Project: Get SEC EDGAR Data

- Wesley, Ying (alphabetical)

Outline

- Download multiple types of TXT files from EDGAR database via API
- Extract, clean and reshape company data and file data separately from the files

Summary

The goal of the project is to get metadata and file's metadata of all US companies recorded in the EDGAR database from the SEC official website. The main work is to do the data cleaning with Python. Final tables, one for company data and the other for filing data, show the clean and tidy descriptions with multiple attributes. It is more straight-forward for people to observe than the original messy .txt data. In this project, the skills of data collection, structured and unstructured data cleaning, python coding will be practiced.

The workflow is simply:

get the EDGAR data → split into company metadata and file data → clean, transform and join into a table separately for company data and filing data → save the results

Input Data

The raw data is from EDGAR website, we use the *sec_edgar_downloader* package in Python to get the raw data. And the companies' unique CIK(CENTRAL INDEX KEY), which is needed in the function to get the data, is downloaded from 'Company' file on the EDGAR website (https://www.sec.gov/Archives/edgar/full-index/). The file types we deal with are: '8-K','10-K','10KSB','10-Q','13F-NT','13F-HR','SC 13G','SD' and 'S-1'.

According to the CIK, we have 17205 companies, each having 1-8 different files in different file types, available for data extraction (each file for different file types is the latest one).

Full File Path for Input and Other Files

- Input for downloading:
  companies with CIKs path: 'company.xlsx'
- Input for cleaning:
  downloaded original data path: 'data/sec_edgar_fillings/'

Example raw data

- SC 13G

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Proc-Type: 2001,MIC-CLEAR
Originator-Name: webmaster@www.sec.gov
Originator-Key-Asymmetric:
 MFgwCgYEVQgBAQICAf8DSgAwRwJAW2sNKK9AVtBzYZmr6aGjlWyK3XmZv3dTINen
 TWSM7vrzLADbmYQaionwg5sDW3P6oaM5D3tdezXMm7z1T+B+twIDAQAB
MIC-Info: RSA-MD5,RSA,
 P7FLFcT/be9zIXbcRRfanpf7M8Tz0+c6E/VMEMadb1A184208v6GzzMePnSFfqwZ
 jfzvama8MZJvm9SBD0KfpQ==

<SEC-DOCUMENT>0000950172-03-003584.txt : 20031222
<SEC-HEADER>0000950172-03-003584.hdr.sgml : 20031222
<ACCEPTANCE-DATETIME>20031222165228
ACCESSION NUMBER:               0000950172-03-003584
CONFORMED SUBMISSION TYPE:      SC 13G
PUBLIC DOCUMENT COUNT:          1
FILED AS OF DATE:               20031222

FILED BY:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 CASSA DEPOSITI E PRESTITI SOCIETA PER AZIONI
                CENTRAL INDEX KEY:                      0001274069
                IRS NUMBER:                             000000000

        FILING VALUES:
                FORM TYPE:              SC 13G

        BUSINESS ADDRESS:
                BUSINESS PHONE:         39 06 42211

        FORMER COMPANY:
                FORMER CONFORMED NAME:  CASSA DEPOSITI & PRESTITI SOCIETA PER ABIONI
                DATE OF NAME CHANGE:    20031222

SUBJECT COMPANY:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 ENI SPA
                CENTRAL INDEX KEY:                      0001002242
                STANDARD INDUSTRIAL CLASSIFICATION:     CRUDE PETROLEUM & NATURAL GAS [1311]
                IRS NUMBER:                             000000000
                FISCAL YEAR END:                        1231

        FILING VALUES:
                FORM TYPE:              SC 13G
                SEC ACT:                1934 Act
                SEC FILE NUMBER:        005-49330
                FILM NUMBER:            031068133

        BUSINESS ADDRESS:
                STREET 1:               PIAZZALE ENRICO MATTEI 1
                CITY:                   ROME ITALY
                STATE:                  L6
                ZIP:                    00144
                BUSINESS PHONE:         011390659822449

        MAIL ADDRESS:
                STREET 1:               PIAZZALE ENRICO MATTEI 1
                CITY:                   ROME ITALY
```

- 8 - K

```
<SEC-DOCUMENT>0001193125-20-050884.txt : 20200227
<SEC-HEADER>0001193125-20-050884.hdr.sgml : 20200227
<ACCEPTANCE-DATETIME>20200227061421
ACCESSION NUMBER:               0001193125-20-050884
CONFORMED SUBMISSION TYPE:      8-K
PUBLIC DOCUMENT COUNT:          14
CONFORMED PERIOD OF REPORT:     20200226
ITEM INFORMATION:               Submission of Matters to a Vote of Security Holders
FILED AS OF DATE:               20200227
DATE AS OF CHANGE:              20200227

FILER:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 Apple Inc.
                CENTRAL INDEX KEY:                      0000320193
                STANDARD INDUSTRIAL CLASSIFICATION:     ELECTRONIC COMPUTERS [3571]
                IRS NUMBER:                             942404110
                STATE OF INCORPORATION:                 CA
                FISCAL YEAR END:                        0926

        FILING VALUES:
                FORM TYPE:              8-K
                SEC ACT:                1934 Act
                SEC FILE NUMBER:        001-36743
                FILM NUMBER:            20658351

        BUSINESS ADDRESS:
                STREET 1:               ONE APPLE PARK WAY
                CITY:                   CUPERTINO
                STATE:                  CA
                ZIP:                    95014
                BUSINESS PHONE:         (408) 996-1010

        MAIL ADDRESS:
                STREET 1:               ONE APPLE PARK WAY
                CITY:                   CUPERTINO
                STATE:                  CA
                ZIP:                    95014

        FORMER COMPANY:
                FORMER CONFORMED NAME:  APPLE INC
                DATE OF NAME CHANGE:    20070109

        FORMER COMPANY:
                FORMER CONFORMED NAME:  APPLE COMPUTER INC
                DATE OF NAME CHANGE:    19970808
</SEC-HEADER>
<DOCUMENT>
<TYPE>8-K
<SEQUENCE>1
<FILENAME>d865740d8k.htm
<DESCRIPTION>8-K
<TEXT>
```

Process Description

First, we use the EDGAR package in Python to download the data through API. Since the data in the EDGAR database is huge, we used a multithreading process in Python to speed up the downloading.

Then, given the .txt data, after careful observation, we found that: 1. all files are almost structured with particular signals, like ‘FILER:’, '</SEC-HEADER>', to separate the parts and 2. the SC 13G has different formats and the others are similar. So we read the data together except for SC 13G file type and extracted the file data and company data separately.

Finally, we clean the data.

For company data, we use regular expressions to remove symbol ‘\t’, space, and next line symbols. We leave the address not combined so as to create a cleaner view of data frame columns. Then we fill in the missing columns for a few files. For example, some of them don’t have “Address 2”. We let the code automatically detect these missing values leave them as null in the DataFrame.

For filing data, we remove the missing data and symbols like ‘\t’, merge the address data, distinguish different rows with the same characters and keep them with extra numbers, rename the index to join and reorder the columns to get a final full clean table.

Main Code Screenshots

- Get data

```
# Import modules
import concurrent.futures
from sec_edgar_downloader import Downloader
import pandas as pd
from tqdm import tqdm
```

```
dl = Downloader('data/')
filetypes = ['8-K','10-K','10KSB','10-Q','13F-NT','13F-HR','SC 13G','SD','S-1']

cik = list(set(df.CIK[1:]))
c = sorted(cik)

for i in tqdm(c):
    def download_edgar(filetype):
        try:
            dl.get(filetype,i,1)
        except:
            pass
    # Initialize multithreading. Loop will be iterated based on diff companies.
    # download_edgar function needs to be modified if one wants to iterate on file types in the multi threads.
    with concurrent.futures.ThreadPoolExecutor() as executor:
        executor.map(download_edgar, filetypes)
```

```
100%|██████████| 51883/51883 [13:34:45<00:00,  1.50s/it]
```

- Company data

```
# integrate all to be one big df

not_in_format = []
for p in tqdm(path):
    try:
        file = open(p)
        lines = [line for line in file.readlines() if line.strip()]
        lines = [x.strip() for x in lines]
        for i in range(len(lines)):
                try:
                    lines[i] = re.sub('\t+','',lines[i])
                except:
                    pass
        file.close()

        try:
            FB_row = lines.index('FILED BY:')
        except:
            FB_row = 9999
        header_row = lines.index('</SEC-HEADER>')
        if FB_row > header_row:
            end = header_row
        else:
            end = FB_row
        companystart = lines.index('COMPANY DATA:') + 1
        lines = lines[companystart:end]
        for i in range(len(col[:22])):
            if lines[i].split(':')[0] != col[i]:
                lines.insert(i,col[i] + ':')


        values = []
        for i in lines:
            values.append(i.split(':')[1])
        if len(values) != df.shape[0]:
            for i in range(df.shape[0]-len(values)):
                values.append('None')
        if lines.index('MAIL ADDRESS:') != 18:
            not_in_format.append(values[1])
            continue
        df[f'{values[1]}'] = values
    except:
        continue
```

- Result

	A	B	C	D	E	F	G	H	I
1	COMPANY	CENTRAL I	STANDARI	IRS NUMB	STATE OF	FISCAL YE/	FILING VAI	FORM TYP	SEC ACT
2	NICHOLAS	1000045	SHORT-TE	5.93E+08	FL	331		SC 13G	1934 Act
3	MEDALLIO	1000209	FINANCE S	43291176	DE	1231		10-K	1934 Act
4	HENRY SCI	1000228	WHOLESA	1.13E+08	DE	1228		SC 13G	1934 Act
5	CORE LAB(	1000229	OIL, GAS F	0	P7	1231		10-K	1934 Act
6	OPTICAL C	1000230	DRAWING	5.41E+08	VA	1031		10-K	1934 Act
7	KENTUCKY	1000232	STATE COI	6.11E+08	KY	1231		10-K	1934 Act
8	IMPAC MO	1000298	REAL ESTA	3.31E+08	MD	1231		10-K	1934 Act
9	SCHWEITZ	1000623	PAPER MIL	6.22E+08	DE	1231		SC 13G	
10	BLONDER	1000683	RADIO & T	5.22E+08	DE	1231		SC 13G	1934 Act
11	NOVAVAX	1000694	BIOLOGICA	2.23E+08	DE	1231		10-K	1934 Act
12	WATERS C	1000697	LABORATC	1.34E+08	DE	1231		SC 13G	1934 Act

- Filing data

```
def cleaning_trans_comp(info):

# remove NAs and tabs
    info = [x for x in info if x == x]

    for i in range(len(info)):
        try:
            info[i] = re.sub('\t+','',info[i])
        except:
            pass

# distinguish duplicated keys in different files
    for i in range(len(info)):
        for ii in range(i+1,len(info)):
            if (info[i].split(':')[0] + ':') in info[ii] and info[i] != '':
                info[ii] = info[ii].split(':')[0] + '_' + str( "%04d" % ii) + ':' + info[ii].split(':')[1]

# transform to dictionary
    dic = {}
    for i in info:
        try:
            dic.update({i.split(':')[0]:i.split(':')[1].strip('\t')})
        except:
            pass

# transform to the dataframe
    df = pd.DataFrame(dic, index = [0])

# put the address together(street, city, state, zip)
    for i, col in enumerate(df.columns):
        if 'ADDRESS' in col:
            df[col] = df[col].str.cat(df.iloc[:,i+1:i+6],sep = ',')
            df.iloc[:, i+1:i+6] = ''

    return df
```

```
def cleaning_file(info):
# remove NAs and tabs
    info = [x for x in info if x == x]

    for i in range(len(info)):
        try:
            info[i] = info[i].replace('\t','')
        except:
            pass

    for i in range(len(info)):
        info[i] = info[i].lstrip()
        if '>' in info[i] and ':' not in info[i]:
            info[i] = info[i].replace('>', '>:')
        elif '>' in info[i] and ':' in info[i]:
            info[i] = info[i].replace(':',';')
            info[i] = info[i].replace('>', '>:')


# distinguish duplicated keys in different files
    for i in range(len(info)):
        for ii in range(i+1,len(info)):
            if (info[i].split(':')[0] + ':') in info[ii] and info[i] != '':
                info[ii] = info[ii].split(':')[0] + '_' + str( "%04d" % ii) + ':' + info[ii].split(':')[1]


    info = [x for x in info if x != '']

    return info
```

```
def trans(info):
# transform to dataframe
    df_file = pd.DataFrame()

    dic_file = {}
    idx = 0
    for k in info:
        for j in k:
            try:
                if j.split(':')[1] != '':
                    dic_file.update({j.split(':')[0]:j.split(':')[1]})
            except:
                pass
        tmp = pd.DataFrame(dic_file, index = [idx])

        df_file = df_file.join(tmp.T,how = 'outer')

        dic_file = {}
        idx += 1

    return df_file
```

```
datafile = pd.DataFrame()
companies = glob.glob('sec_edgar_filings/*')
jump = 0

# def final_function(companies):
for company in tqdm(companies):
    try:
        path = glob.glob(company + '/*/*.txt')

        infofile, infoFB = get_file(path)

        infofile = list(map(cleaning_file, infofile))
        infofile = trans(infofile)


        if infoFB!= []:
            infoFB = cleaning_trans_comp(infoFB)

            new = []
            for i in infoFB.columns:
                new.append('FILED BY:' + i)
            infoFB.columns = new

            infoFB.index = infofile.T[infofile.T['CONFORMED SUBMISSION TYPE'] == 'SC 13G'].index
            infofile = infofile.T.join(infoFB, how = 'left', lsuffix = 'CONFORMED SUBMISSION TYPE',
                                                             rsuffix = 'FORM TYPE')

            result = infofile.T.drop_duplicates()
            df = result.T
        else:
            df = infofile.T.drop_duplicates()

        df.index = df.index.map(lambda x: x + jump)
        jump += len(df.T.columns)
        datafile = datafile.join(df.T, how = 'outer')
    except:
        pass
```

- Result

```
result.head()
```

	Filing_Company_CIK	CONFORMED SUBMISSION TYPE	<ABS-ASSET-CLASS>	<ABS-ASSET-CLASS>_0015	<ABS-SUB-ASSET-CLASS>	<ACCEPTANCE-DATETIME>	<DEPOSITOR-CIK>	<IMS-DOCUMENT>	<IMS-HEADER>	<ITEM-NUMBER>
0	1000045	10-K	NaN	NaN	NaN	20190628090223	NaN	NaN	NaN	NaN
1	1000045	10-Q	NaN	NaN	NaN	20200214083203	NaN	NaN	NaN	NaN
2	1000045	10KSB	NaN	NaN	NaN	20050629170036	NaN	NaN	NaN	NaN
3	1000045	8-K	NaN	NaN	NaN	20200225163027	NaN	NaN	NaN	NaN
4	1000045	SC 13G	NaN	NaN	NaN	20200212154805	NaN	NaN	NaN	NaN