Information Retrieval

COMP 479 Project 4 DEMO

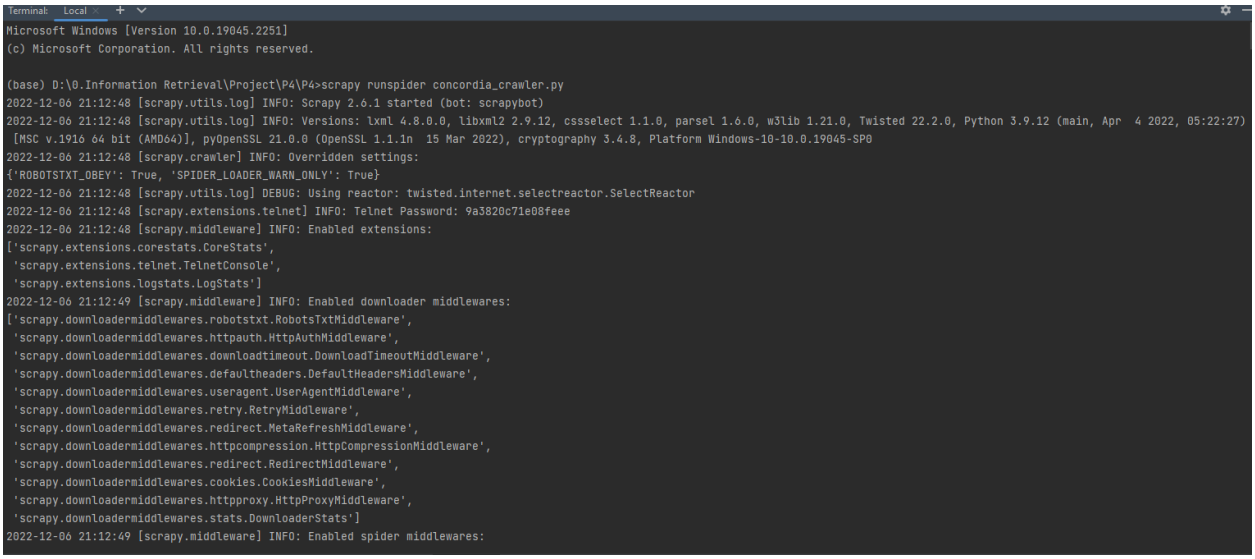
Lin Ling (40153877)

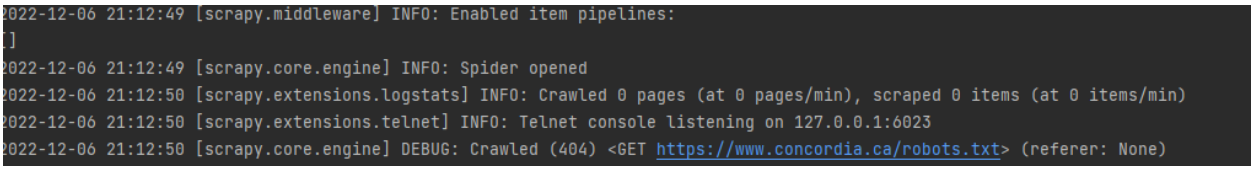
A report submitted in partial fulfilment of the requirements of Comp479.

Concordia University

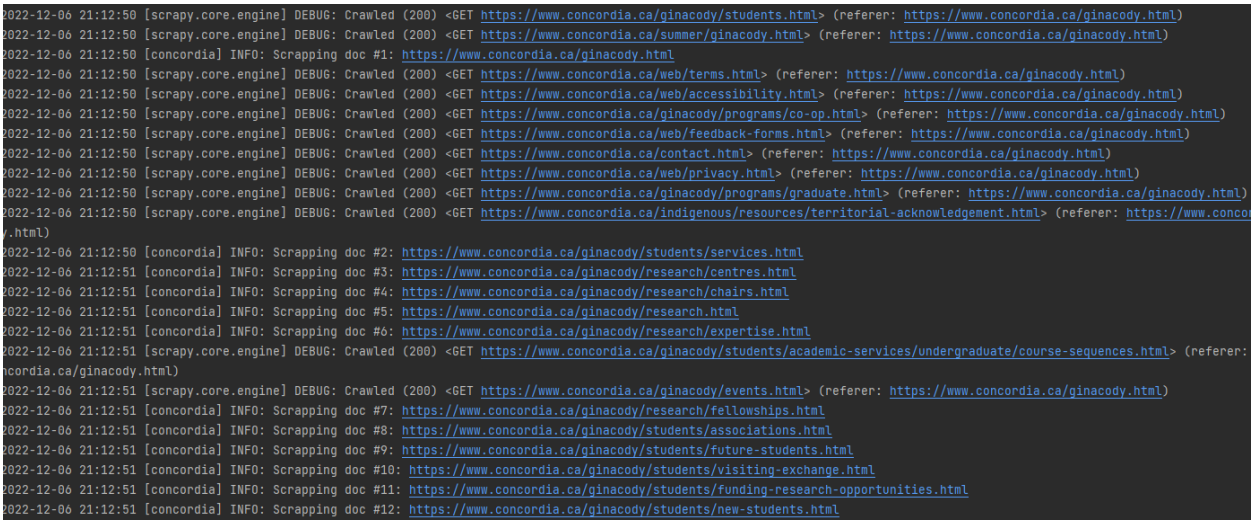
If you run the concordia_crawler.py at terminal: scrapy runspider concordia_crawler.py

And the terminal will show what happening as follows:





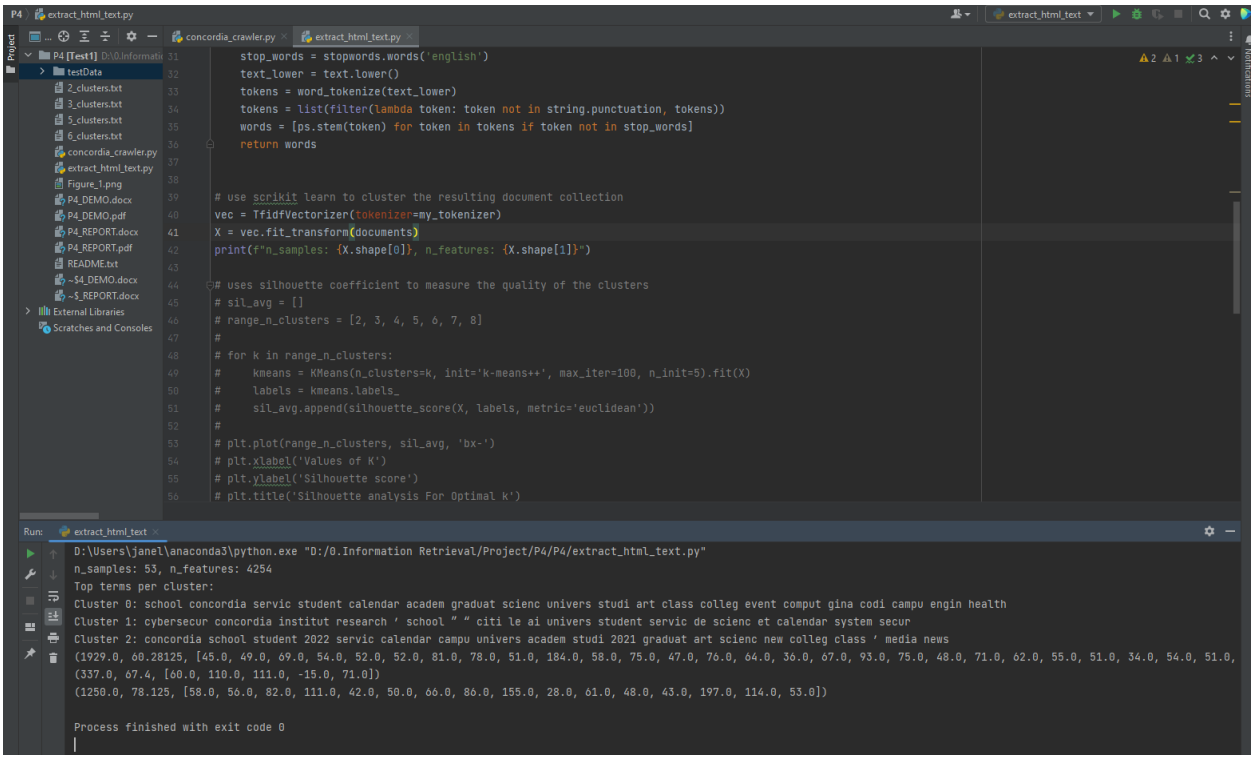
Start to crawl and download the web pages, start from doc#1 and will end at doc#100 as files limit set up to 100.



The spider will be closed as limit reached.

```
2022-12-06 21:12:55 [scrapy.core.engine] DEBUG: Crawled (200) <GET https://www.concordia.ca/cunews/main/stories/2020/07/14/new-classrooms-and-study-locations-coming-to-concord
uilding.html?rootnav=news/stories> (referer: https://www.concordia.ca/cunews/main/stories/2022/04/11/concordias-indoor-faubourg-building-parking-reopens.html?c=/hospitality)
2022-12-06 21:12:55 [concordia] INFO: Scrapping doc #100: https://www.concordia.ca/news/stories/2022/10/28/concordia-releases-its-final-report-of-the-presidents-task-force-on-
ism.html?c=/hospitality
2022-12-06 21:12:55 [scrapy.core.engine] INFO: Closing spider (Limit reached.)
```

If you run the extract_html_text.py, it will print out the result of two different clustering runs.



Print out afinn scores is First is total afinn score, second is average afinn score and rest is the list of each document’ afinn scores.

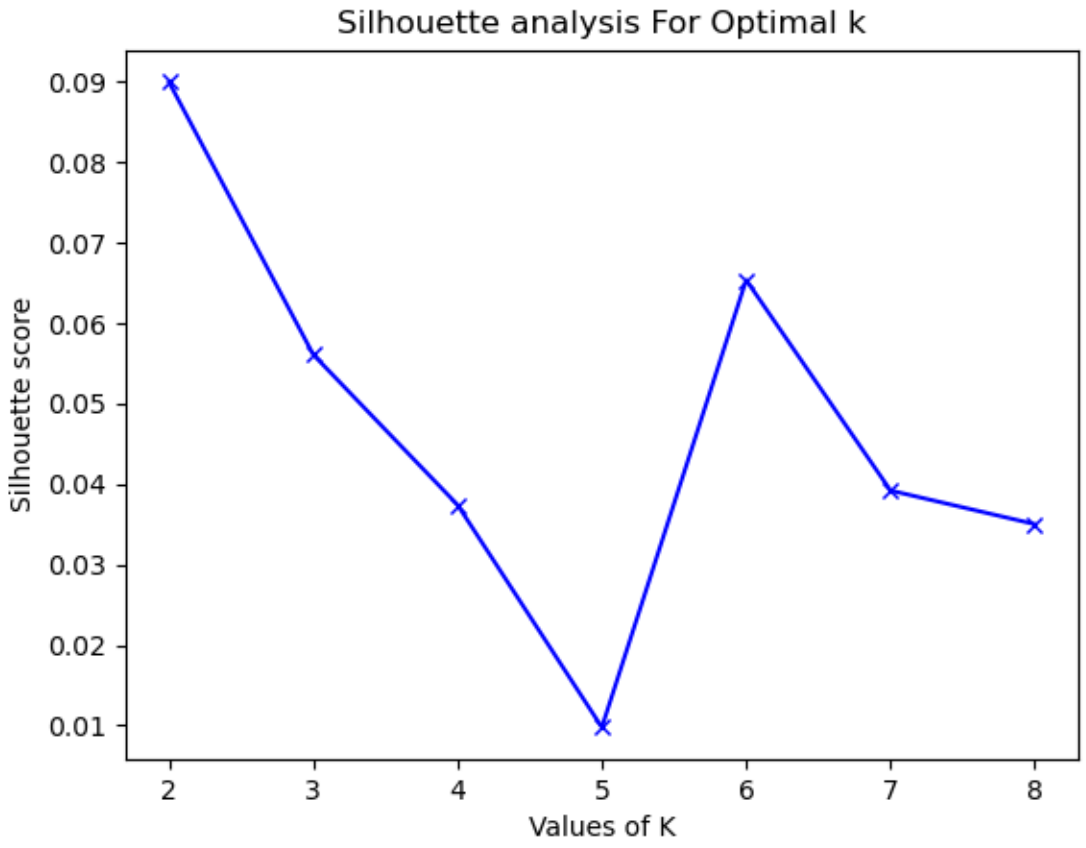
If you want to check more details, please check report here. If you want to see the output of them, please check k=3 output and k=6 output. And in the outputs, you could find the sentiment scores for each cluster.

uses silhouette coefficient to measure the quality of the clusters

```python
# uses silhouette coefficient to measure the quality of the clusters
sil_avg = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8]

for k in range_n_clusters:
    kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=100, n_init=5).fit(X)
    labels = kmeans.labels_
    sil_avg.append(silhouette_score(X, labels, metric='euclidean'))

plt.plot(range_n_clusters, sil_avg, 'bx-')
plt.xlabel('Values of K')
plt.ylabel('Silhouette score')
plt.title('Silhouette analysis For Optimal k')
plt.show()
```



To find an optimal value for the number of clusters K, we use a silhouette plot to display a measure of how close each point in one cluster is to a point in the neighboring clusters and thus provide a way to assess parameters like the number of clusters visually. Let’s see how it works.

- Compute K-means clustering algorithm for a range of values.
- For each value of K, find the average silhouette score of data points:
- Plot the collection of silhouette scores for each value of K
- Select the number of clusters when the silhouette score is maximum:

It shows k=2 is optimal value and here is the output k = 2.

It shows k=5 is optimal value and here is the output k = 5.