Concordia University

Engineering and Computer Science

COMP 479/6791
Information Retrieval and Web Search

Assignment 4

Report

Under the guidance of Dr. Sabine Bergler

Vasu Ratanpara 40135264

December 2021

Introduction

In this project, We had to use any one of the mentioned web crawlers from the Project 4 description. I have used scrapy (https://scrapy.org). After scrapping the text from the web pages we have to create 3 and 6 clusters from the same data. Later on, we had to do a sentimental analysis of each cluster and tell the score and see if the is positive, negative or neutral.

Design Summary & illustration

I have only one main file which is main.py which is responsible for starting the crawler and scrapping the given number of pages given by the user. After that for the k-mean, I am reading the data.json which contains text, title and URLs which got from 1st part of the project. It will generate the first 3 clusters and later on 6 clusters. I will do Afinn Analysis on each cluster afterwards.

Web Crawler

Since I am using scrapy, It has an in-built option to obey the robots.txt which can be set by passing/setting up ROBOTSTXT_OBEY = True. The way I scrapped web pages are recursive. First I am reading the web pages and saving them as HTML files and processing them further. I extracted all the text and saved the title, text and URLs of each page in data.json one by one.

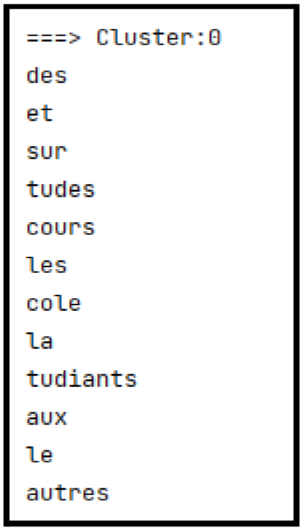
K-means Clustering

For the K-means clustering, I have used the sci-kit learn package. **The Clustering is done using the TF-IDF.** KMeans normally works with numbers only: I need to have numbers. To get numbers, I did a common step known as feature extraction.

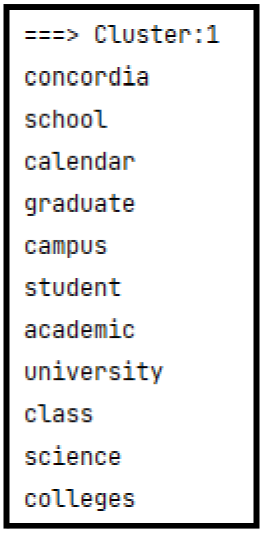
The feature I used is TF-IDF, a numerical statistic. This statistic uses term frequency and inverse document frequency. In nutshell, I used statistics to get to numerical features. I have used the existing implementation of the TF-IDF algorithm in scikit-learn. The method TfidfVectorizer() implements the TF-IDF algorithm. Briefly, the method TfidfVectorizer converts a collection of raw documents to a matrix of TF-IDF features.

After I got numerical features, I initialize the KMeans algorithm with K=3 and later K=6 and I got the following results.

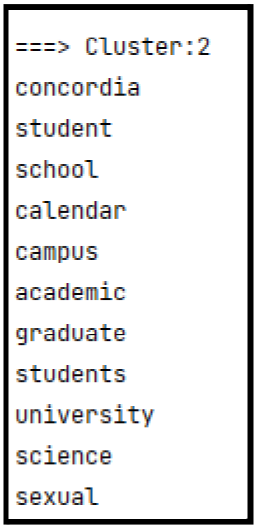
K=3



```
===> Cluster:0
des
et
sur
tudes
cours
les
cole
la
tudiants
aux
le
autres
```



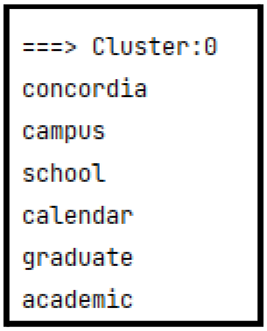
```
===> Cluster:1
concordia
school
calendar
graduate
campus
student
academic
university
class
science
colleges
```



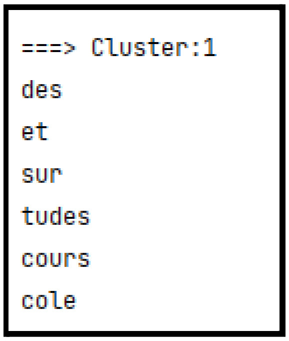
```
===> Cluster:2
concordia
student
school
calendar
campus
academic
graduate
students
university
science
sexual
```

Cluster	0	1	2
Given Name	French Cluster	Education Cluster	Education Cluster

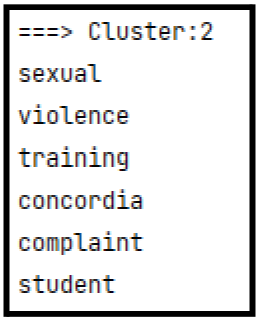
K=6



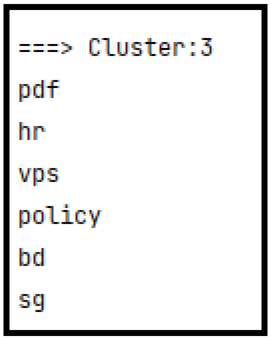
```
===> Cluster:0
concordia
campus
school
calendar
graduate
academic
```



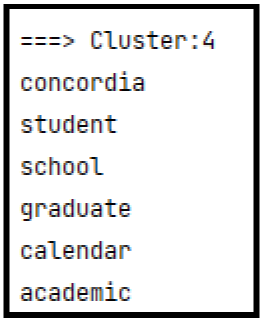
```
===> Cluster:1
des
et
sur
tudes
cours
cole
```



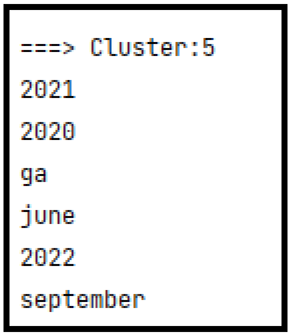
```
===> Cluster:2
sexual
violence
training
concordia
complaint
student
```



```
===> Cluster:3
pdf
hr
vps
policy
bd
sg
```



```
===> Cluster:4
concordia
student
school
graduate
calendar
academic
```



```
===> Cluster:5
2021
2020
ga
june
2022
september
```

Cluster	0	1	2
Given Name	Education	French	English Strong words
Cluster	3	4	5
Given Name	Short forms	Education	Months, Years & Dates

As you can see the clusters above, first we will talk about clusters where K=3, Cluster 0 is a good example of a cluster since it only has french words, While Cluster 1 and 2 are bad examples since they both have many similar words. For K=6, Cluster 1, 2,3 and 5 are good examples of clusters.

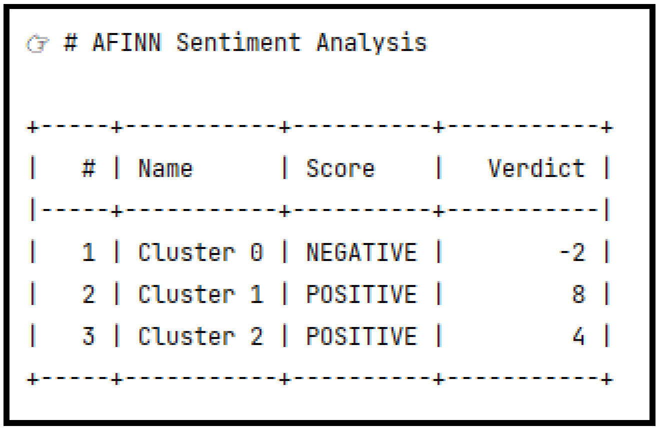
In the first Clustering where K=3, we only got abstract cluttering but with K=6 we got a deep classification of clusters which were further dived and created separate clusters from the original data.

Afinn Sentiment Analysis

The Implementation of Afinn is used from the mentioned package from the project document. According to my opinion, the best way to determine the Sentiment Score for any clusters is to calculate the score of each word in that cluster and add them up. The total score for each cluster will represent how positive, negative or neutral it is as compared to other clusters.

The Sentimental analysis can be very useful especially with real-time data. You can get to know the general opinions of a group of people. It's also really easy to cluster them and apply the analysis. We can determine what kind of expressions are described by the person.

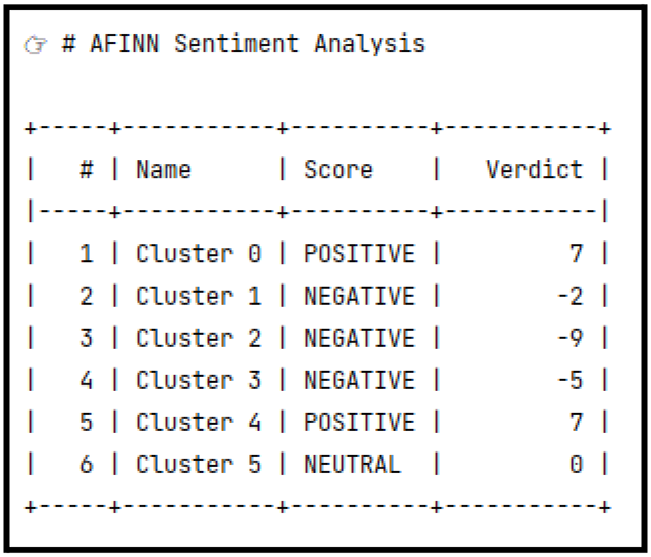
K=3



```
☞ # AFINN Sentiment Analysis

+-----+-----------+----------+-----------+
|   # | Name      | Score    |   Verdict |
|-----+-----------+----------+-----------|
|   1 | Cluster 0 | NEGATIVE |        -2 |
|   2 | Cluster 1 | POSITIVE |         8 |
|   3 | Cluster 2 | POSITIVE |         4 |
+-----+-----------+----------+-----------+
```

K=6



```
☞ # AFINN Sentiment Analysis

+-----+-----------+----------+-----------+
|   # | Name      | Score    |   Verdict |
|-----+-----------+----------+-----------|
|   1 | Cluster 0 | POSITIVE |         7 |
|   2 | Cluster 1 | NEGATIVE |        -2 |
|   3 | Cluster 2 | NEGATIVE |        -9 |
|   4 | Cluster 3 | NEGATIVE |        -5 |
|   5 | Cluster 4 | POSITIVE |         7 |
|   6 | Cluster 5 | NEUTRAL  |         0 |
+-----+-----------+----------+-----------+
```

Learnings

My Experience was very good with the project. It’s a really very helpful project to understand how Search Engines crawl web pages which can be later on used by indexing, clustering etc.