8.5    1.2
1        1  -5.0  204.0    62.5   71.3
2        2  41.0  212.0   110.0  117.2
document with highest sentiment 39_events.txt 212.0
document with lowest sentiment 4_confidentialite.txt -13.0
```

```
   cluster   min    max  median   mean
0        0  55.0   55.0    55.0   55.0
1        1  -5.0  204.0    59.5   70.8
2        2 -13.0   -7.0   -10.0   -9.6
3        3  57.0  173.0   110.0  112.5
4        4  51.0  109.0    71.0   75.8
5        5  41.0  212.0    44.0   99.0
document with highest sentiment 39_events.txt 212.0
document with lowest sentiment 4_confidentialite.txt -13.0
```

To calculate cluster sentiment I tried different formulas: min, max, median, mean. I also thought about using a more sophisticated formula like weighted average, etc. but to do that we had to somehow estimate the weight for each document which was difficult and not appropriate for this project. From the methods I used the mean was the most meaningful as a broad overview over the sentiment. The min and max formulas where only helpful when looking at both of them and even than they are not very meaningful since you can only derive the range of the sentiment of the documents with them. You don't get a meaningful measure of the sentiments of all of the documents since the score is decided by one document only. The median was mostly very close to the mean and also viable. For some clusters the median gives a better measure of general tendency of the sentiment scores because it is not as sensitive to outliers as the mean. But for the majority of the examples as can be seen in the graphs (mean is marked as a blue cross) the mean gives a meaningful generale sense of the sentiment of the cluster. This is why I used the mean as my main measure.





Experience with crawling and scraping

My experience with crawling and scraping was mostly positive. The only negative experience was to get my crawler working and deciding on which crawler framework to use. I tried to use spidy at the beginning but I encountered an error when trying to install the package. So I switched to scrapy which was easy to install and use. I later realised that spidy uses an outdated dependency. This was the reason why I encountered an error when trying to install spidy. With scrapy all of the functionality required in the project were simple to implement and well documented online. I was surprised how easy it is to scrape hundres of pages in a few minutes without much prior knowledge. I will definitely use scrapy in future projects whenever I need to scrape web pages. The only difficult part in the implementation of my crawler was the attribute to stop the crawler after a set amount of pages. There is a setting in scrapy for that but it did not work for me. So I needed to implement it myself. Also obeying the robots.txt was more difficult than expected because most concordia pages don't have a robots.txt and only give the information about robot rules in the meta tag of the web page. Therefore I implemented a function to check that meta tag using BeautifulSoup. Generally, BeautifulSoup is a very simple and helpful tool to extract information from web pages. I made the experience in project one that it was rather difficult to extract information from a html document without using BeautifulSoup. For the crawler I also tried different setting regarding the depths and the breadth of the search. A rather deep search mostly returned more similar documents and a rather breadth-focused search returned a more diverser document set.

20 most informative words

The 20 most informative words were very helpful to get an overview of the topics in the clusters. This was very helpful as mentioned above for the naming of the clusters. The 20 words only became helpful when I removed stopwords and other words that did not contain any alphanumeric letters. For example without stop word removal the following top 20 words were printed:

Cluster 0: , and of the . to & concordia in for arts school calendar : a services academic faculty student graduate

Cluster 1: , the of and ” “ . to in a concordia & s for ’ putrescine sustainability program we on

With stop words removal the words were more helpful:

without stemming

Cluster 1: concordia school calendar graduate academic arts services science student university schools class colleges research campus faculty students fine news media

With stemming the 'quality' of the words was almost the same only that that the words were stemmed and therefore sometimes hard to differentiate e.g.:

Cluster 2: counsel p.m. health servic school appoint psycholog dec student concordia mental concordia-events-categori psychologist support calendar medic a.m. event-audi well graduat

The files of the 20 most informative words can be found in the Project Directory named 20_most_informative_words_6, 20_most_informative_words_6.txt. To obtain my 20 words per cluster I sorted the centroids and printed the top terms that were most influential to form this centroid. I did not use any discriminative methods.

I certify that this submission is my original work and meets the Faculty's Expectations of Originality.

Signature ID: 40256077 Date 2022-12-12