Web Mining Lab - 2

Aadhitya Swarnesh

[redacted]

- 15 February 2021

Question - 1

Collect any 10 documents (English text documents) from the web and create inverted index by doing necessary preprocessing steps using python.

We use the BeautifulSoup library of Python in order to parse through the websites.

We use the Requests library of Python in order to make the web page requests.

We use the RE library of Python for pattern matching irrespective of the case or capitalisation of the content in the website.

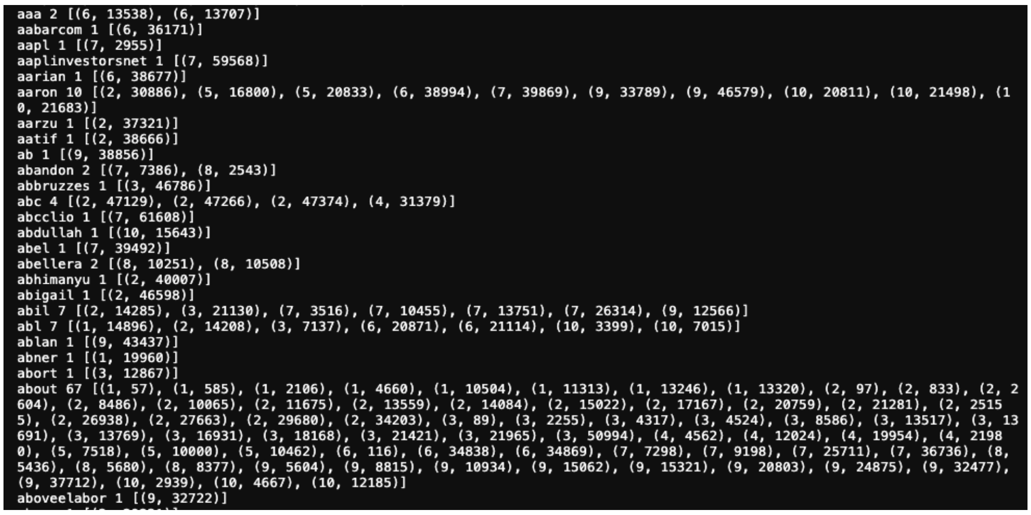
We use the NLTK library of python to perform Stemming and other NLP tasks before making predictions

The code for the program is as follows :

Code :

```
import requests
from bs4 import BeautifulSoup
from bs4.element import Comment
import re
import string
import nltk
from nltk.stem.porter import PorterStemmer

index = {}

def visible_text(element):
  if element.parent.name in ['style', 'title', 'script', 'head', '[document]', 'class',
'a', 'li']:
    return False
  elif isinstance(element, Comment):
    return False
  elif re.match(r"[\s\r\n]+",str(element)):
    return False
  return True

def read_url(url, url_number):

  ps = PorterStemmer()

  r = requests.get(url)
  soup = BeautifulSoup(r.content, 'html.parser')
  text = soup.findAll(text = True)
  result = list(filter(visible_text, text))
  counter = 0;
  words = []
  for i in result:
    temp = i.split(' ')
    for word in temp:
      k = []
      temp_word = word.lower()
      for c in temp_word:
        if c not in list(string.punctuation):
          k.append(c)
      temp_word = ''.join(k)
      words.append(temp_word)
  for i in words:
    if(i.isalpha()):
      i = ps.stem(i)
      if not i in index.keys():
        index[i] = [(url_number, counter)]
        counter = counter + len(i) + 1
      else:
```

```
      index[i].append((url_number, counter))
      counter = counter + len(i) + 1

  return None

urls = [
   'https://en.wikipedia.org/wiki/Google',
   'https://en.wikipedia.org/wiki/Facebook',
   'https://en.wikipedia.org/wiki/Netflix',
   'https://en.wikipedia.org/wiki/Amazon_(company)',
   'https://en.wikipedia.org/wiki/Microsoft',
   'https://en.wikipedia.org/wiki/Tesla,_Inc.',
   'https://en.wikipedia.org/wiki/Apple_Inc.',
   'https://en.wikipedia.org/wiki/Silicon_Valley_(TV_series)',
   'https://en.wikipedia.org/wiki/Wikipedia',
   'https://en.wikipedia.org/wiki/Uber',
]
for i in range(len(urls)) :
   read_url(urls[i], (i+1))


sorted_keys = sorted(index.keys())

f = open("output2.txt", "w")
output_line = "Word".ljust(15) + "Frequency".ljust(15) + "Posting
List".ljust(15) + "\n"
f.writelines(output_line)
for i in sorted_keys:
  print(i, len(index[i]), index[i])
  output_string = str(i).ljust(15) + str(len(index[i])).ljust(15) +
str(index[i]).ljust(15) + "\n"
  f.writelines(output_string)
  f.writelines('\n')

f.close()
```



```
aaa 2 [(6, 13538), (6, 13707)]
aabarcom 1 [(6, 36171)]
aapl 1 [(7, 2955)]
aaplinvestorsnet 1 [(7, 59568)]
aarian 1 [(6, 38677)]
aaron 10 [(2, 30886), (5, 16800), (5, 20833), (6, 38994), (7, 39869), (9, 33789), (9, 46579), (10, 20811), (10, 21498), (1
0, 21683)]
aarzu 1 [(2, 37321)]
aatif 1 [(2, 38666)]
ab 1 [(9, 38856)]
abandon 2 [(7, 7386), (8, 2543)]
abbruzzes 1 [(3, 46786)]
abc 4 [(2, 47129), (2, 47266), (2, 47374), (4, 31379)]
abcclio 1 [(7, 61608)]
abdullah 1 [(10, 15643)]
abel 1 [(7, 39492)]
abellera 2 [(8, 10251), (8, 10508)]
abhimanyu 1 [(2, 40007)]
abigail 1 [(2, 46598)]
abil 7 [(2, 14285), (3, 21130), (7, 3516), (7, 10455), (7, 13751), (7, 26314), (9, 12566)]
abl 7 [(1, 14896), (2, 14208), (3, 7137), (6, 20871), (6, 21114), (10, 3399), (10, 7015)]
ablan 1 [(9, 43437)]
abner 1 [(1, 19960)]
abort 1 [(3, 12867)]
about 67 [(1, 57), (1, 585), (1, 2106), (1, 4660), (1, 10504), (1, 11313), (1, 13246), (1, 13320), (2, 97), (2, 833), (2, 2
604), (2, 8486), (2, 10065), (2, 11675), (2, 13559), (2, 14084), (2, 15022), (2, 17167), (2, 20759), (2, 21281), (2, 2515
5), (2, 26938), (2, 27663), (2, 29680), (2, 34203), (3, 89), (3, 2255), (3, 4317), (3, 4524), (3, 8586), (3, 13517), (3, 13
691), (3, 13769), (3, 16931), (3, 18168), (3, 21421), (3, 21965), (3, 50994), (4, 4562), (4, 12024), (4, 19954), (4, 2198
0), (5, 7518), (5, 10000), (5, 10462), (6, 116), (6, 34838), (6, 34869), (7, 7298), (7, 9198), (7, 25711), (7, 36736), (8,
5436), (8, 5680), (8, 8377), (9, 5604), (9, 8815), (9, 10934), (9, 15062), (9, 15321), (9, 20803), (9, 24875), (9, 32477),
(9, 37712), (10, 2939), (10, 4667), (10, 12185)]
aboveelabor 1 [(9, 32722)]
```