Hiring trends of US Banks

Authors

Akshata Honrao, Northeastern University
Nancy Packiyanathan, Northeastern University
Rushali Udhani, Northeastern University
Sonal Thube, Northeastern University

February 8, 2019

Overview

The services provided by Financial institutions in the US are changing rapidly. Changing business models and the technological revolution has fueled the growth of a new breed of financial products and services collectively known as "Fintech".

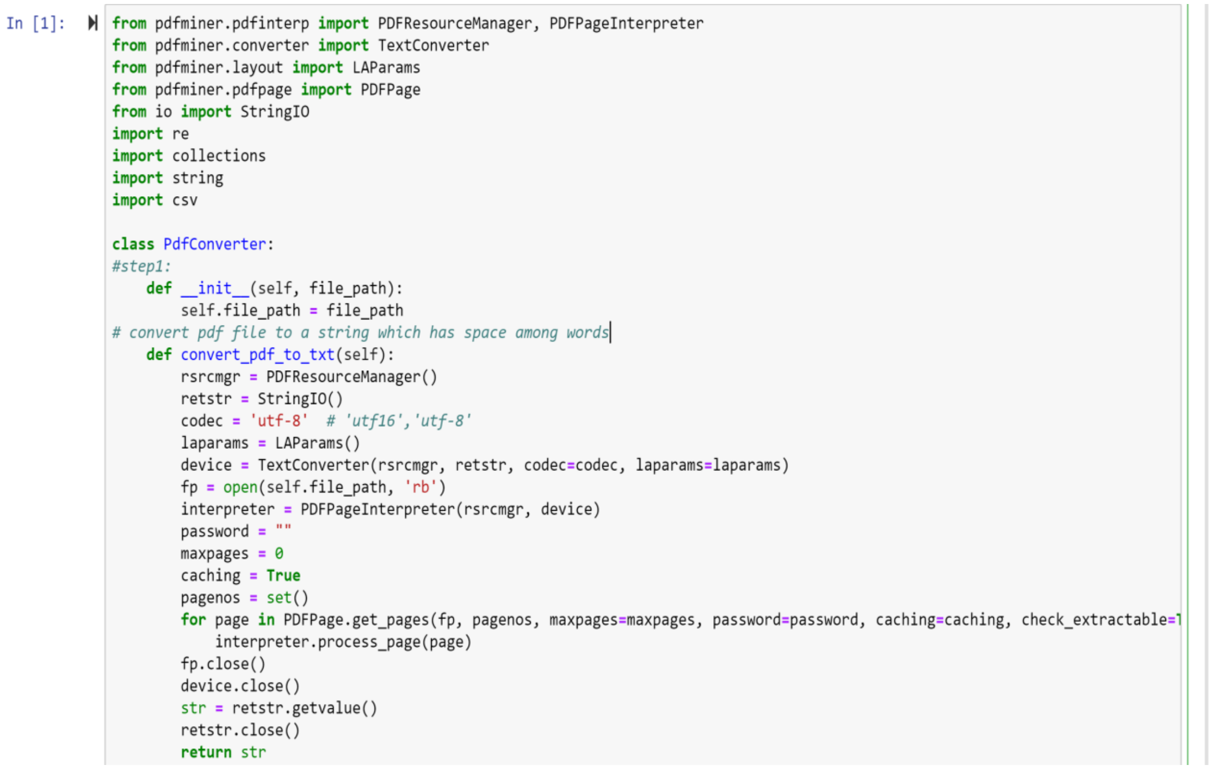
With changing demographics, automation efforts and demand for new products and services, large financial institutions are realizing the power of technologies like data science, AI, cloud technologies and machine learning and are heavily investing to upgrade their technological platforms to cater to the upcoming revolution. Things are fast evolving and as we enter 2019, it is interesting to understand the hiring trends in the top financial institutions in the US.

Our goal in this case study is to conduct a study on the job openings in the top US Banks in the United States and analyze trends in the industry particularly in the area of Fintech.

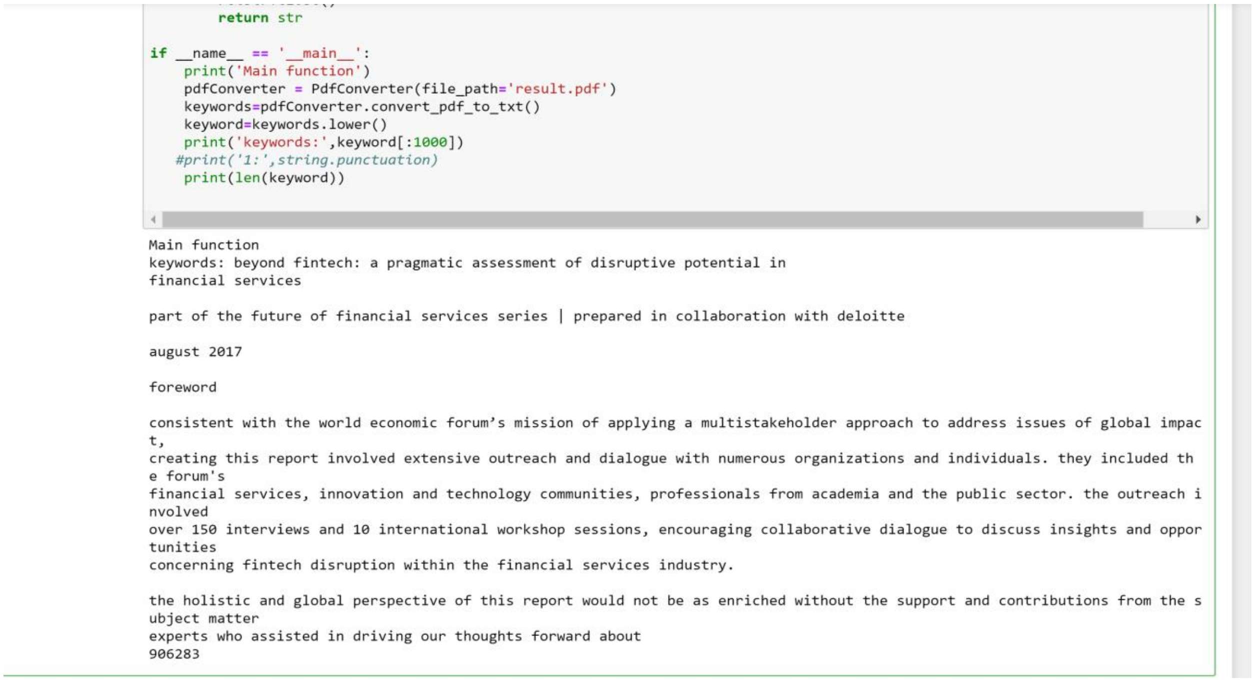
Part One- Frequently used keywords in Fintech

Our first objective is to understand the keywords that are typically used in Fintech. For this, We used the four reports provided by World Economic forum (WEF).

We extracted the keywords from these documents to build a dictionary of top keywords that could be used to describe the fintech space. We did this using **PDFMinersix.**



```
In [1]:  from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
         from pdfminer.converter import TextConverter
         from pdfminer.layout import LAParams
         from pdfminer.pdfpage import PDFPage
         from io import StringIO
         import re
         import collections
         import string
         import csv

         class PdfConverter:
         #step1:
             def __init__(self, file_path):
                 self.file_path = file_path
         # convert pdf file to a string which has space among words
             def convert_pdf_to_txt(self):
                 rsrcmgr = PDFResourceManager()
                 retstr = StringIO()
                 codec = 'utf-8'  # 'utf16','utf-8'
                 laparams = LAParams()
                 device = TextConverter(rsrcmgr, retstr, codec=codec, laparams=laparams)
                 fp = open(self.file_path, 'rb')
                 interpreter = PDFPageInterpreter(rsrcmgr, device)
                 password = ""
                 maxpages = 0
                 caching = True
                 pagenos = set()
                 for page in PDFPage.get_pages(fp, pagenos, maxpages=maxpages, password=password, caching=caching, check_extractable=T
                     interpreter.process_page(page)
                 fp.close()
                 device.close()
                 str = retstr.getvalue()
                 retstr.close()
                 return str
```



```
        return str

if __name__ == '__main__':
    print('Main function')
    pdfConverter = PdfConverter(file_path='result.pdf')
    keywords=pdfConverter.convert_pdf_to_txt()
    keyword=keywords.lower()
    print('keywords:',keyword[:1000])
   #print('1:',string.punctuation)
    print(len(keyword))
```

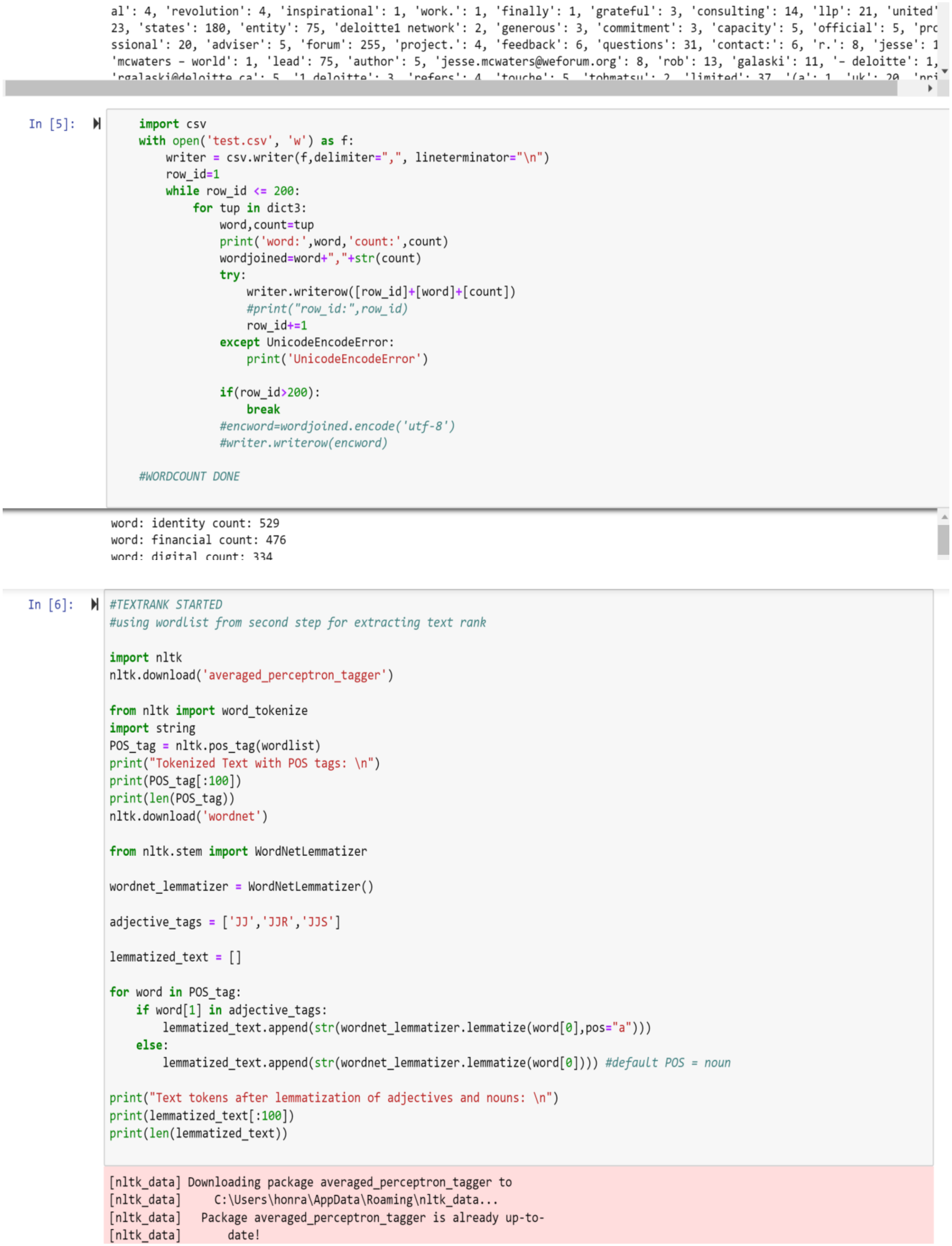
```
Main function
keywords: beyond fintech: a pragmatic assessment of disruptive potential in
financial services

part of the future of financial services series | prepared in collaboration with deloitte

august 2017

foreword

consistent with the world economic forum's mission of applying a multistakeholder approach to address issues of global impac
t,
creating this report involved extensive outreach and dialogue with numerous organizations and individuals. they included th
e forum's
financial services, innovation and technology communities, professionals from academia and the public sector. the outreach i
nvolved
over 150 interviews and 10 international workshop sessions, encouraging collaborative dialogue to discuss insights and oppor
tunities
concerning fintech disruption within the financial services industry.

the holistic and global perspective of this report would not be as enriched without the support and contributions from the s
ubject matter
experts who assisted in driving our thoughts forward about
906283
```

We extracted the top 100 keywords and built three lists using the following approaches:

a. Wordcount

b. TF/IDF - Term Frequency/ Inverse Document Frequency

c. TextRank



```
al': 4, 'revolution': 4, 'inspirational': 1, 'work.': 1, 'finally': 1, 'grateful': 3, 'consulting': 14, 'llp': 21, 'united'
23, 'states': 180, 'entity': 75, 'deloitte1 network': 2, 'generous': 3, 'commitment': 3, 'capacity': 5, 'official': 5, 'pro
ssional': 20, 'adviser': 5, 'forum': 255, 'project.': 4, 'feedback': 6, 'questions': 31, 'contact:': 6, 'r.': 8, 'jesse': 1
'mcwaters - world': 1, 'lead': 75, 'author': 5, 'jesse.mcwaters@weforum.org': 8, 'rob': 13, 'galaski': 11, '- deloitte': 1,
'rgalaski@deloitte.ca': 5, '1 deloitte': 3, 'refers': 4, 'touche': 5, 'tohmatsu': 2, 'limited': 37, '(a': 1, 'uk': 20, 'pri
```

In [5]:

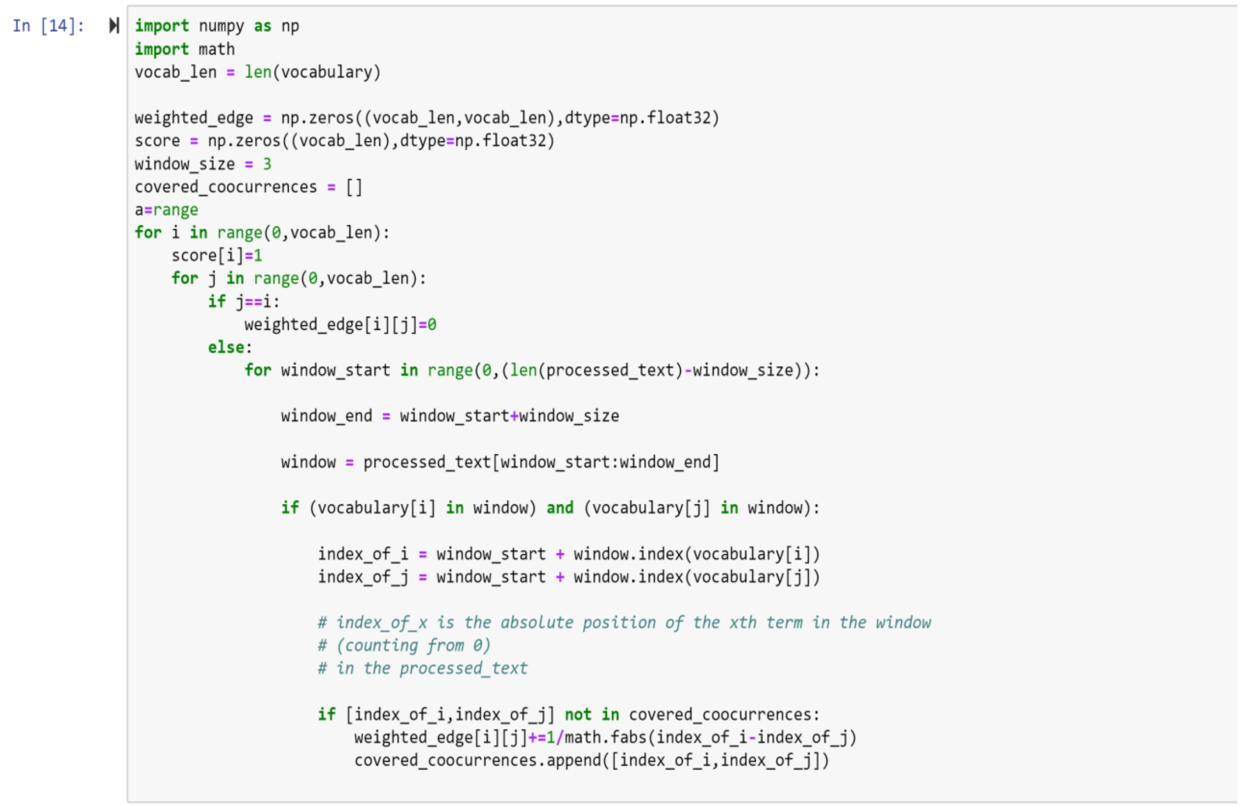
```python
import csv
with open('test.csv', 'w') as f:
    writer = csv.writer(f,delimiter=",", lineterminator="\n")
    row_id=1
    while row_id <= 200:
        for tup in dict3:
            word,count=tup
            print('word:',word,'count:',count)
            wordjoined=word+","+str(count)
            try:
                writer.writerow([row_id]+[word]+[count])
                #print("row_id:",row_id)
                row_id+=1
            except UnicodeEncodeError:
                print('UnicodeEncodeError')

            if(row_id>200):
                break
            #encword=wordjoined.encode('utf-8')
            #writer.writerow(encword)

#WORDCOUNT DONE
```

```
word: identity count: 529
word: financial count: 476
word: digital count: 334
```

In [6]:

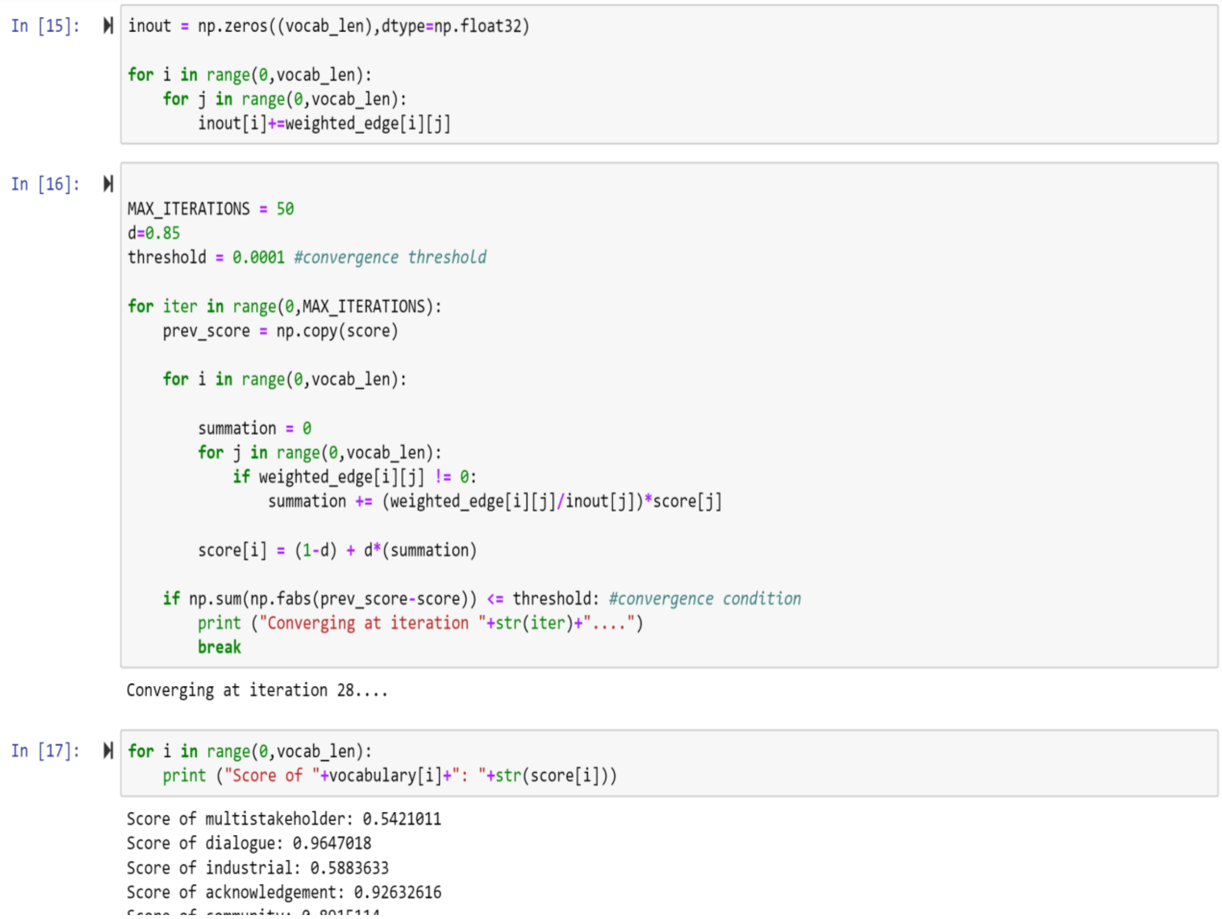
```python
#TEXTRANK STARTED
#using wordlist from second step for extracting text rank

import nltk
nltk.download('averaged_perceptron_tagger')

from nltk import word_tokenize
import string
POS_tag = nltk.pos_tag(wordlist)
print("Tokenized Text with POS tags: \n")
print(POS_tag[:100])
print(len(POS_tag))
nltk.download('wordnet')

from nltk.stem import WordNetLemmatizer

wordnet_lemmatizer = WordNetLemmatizer()

adjective_tags = ['JJ','JJR','JJS']

lemmatized_text = []

for word in POS_tag:
    if word[1] in adjective_tags:
        lemmatized_text.append(str(wordnet_lemmatizer.lemmatize(word[0],pos="a")))
    else:
        lemmatized_text.append(str(wordnet_lemmatizer.lemmatize(word[0]))) #default POS = noun

print("Text tokens after lemmatization of adjectives and nouns: \n")
print(lemmatized_text[:100])
print(len(lemmatized_text))
```

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



In [14]:

```
import numpy as np
import math
vocab_len = len(vocabulary)

weighted_edge = np.zeros((vocab_len,vocab_len),dtype=np.float32)
score = np.zeros((vocab_len),dtype=np.float32)
window_size = 3
covered_coocurrences = []
a=range
for i in range(0,vocab_len):
    score[i]=1
    for j in range(0,vocab_len):
        if j==i:
            weighted_edge[i][j]=0
        else:
            for window_start in range(0,(len(processed_text)-window_size)):

                window_end = window_start+window_size

                window = processed_text[window_start:window_end]

                if (vocabulary[i] in window) and (vocabulary[j] in window):

                    index_of_i = window_start + window.index(vocabulary[i])
                    index_of_j = window_start + window.index(vocabulary[j])

                    # index_of_x is the absolute position of the xth term in the window
                    # (counting from 0)
                    # in the processed_text

                    if [index_of_i,index_of_j] not in covered_coocurrences:
                        weighted_edge[i][j]+=1/math.fabs(index_of_i-index_of_j)
                        covered_coocurrences.append([index_of_i,index_of_j])
```



In [15]:

```
inout = np.zeros((vocab_len),dtype=np.float32)

for i in range(0,vocab_len):
    for j in range(0,vocab_len):
        inout[i]+=weighted_edge[i][j]
```

In [16]:

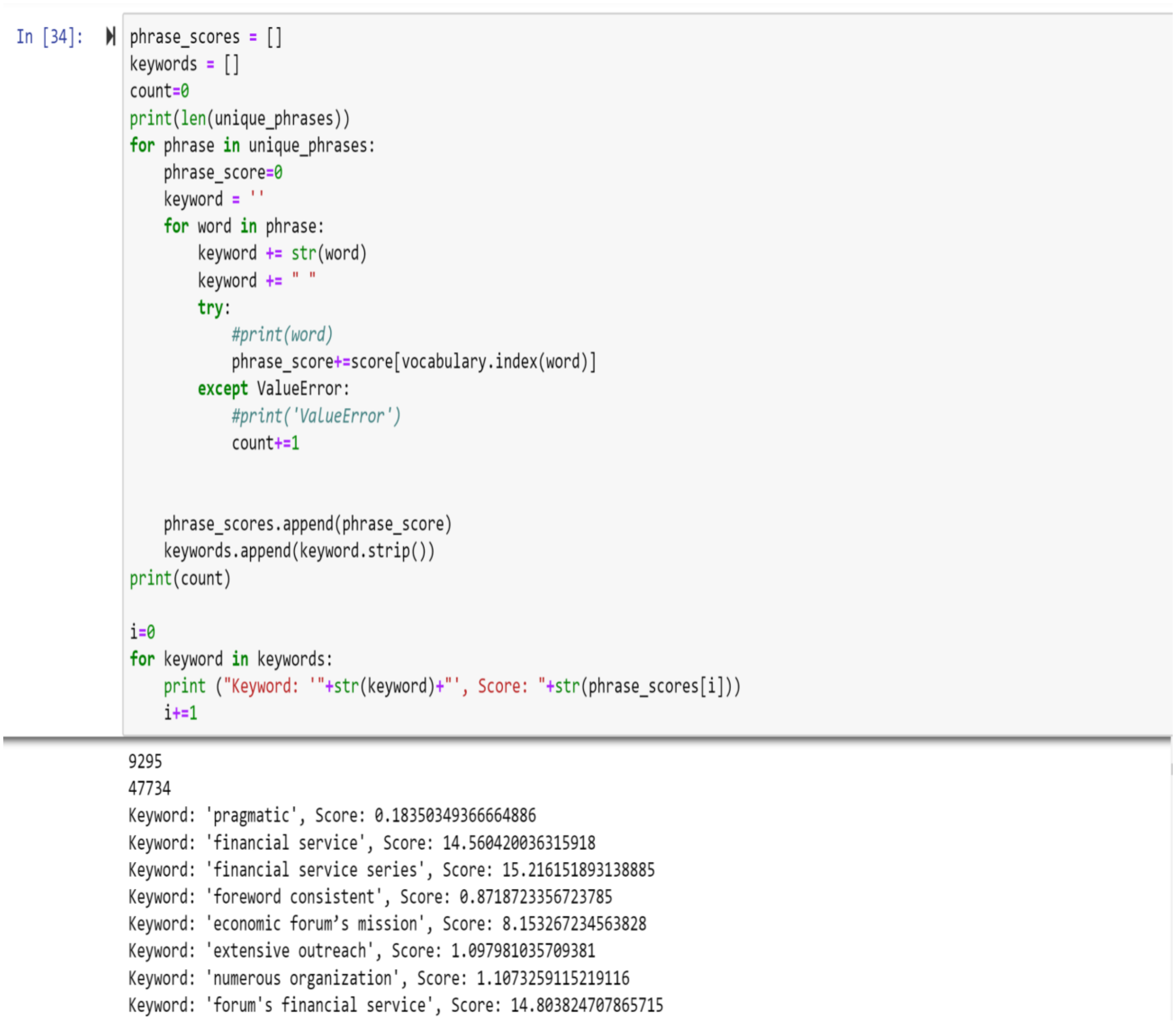
```

MAX_ITERATIONS = 50
d=0.85
threshold = 0.0001 #convergence threshold

for iter in range(0,MAX_ITERATIONS):
    prev_score = np.copy(score)

    for i in range(0,vocab_len):

        summation = 0
        for j in range(0,vocab_len):
            if weighted_edge[i][j] != 0:
                summation += (weighted_edge[i][j]/inout[j])*score[j]

        score[i] = (1-d) + d*(summation)

    if np.sum(np.fabs(prev_score-score)) <= threshold: #convergence condition
        print ("Converging at iteration "+str(iter)+"....")
        break
```

```
Converging at iteration 28....
```

In [17]:

```
for i in range(0,vocab_len):
    print ("Score of "+vocabulary[i]+": "+str(score[i]))
```

```
Score of multistakeholder: 0.5421011
Score of dialogue: 0.9647018
Score of industrial: 0.5883633
Score of acknowledgement: 0.92632616
Score of community: 0.8915114
```



```python
phrase_scores = []
keywords = []
count=0
print(len(unique_phrases))
for phrase in unique_phrases:
    phrase_score=0
    keyword = ''
    for word in phrase:
        keyword += str(word)
        keyword += " "
        try:
            #print(word)
            phrase_score+=score[vocabulary.index(word)]
        except ValueError:
            #print('ValueError')
            count+=1


    phrase_scores.append(phrase_score)
    keywords.append(keyword.strip())
print(count)

i=0
for keyword in keywords:
    print ("Keyword: '"+str(keyword)+"', Score: "+str(phrase_scores[i]))
    i+=1
```

```
9295
47734
Keyword: 'pragmatic', Score: 0.18350349366664886
Keyword: 'financial service', Score: 14.560420036315918
Keyword: 'financial service series', Score: 15.216151893138885
Keyword: 'foreword consistent', Score: 0.8718723356723785
Keyword: 'economic forum’s mission', Score: 8.153267234563828
Keyword: 'extensive outreach', Score: 1.097981035709381
Keyword: 'numerous organization', Score: 1.1073259115219116
Keyword: 'forum's financial service', Score: 14.803824707865715
```

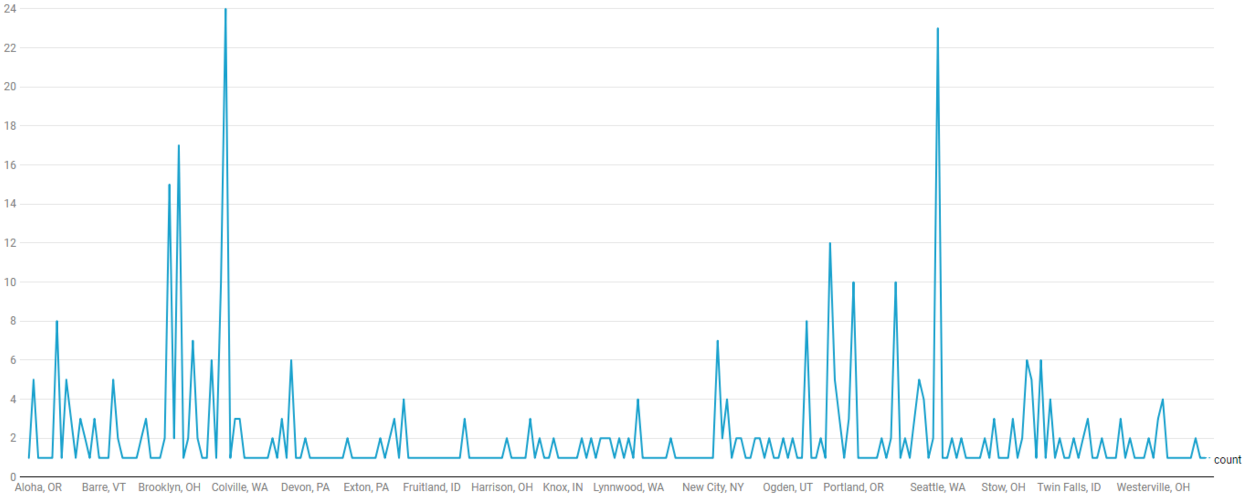
Part two- Web Scraping of data from two US Bank Websites

We scraped the hiring data from top two US Banks- KeyCorp, Fifth Third Corp using **BeautifulSoup4 and Selenium.**

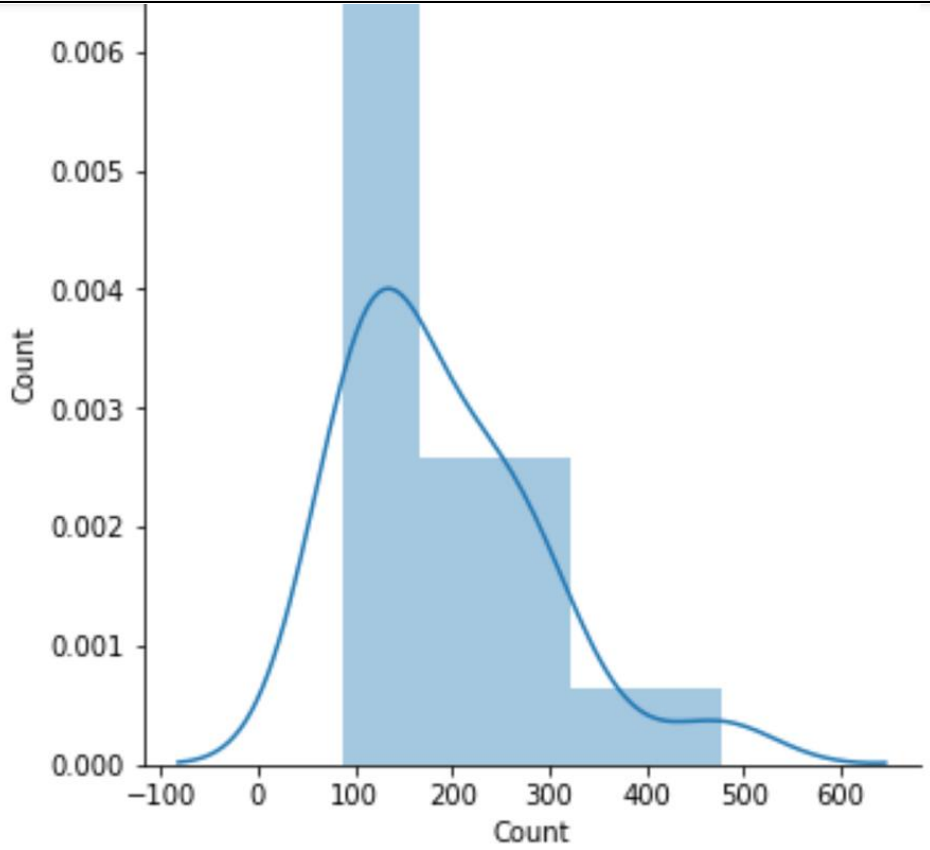
Part three- Analysis

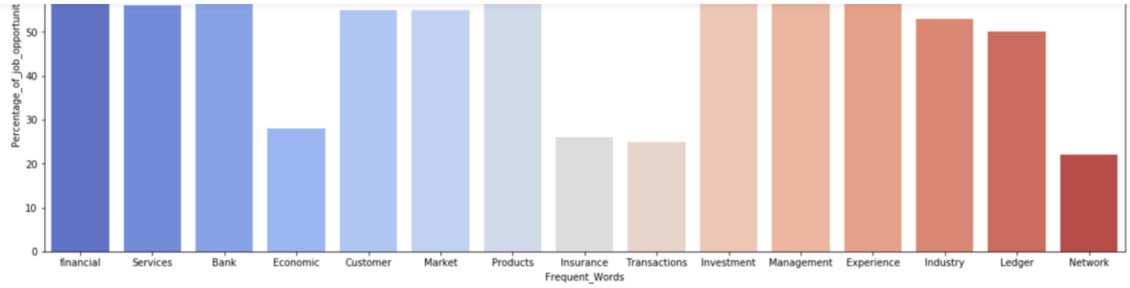
We analyzed this data and drew insights on hiring patterns and trends.

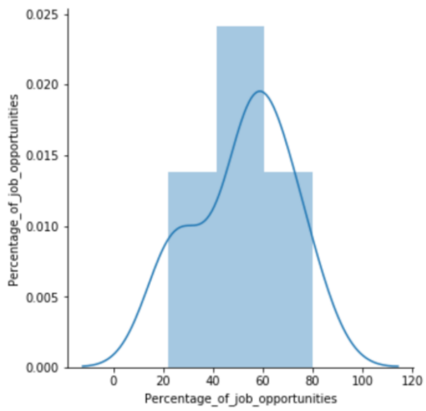
Job Count Grouped by Location - Keycorp Bank

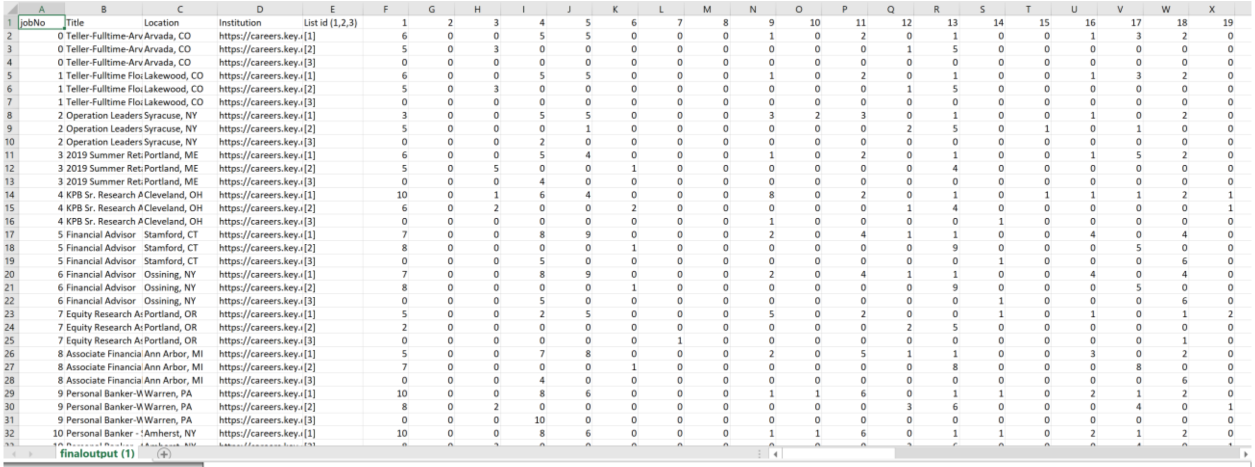


Get the data • Created with Datawrapper



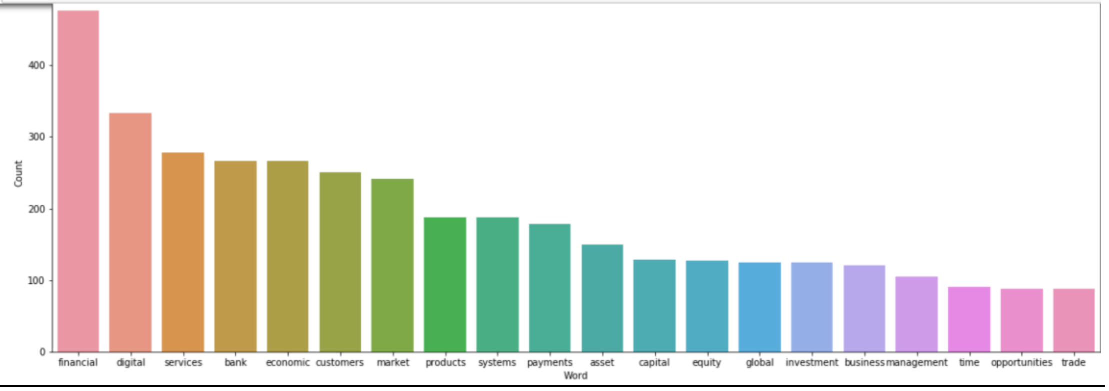






	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	jobNo	Title	Location	Institution	List id (1,2,3)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
2	0	Teller-Fulltime-Arv	Arvada, CO	https://careers.key.ı	[1]	6	0	0	5	5	0	0	0	1	0	2	0	1	0	0	1	3	2	0
3	0	Teller-Fulltime-Arv	Arvada, CO	https://careers.key.ı	[2]	5	0	3	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0
4	0	Teller-Fulltime-Arv	Arvada, CO	https://careers.key.ı	[3]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	Teller-Fulltime Flo:	Lakewood, CO	https://careers.key.ı	[1]	6	0	0	5	5	0	0	0	1	0	2	0	1	0	0	1	3	2	0
6	1	Teller-Fulltime Flo:	Lakewood, CO	https://careers.key.ı	[2]	5	0	3	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0
7	1	Teller-Fulltime Flo:	Lakewood, CO	https://careers.key.ı	[3]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	2	Operation Leaders	Syracuse, NY	https://careers.key.ı	[1]	3	0	0	5	5	0	0	0	3	2	3	0	1	0	0	1	0	2	0
9	2	Operation Leaders	Syracuse, NY	https://careers.key.ı	[2]	5	0	0	0	1	0	0	0	0	0	0	2	5	0	1	0	1	0	0
10	2	Operation Leaders	Syracuse, NY	https://careers.key.ı	[3]	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	3	2019 Summer Ret:	Portland, ME	https://careers.key.ı	[1]	6	0	0	5	4	0	0	0	1	0	2	0	1	0	0	1	5	2	0
12	3	2019 Summer Ret:	Portland, ME	https://careers.key.ı	[2]	5	0	5	0	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0
13	3	2019 Summer Ret:	Portland, ME	https://careers.key.ı	[3]	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	4	KPB Sr. Research A	Cleveland, OH	https://careers.key.ı	[1]	10	0	1	6	4	0	0	0	8	0	2	0	1	0	1	1	1	2	1
15	4	KPB Sr. Research A	Cleveland, OH	https://careers.key.ı	[2]	6	0	2	0	0	2	0	0	0	0	0	1	4	0	0	0	0	0	1
16	4	KPB Sr. Research A	Cleveland, OH	https://careers.key.ı	[3]	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
17	5	Financial Advisor	Stamford, CT	https://careers.key.ı	[1]	7	0	0	8	9	0	0	0	2	0	4	1	1	0	0	4	0	4	0
18	5	Financial Advisor	Stamford, CT	https://careers.key.ı	[2]	8	0	0	0	0	1	0	0	0	0	0	0	9	0	0	0	5	0	0
19	5	Financial Advisor	Stamford, CT	https://careers.key.ı	[3]	0	0	0	5	0	0	0	0	0	0	0	0	0	1	0	0	0	6	0
20	6	Financial Advisor	Ossining, NY	https://careers.key.ı	[1]	7	0	0	8	9	0	0	0	2	0	4	1	1	0	0	4	0	4	0
21	6	Financial Advisor	Ossining, NY	https://careers.key.ı	[2]	8	0	0	0	0	1	0	0	0	0	0	0	9	0	0	0	5	0	0
22	6	Financial Advisor	Ossining, NY	https://careers.key.ı	[3]	0	0	0	5	0	0	0	0	0	0	0	0	0	1	0	0	0	6	0
23	7	Equity Research As	Portland, OR	https://careers.key.ı	[1]	5	0	0	2	5	0	0	0	5	0	2	0	0	1	0	1	0	1	2
24	7	Equity Research As	Portland, OR	https://careers.key.ı	[2]	2	0	0	0	0	0	0	0	0	0	0	2	5	0	0	0	0	0	0
25	7	Equity Research As	Portland, OR	https://careers.key.ı	[3]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
26	8	Associate Financial	Ann Arbor, MI	https://careers.key.ı	[1]	5	0	0	7	8	0	0	0	2	0	5	1	1	0	0	3	0	2	0
27	8	Associate Financial	Ann Arbor, MI	https://careers.key.ı	[2]	7	0	0	0	0	1	0	0	0	0	0	0	8	0	0	0	8	0	0
28	8	Associate Financial	Ann Arbor, MI	https://careers.key.ı	[3]	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
29	9	Personal Banker-W	Warren, PA	https://careers.key.ı	[1]	10	0	0	8	6	0	0	0	1	1	6	0	1	1	0	2	1	2	0
30	9	Personal Banker-W	Warren, PA	https://careers.key.ı	[2]	8	0	2	0	0	0	0	0	0	0	0	3	6	0	0	0	4	0	1
31	9	Personal Banker-W	Warren, PA	https://careers.key.ı	[3]	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	10	Personal Banker - :	Amherst, NY	https://careers.key.ı	[1]	10	0	0	8	6	0	0	0	1	1	6	0	1	1	0	2	1	2	0

finaloutput (1)


Count
financial
digital
services
bank
economic
customers
market
products
systems
payments
asset
capital
equity
global
investment
business
management
time
opportunities
trade
Word