In [3]:

```
def DE_LOB():
    #!pip install google
    def k_google(company_name):
        from googlesearch import search
        count = 0
        try :
            from googlesearch import search
        except ImportError:
            print("Can't find module named 'google'")
        searchresult=[]
        for i in search(query=company_name,tld='co.in',lang='en',num=1,start=0,stop=1,
            #Here 'num' refers to the number or URLs that we need
            count += 1
            #print (count)
            #print(i + '\n')
            searchresult.append(i)
    #     gs = Gsearch_python(company_name)
    #     gs.Gsearch()
        return(searchresult)


    x=input('Company_Name:- ')
    z=x+ ' wikipedia'
    z1=x+ ' Industries'
    z2=x+ ' about-us'
    # lls=k_google(z)
    # lls2=lls+k_google(z1)
    # lls3=lls2+k_google(z2)
    lls=k_google(z)+k_google(z1)+k_google(z2)

    def k_description(x):
        #!pip install selenium
        import selenium
        from selenium import webdriver
        from selenium.webdriver.common.keys import Keys
        from selenium.webdriver.chrome.options import Options
        import time
        driver = webdriver.Chrome(executable_path= r"C:\chromedriver.exe")
        driver.minimize_window()
        driver.get(x)
        element= driver.find_element_by_css_selector('body')
        #print(element)
        time.sleep(8)
        element.send_keys(Keys.CONTROL+'a')
        time.sleep(2)
        element.send_keys(Keys.CONTROL+'c')
        #quit()
        #print("*******************************Copied*******************************")

    def LOB():
#         !pip install pandas
#         !pip install numpy
        import numpy as np
        import pandas as pd
        import nltk
        #nltk.download('punkt') # one time execution
        import re
        from nltk import punkt

        from nltk.tokenize import sent_tokenize
        sentences = []
        for s in df['desc']:
          sentences.append(sent_tokenize(s))

        sentences = [y for x in sentences for y in x] # flatten list

        clean_sentences = pd.Series(sentences).str.replace("[^a-zA-Z]", " ")
        clean_sentences = [s.lower() for s in clean_sentences]

        nltk.download('stopwords')

        from nltk.corpus import stopwords
        stop_words = stopwords.words('english')

        def remove_stopwords(sen):
            sen_new = " ".join([i for i in sen if i not in stop_words])
            return sen_new

        clean_sentences = [remove_stopwords(r.split()) for r in clean_sentences]
        import numpy as np
        word_embeddings = {}
        #f = open('glove.6B.100d.txt', encoding='utf-8')
        f = open(r"C:\Users\vin8.1\Files\AML\glove.6B.100d.txt", encoding='utf-8')
        #f = open(r"C:\Users\krish\Files\glove.840B.300d\glove.840B.300d.txt", encodin
        for line in f:
            values = line.split()
            word = values[0]
            coefs = np.asarray(values[1:], dtype='float32')
            word_embeddings[word] = coefs
        f.close()

        sentence_vectors = []
        for i in clean_sentences:
          if len(i) != 0:
            v = sum([word_embeddings.get(w, np.zeros((100,))) for w in i.split()])/(le
          else:
            v = np.zeros((100,))
          sentence_vectors.append(v)

        sim_mat = np.zeros([len(sentences), len(sentences)])

        from sklearn.metrics.pairwise import cosine_similarity

        for i in range(len(sentences)):
          for j in range(len(sentences)):
            if i != j:
              sim_mat[i][j] = cosine_similarity(sentence_vectors[i].reshape(1,100), se

        import networkx as nx
        nx_graph = nx.from_numpy_array(sim_mat)
        scores = nx.pagerank(nx_graph)

        ranked_sentences = sorted(((scores[i],s) for i,s in enumerate(sentences)), rev

    #     for i in range(75):
    #       print(ranked_sentences[i][1])

#         rsl=[]
#         for i in range(75):
#             rsl.append(ranked_sentences[i][1])
```

```
        #print(rsl)

        lsr=len(ranked_sentences)
        rsl=[]
        if lsr<75:
            for i in range(lsr):
                rsl.append(ranked_sentences[i][1])
        else:
            for i in range(75):
                rsl.append(ranked_sentences[i][1])
        rsllst = [x.upper() for x in rsl]
        #print(rsl)
        #print(rsllst)
        Repolist=[]
        #Repolist = open(r"C:\Users\vin8.1\Files\lob_repo.txt").read().splitlines()
        Repolist = open(r"C:\Users\vin8.1\Files\AML\lob_repo.txt").read().splitlines()
        #print(type(Repolist))
        #print(Repolist)

        rlst = [x.upper() for x in Repolist]
        #print(rlst)

        loblst=[]
        for string in rlst:
            for string2 in rsllst:
                if string in string2:
                    loblst.append(string)
        #print(loblst)

        def Remove_dup(list_name):
            final_list = []
            for x in list_name:
                if x not in final_list:
                    final_list.append(x)
            return final_list
    #     LOB=Remove_dup(loblst)
    #     print (LOB)
        LOB.lblst=Remove_dup(loblst)

    #!pip install pyperclip
    import pyperclip
    out_list=[]
    for i in lls:
        print(i)
        #print("Click the Link to navigate to the Webpage: "+i)

        k_description(i)

        s = pyperclip.paste()
        #print(s)

        temp="text"+str(lls.index(i))+".txt"
          #Creating multiple text files(each text file for each URL)
        with open(temp,'w',encoding="utf-8") as g:
            g.write(s)

        filepath="text"+str(lls.index(i))+".txt"
        temp2="textout"+str(lls.index(i))+".txt"
        with open(filepath, encoding="utf8") as infile, open(temp2, 'w',encoding="utf8
            for line in infile:
                line.strip(',')
```

```python
                if not line.strip(): continue  # skip the empty line
                outfile.write(line)

        try:
            from collections import OrderedDict
        except ImportError:
            from ordereddict import OrderedDict
        import pandas as pd
        colnames=['desc']
        temp3="textout"+str(lls.index(i))+".txt"
        df = pd.read_csv(temp3,names=colnames, header=None, encoding="utf8")

        LOB()
        URL=i
        LoB=LOB.lblst
        lb_out=[URL,LoB];
        #return lb_out
        out_list.append(lb_out)
    return out_list
    #print out_list
DE_LOB()
```

```
Company_Name:- cipla
https://en.wikipedia.org/wiki/Cipla (https://en.wikipedia.org/wiki/Cipla)

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\vin8.1\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

http://www.ciplaindustries.com/ (http://www.ciplaindustries.com/)

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\vin8.1\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

https://en.wikipedia.org/wiki/Cipla (https://en.wikipedia.org/wiki/Cipla)

[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\vin8.1\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[3]:

```
[['https://en.wikipedia.org/wiki/Cipla',
  ['PERSONAL CARE PRODUCTS',
   'BIOTECHNOLOGY COMPANY',
   'HEALTHCARE',
   'PERSONAL CARE',
   'INSURANCE',
   'MANUFACTURING']],
 ['http://www.ciplaindustries.com/', ['INDUSTRIES', 'RETAIL']],
 ['https://en.wikipedia.org/wiki/Cipla',
  ['PERSONAL CARE PRODUCTS',
   'BIOTECHNOLOGY COMPANY',
   'HEALTHCARE',
   'PERSONAL CARE',
   'INSURANCE',
   'MANUFACTURING']]]
```