CS 1005

Discrete Structures

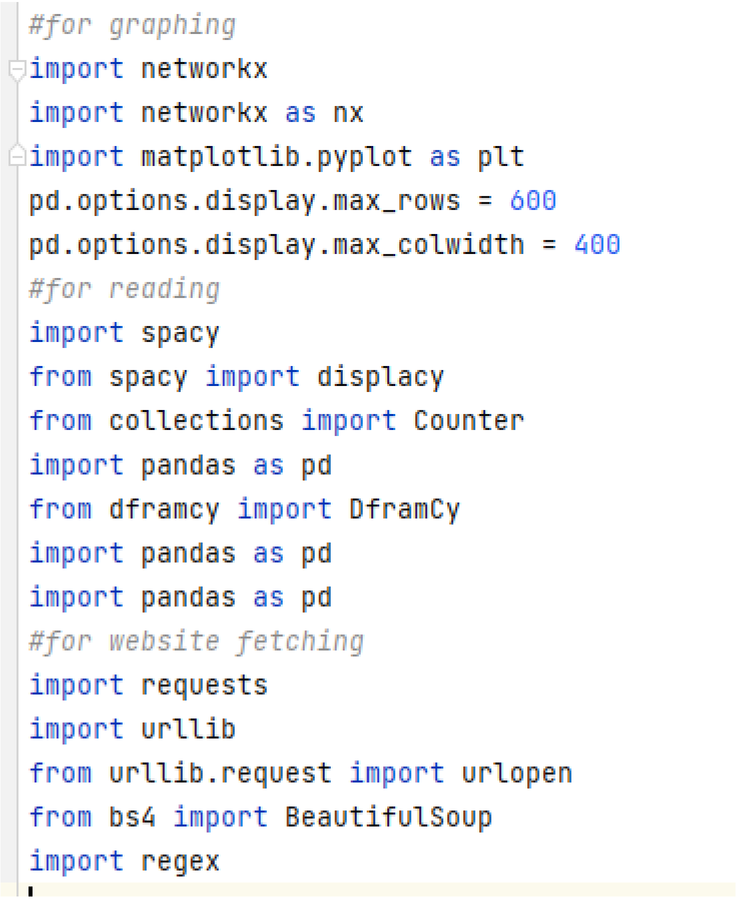
Project Phase-II

Group 10

20I-1866	Faraz Ud
20I-1822	Rayed Sayed
20I-1893	Ahmad
20I-1811	Danyal Memon

Table of Contents

Libraries used



```
#for graphing
import networkx
import networkx as nx
import matplotlib.pyplot as plt
pd.options.display.max_rows = 600
pd.options.display.max_colwidth = 400
#for reading
import spacy
from spacy import displacy
from collections import Counter
import pandas as pd
from dframcy import DframCy
import pandas as pd
import pandas as pd
#for website fetching
import requests
import urllib
from urllib.request import urlopen
from bs4 import BeautifulSoup
import regex
```

Web-Scrapping (Step 1)

Websites used to collect data

- ARID
- UET
- COMSATS

Links to webpages

- http://www.uaar.edu.pk/about-us.php?content_id=100
- http://www.uaar.edu.pk/about-us.php?content_id=100
- http://www.uaar.edu.pk/fss/index.php
- https://uet.edu.pk/aboutuet/aboutinfo/index.html?RID=about_uet_future_vision
- https://www.uet.edu.pk/
- https://www.uet.edu.pk/aboutuet/aboutinfo/index.html?RID=spinoffcompanies
- http://islamabad.comsats.edu.pk/
- https://admissions.comsats.edu.pk/

Code for part 1

- Fetching data from Website and storing in .txt files

```
#function to get data
def dataall(url):
    h2_headers = []
    for link in url:
        print(link)
        link = requests.get(link)
        html = link.text
        text1 = BeautifulSoup(html, 'html.parser')
        for header in text1:
            for paragraph in text1:
                header_contents = header.text
                h2_headers.append(header_contents)

    return h2_headers
if __name__ == '__main__':
    print("Reading Links----")
    urluaar1=[]
    urluaar1=("http://www.uaar.edu.pk/index.php","http://www.uaar.edu.pk/about-
us.php?content_id=100","http://www.uaar.edu.pk/fss/index.php")
    urluaar1=dataall(urluaar1)

    urluet=[]
    urluet =
("https://www.uet.edu.pk/","https://uet.edu.pk/aboutuet/aboutinfo/index.html?RID
=about_uet_future_vision","https://www.uet.edu.pk/aboutuet/aboutinfo/index.html?
RID=spinoffcompanies")
    urluet = dataall(urluet)

    urlcomsats1=[]
    urlcomsats1 =
("https://admissions.comsats.edu.pk","http://islamabad.comsats.edu.pk","https://
www.comsats.edu.pk/AboutCIIT/")
    urlcomsats1 = dataall(urlcomsats1)


    # WRITING DATA IN A FILE
    #http://islamabad.comsats.edu.pk/

    Fast_data = [urluaar1]
    Giki_data = [urluet]
    Comsats_data = [urlcomsats1]
    print("Saving Data----")
    File_object = open(r"Uaar.txt", "w+")


    try:
        File_object.writelines(urluaar1)
    finally:
        File_object.close()

    File_object2 = open(r"Uet.txt", "w+")

    try:
        File_object2.writelines(urluet)
    finally:
        File_object2.close()

    File_object3 = open(r"Comsats.txt", "w+")

    try:
        File_object3.writelines(urlcomsats1)
    finally:
        File_object3.close()
```

Comparison of Nouns, adjectives, and verbs in websites (Step 2)

Arid University						
	NOUNS		Adjectives		Verbs	
#	character	count	character	count	character	count
1	Home	82	4th	32	Contact	16
2	Downloads	70	Curricular	26	says	15
3	|	58	-	21	Directorates	13
4	research	45	various	17	ORIC	13
5	Directorate	39	other	16	CASD	13
6	faculty	34	former	14	Facts	13
7	Team	26	Available	13	started	13
8	education	25	agricultural	13	irrigated	12
9	country	24	new	12	provide	12
10	development	24	social	12	providing	12

UET						
	NOUNS		Adjectives		Verbs	
#	character	count	character	count	character	count
1	research	65	more	35	has	47
2	detail	52	various	20	Read	45
3	students	50	main	20	emailprotected	31
4	services	36	new	16	established	29
5	departments	33	local	16	provide	25
6	world	32	academic	14	following	20
7	development	29	Financial	13	Follow	17
8	engineering	28	Featured	13	related	14
9	centre	28	Extra	13	ACADEMICS	13
10	languages	28	curricular	13	Apply	13

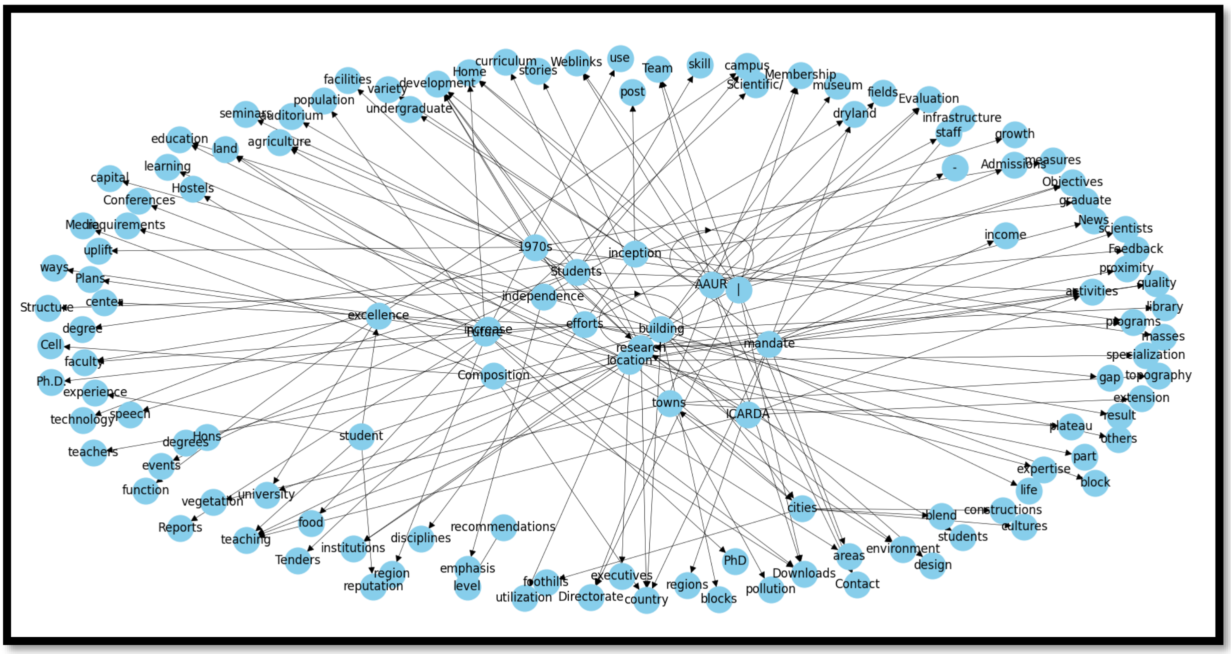
Comsats						
	NOUNS		Adjectives		Verbs	
#	character	count	character	count	character	count
1	STATUTES	35	academic	30	Posted	75
2	students	25	More	30	Walk	35
3	environment	20	Close	11	Apply	22
4	Invitation	20	social	10	feel	10
5	information	20	following	6	Read	10
6	TERMS	20	specific	6	become	10
7	CONDITIONS	20	AutoEventWireup="true	6	provides	10
8	SERVICE	20	Virtual	5	given	10
9	APPOINTMENT	20	toApply	5	study	10
10	Line	18	honored	5	occurred	6

Code for part 2

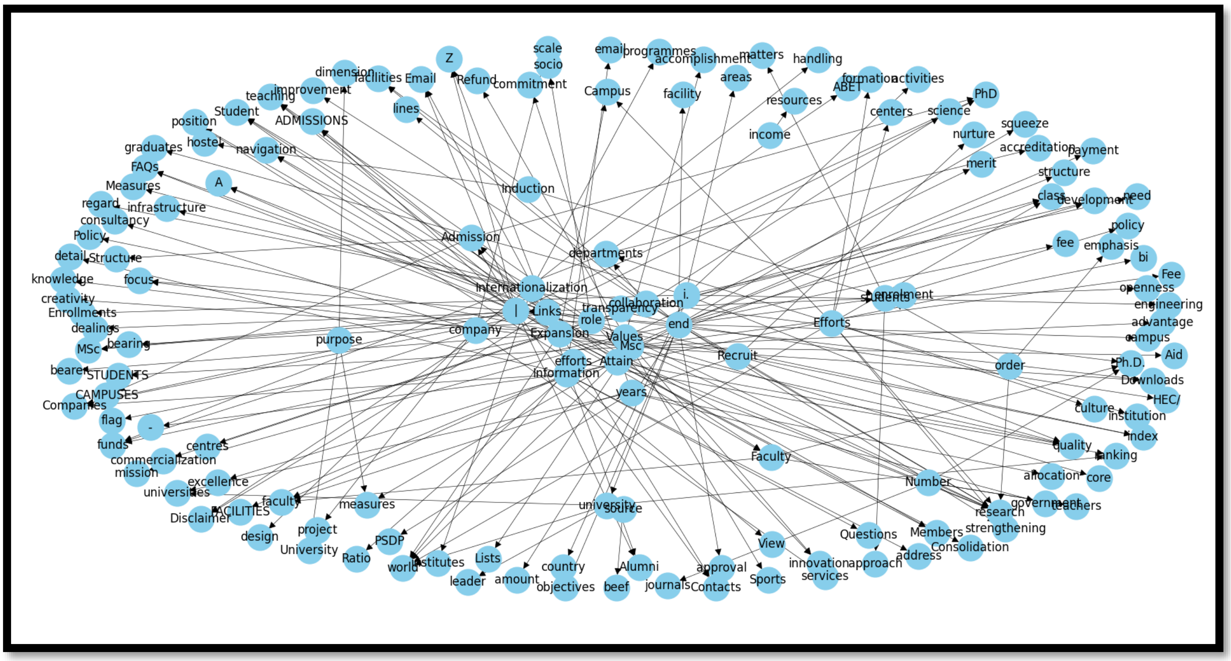
```
#spacy.cli.download("en_core_web_sm")
nlp = spacy.load("en_core_web_sm")
print()
ans=True
while ans:
    print ("""
    1.Arid University
    2.Uet University
    3.Comsat University
    4.Exit/Quit
    """)
    ans=input("Choice University Website to read : ")
    if ans=="1":
        filepath = "Uaar.txt"
        break
    elif ans=="2":
        filepath = "uet.txt"
        break
    elif ans=="3":
        filepath = "comsats.txt"
        break
    elif ans=="4":
        exit()
    elif ans !="":
      print("\n Not Valid Choice Try again")


text = open(filepath, encoding='utf-8', errors='ignore').read()
#print (text)
document = nlp(text)

nouns = []
adjectives = []
verbs = []

for token in document:
    if (token.pos_ == "NOUN"):
        if(token.text!="%"):
            nouns.append(token.text)
    if token.pos_ == "ADJ":
         adjectives.append(token.text)
    if token.pos_ == "VERB":
         verbs.append(token.text)


nouns_tally = Counter(nouns)
adjectives_tally = Counter(adjectives)
verbs_tally = Counter(verbs)

print("---------------NOUNS---------------")
NounsData = pd.DataFrame(nouns_tally.most_common(), columns=['character', 'count'])
print(NounsData)

print("---------------Adjectives---------------")
AdjectiveData = pd.DataFrame(adjectives_tally.most_common(), columns=['character',
'count'])
print(AdjectiveData)

print("---------------Verbs---------------")
VerbsData = pd.DataFrame(verbs_tally.most_common(), columns=['character', 'count'])
print(VerbsData)
```

Graph of all the nouns and nodes (Step 3)

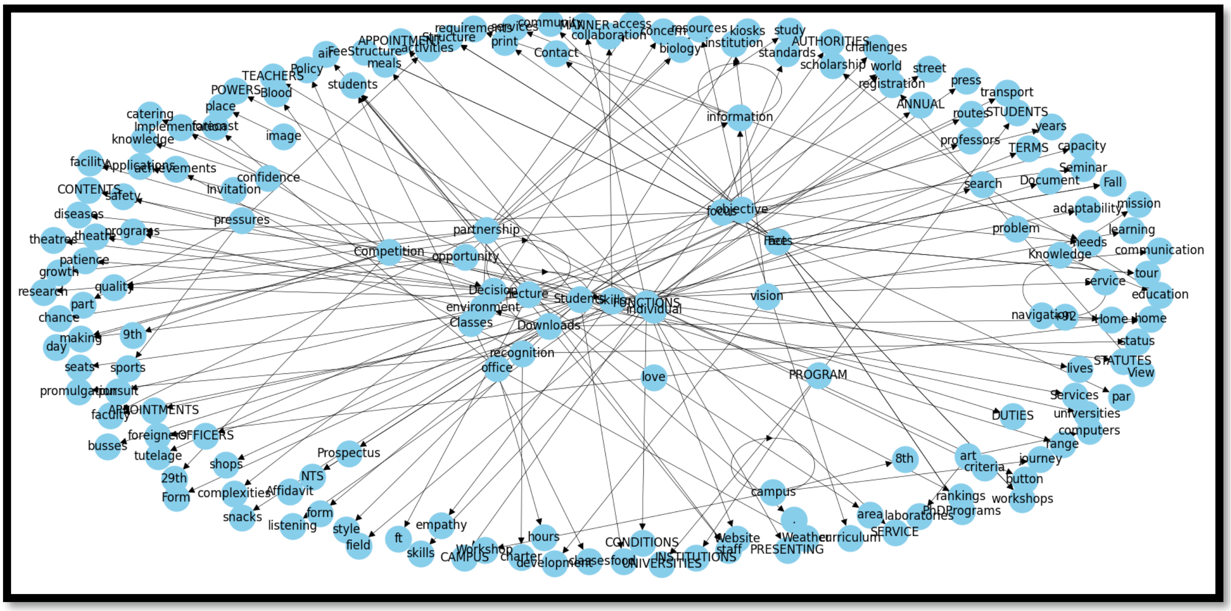
Graphs of All searches



Arid University



UET



Comsats

Number of Connected components and total nodes

Arid University

Number of Connected Components: 156

Total Nodes: 127

UET

Number of Connected Components: 189

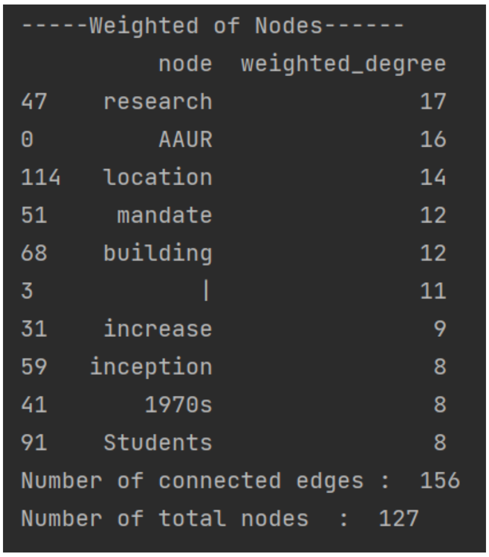
Total Nodes: 153

COMSATS

Number of Connected Components: 175

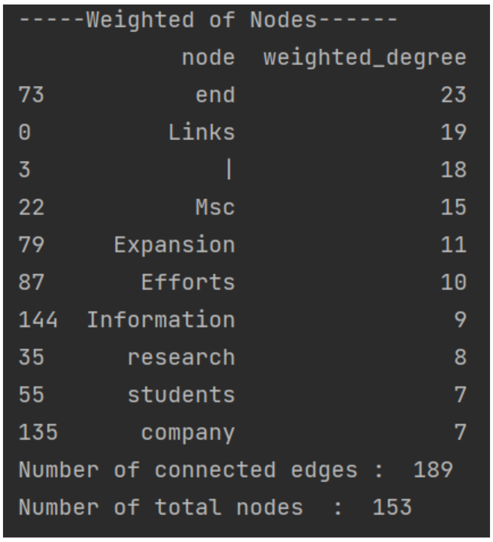
Total Nodes: 171

Top 10 Nouns with highest degree and their degree value



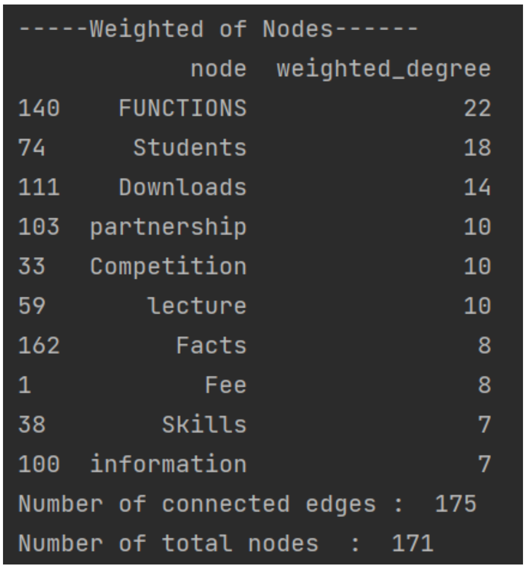
```
-----Weighted of Nodes------
          node  weighted_degree
47    research               17
0         AAUR               16
114   location               14
51     mandate               12
68    building               12
3            |               11
31    increase                9
59   inception                8
41       1970s                8
91    Students                8
Number of connected edges :  156
Number of total nodes  :  127
```

ARID University



```
-----Weighted of Nodes------
            node  weighted_degree
73           end               23
0          Links               19
3              |               18
22           Msc               15
79     Expansion               11
87       Efforts               10
144  Information                9
35      research                8
55      students                7
135      company                7
Number of connected edges :  189
Number of total nodes  :  153
```

UET



```
-----Weighted of Nodes------
            node  weighted_degree
140    FUNCTIONS               22
74      Students               18
111    Downloads               14
103  partnership               10
33   Competition               10
59       lecture               10
162        Facts                8
1            Fee                8
38        Skills                7
100  information                7
Number of connected edges :  175
Number of total nodes  :  171
```

COMSATS

All nouns within 5 words from the noun “quality”

-----5 words near "Quality"------ for Arid University

scientists
infrastructure
teaching
research
development
education
teachers
executives
fields
specialization
scientists
infrastructure
teaching
research

development
education
teachers
executives
fields
specialization
scientists
infrastructure
teaching
research
development
education

teachers
executives
fields
specialization
scientists
infrastructure
teaching
research
development
education
teachers
executives
fields
specializatio

5 words near "Quality"------for UET

excellence
teaching
research
transparency
openness

faculty
university
leader
ranking
world

end
objectives
focus
areas
university

teaching
research
position
world
class

excellence
teaching
research
transparency
openness

faculty
university
leader
ranking
world

end
objectives
focus
areas
university

teaching
research
position
world
class

excellence
teaching
research
transparency
openness

faculty
university

leader
ranking
world

end
objectives
focus
areas
university

teaching
research
position
world
class

excellence
teaching
research
transparency

openness

faculty
university
leader
ranking
world

end
objectives
focus
areas
university

teaching
research
position
world
class

-----5 words near "Quality"------for COMSATS

snacks
meals
universities
rankings
Knowledge

information
resources

environment
study
growth

snacks
meals
universities
rankings

Knowledge

information
resources
environment
study
growth

snacks

meals

universities

rankings

Knowledge

information

resources

environment

study

growth

snacks

meals

universities

rankings

Knowledge

information

resources

environment

study

growth

snacks

meals

universities

rankings

Knowledge

information

resources

environment

study

growth

code for part 3

```
Nouns_edges=[]

for sent_i,sent in enumerate(document.sents):
    nouns=[]
    for token in sent:
        if (token.pos_ == "NOUN"):
            if(token.text!="%"):
                nouns.append(token.text)
    Nouns_edges.append(nouns)

for i in Nouns_edges:
    if not i:
        Nouns_edges.remove(i)
# for i in Nouns_edges:
#     print(i)

g=nx.DiGraph()
g.add_nodes_from(nouns)
edge = []
for i in Nouns_edges:
    for j in i[1:]:
        edge.append([i[0],j])

g.add_edges_from(edge)



#nx.draw(g)


print("-----5 words near \"Quality\"------")
num=0
for sent_i,sent in enumerate(document.sents):
    nouns=[]
    for token in sent:
        if (token.pos_ == "NOUN"):
            if(token.text!="%"):
                num=num-1
            if(token.text=='quality'):
                num=6
                print("\n")
            elif(num>0):
                print(token.text)

print("-----Weighted of Nodes------")
nx.degree(g, weight='Weight')
weighted_degrees = dict(networkx.degree(g, weight='Weight'))
networkx.set_node_attributes(g, name='weighted_degree', values=weighted_degrees)

weighted_degree_df = pd.DataFrame(g.nodes(data='weighted_degree'), columns=['node',
'weighted_degree'])
weighted_degree_df = weighted_degree_df.sort_values(by='weighted_degree',
ascending=False)
print(weighted_degree_df[:10])

print("Number of connected edges : ",g.number_of_edges())
print("Number of total nodes  : ",g.number_of_nodes())

plt.figure(figsize=(8,8))
nx.draw(g, with_labels=True, node_color='skyblue', width=.3, font_size=8)

plt.draw()
plt.show()
```

Challenges Faced

- Inexperience in Python was one of the biggest challenges we faced. As we all have been studying C++, the syntax of it is very different of that from python.
- The libraries to use were another challenge as we already lacked behind in the language itself.
- Selecting the right website to scrap as many websites do not allow scrapping.
- Searching for websites that contained the word "Quality".
- Using list of nouns to create edges for the graph.
- One of our member was not available due to unfortunate, so we were not able to record a short demo explaining and created this graph

References

Prof. Arshad Allam's video link: https://youtu.be/PPSfrEanRFk

Link given with Project: https://melaniewalsh.github.io/Intro-Cultural-Analytics/welcome.html