EDGAR Project – Sentiment Analysis on Financial Statements

In the US, the Securities and Exchange Commission (SEC) requires public companies to file 10-K report annually. The 10-K form gives detailed report on the financial position of the company and contains a summary of relevant topics such as business activities, risk factors and legal proceedings.

You work as a data engineer / data scientist at a fund. Your employer wants to determine whether negative sentiment 10-K filings can be used to predict short term movements in share prices. You will create a package called *edgar* which will contain all code needed for this task.

Specifically, for each 10-k filing, you want to count the number of words classified as 'Negative', 'Positive', 'Uncertain', etc, and determine whether this has an impact on short-term price movements.

For word classifications, we will use the Loughran-McDonald dictionary for word classifications designed specifically for financial documents. In this dictionary, words such as 'adversity', 'controversial' and 'failing' are negative, whereas words such as 'achieving', 'beneficial', and 'succeed' are positive.

The sentiments classifications in the dictionary include negative, positive, uncertainty, superfluous among others. The dictionary can be found here:
https://sraf.nd.edu/textual-analysis/resources/#Master%20Dictionary

Navigating the Website

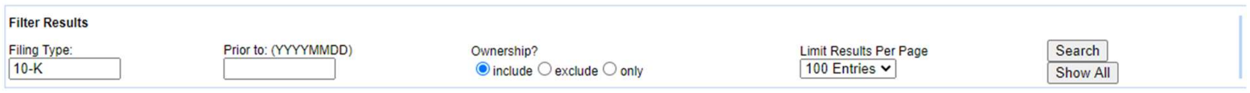
Step 1. Search for a Company

The SEC EDGAR company filings webpage can be found here:
https://www.sec.gov/edgar/searchedgar/companysearch.html

To search for company filings, type in the company's ticker symbol and click search. For example, to search for Apple's filings, search for AAPL. This will take you to https://www.sec.gov/cgi-bin/browse-edgar?CIK=320193.

Step 2. Filtering Search Results

Step 1 will take you to the search results. To search for 10-K filings, type in "10-K" in the Filing Type filter box and click "Search".

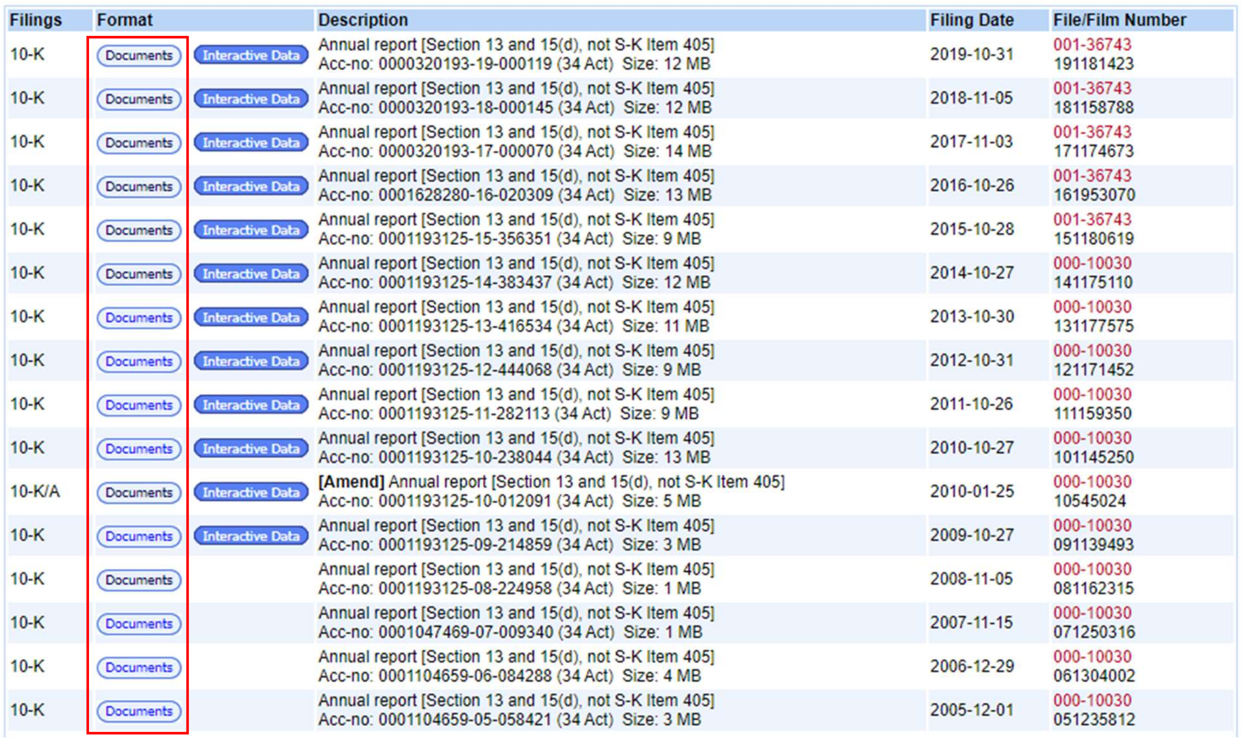


Filter Results

Filing Type: 10-K
Prior to: (YYYYMMDD)
Ownership? [✓ include] exclude only
Limit Results Per Page: 100 Entries

Figure 1. Filtering Search Results

This will then give you a table of 10-K filings for the company. To see the company filings for a given year, click the "Documents" button for the desired year. This will take you to the report's artifact page.

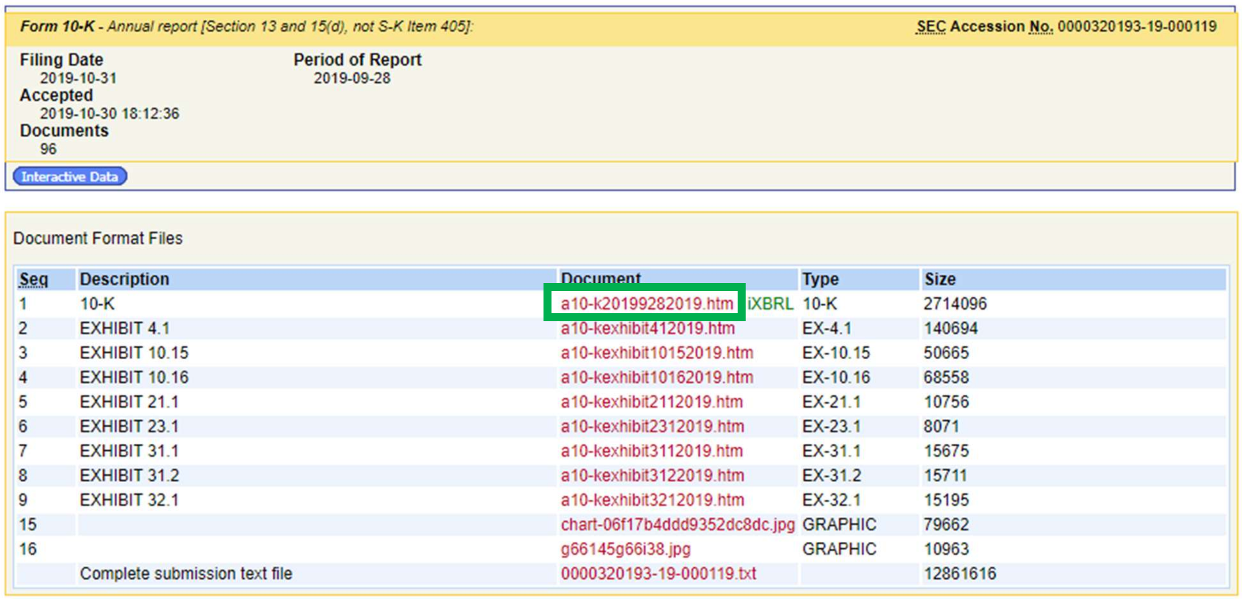


Filings	Format	Description	Filing Date	File/Film Number
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320193-19-000119 (34 Act) Size: 12 MB	2019-10-31	001-36743 191181423
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320193-18-000145 (34 Act) Size: 12 MB	2018-11-05	001-36743 181158788
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320193-17-000070 (34 Act) Size: 14 MB	2017-11-03	001-36743 171174673
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001628280-16-020309 (34 Act) Size: 13 MB	2016-10-26	001-36743 161953070
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-15-356351 (34 Act) Size: 9 MB	2015-10-28	001-36743 151180619
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-14-383437 (34 Act) Size: 12 MB	2014-10-27	000-10030 141175110
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-13-416534 (34 Act) Size: 11 MB	2013-10-30	000-10030 131177575
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-12-444068 (34 Act) Size: 9 MB	2012-10-31	000-10030 121171452
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-11-282113 (34 Act) Size: 9 MB	2011-10-26	000-10030 111159350
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-10-238044 (34 Act) Size: 13 MB	2010-10-27	000-10030 101145250
10-K/A	Documents Interactive Data	**[Amend]** Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-10-012091 (34 Act) Size: 5 MB	2010-01-25	000-10030 10545024
10-K	Documents Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-09-214859 (34 Act) Size: 3 MB	2009-10-27	000-10030 091139493
10-K	Documents	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-08-224958 (34 Act) Size: 1 MB	2008-11-05	000-10030 081162315
10-K	Documents	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001047469-07-009340 (34 Act) Size: 1 MB	2007-11-15	000-10030 071250316
10-K	Documents	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001104659-06-084288 (34 Act) Size: 4 MB	2006-12-29	000-10030 061304002
10-K	Documents	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001104659-05-058421 (34 Act) Size: 3 MB	2005-12-01	000-10030 051235812

Figure 2. Example of Search Results

Step 3. Navigating the Artifacts Page

When on the Artifacts page, we can click the 10-K artifact to get the 10-K report in html format.



Form 10-K - *Annual report [Section 13 and 15(d), not S-K Item 405]:* **SEC Accession No.** 0000320193-19-000119

Filing Date 2019-10-31
Accepted 2019-10-30 18:12:36
Documents 96
Period of Report 2019-09-28

Interactive Data

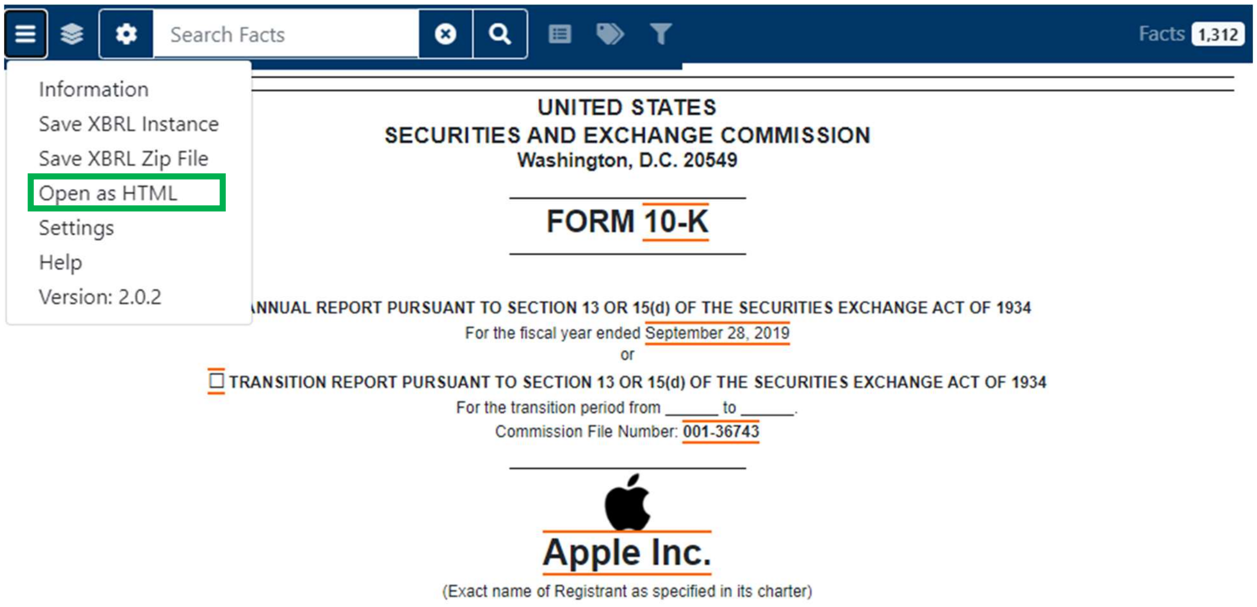
Document Format Files

Seq	Description	Document	Type	Size
1	10-K	a10-k20199282019.htm iXBRL	10-K	2714096
2	EXHIBIT 4.1	a10-kexhibit412019.htm	EX-4.1	140694
3	EXHIBIT 10.15	a10-kexhibit10152019.htm	EX-10.15	50665
4	EXHIBIT 10.16	a10-kexhibit10162019.htm	EX-10.16	68558
5	EXHIBIT 21.1	a10-kexhibit2112019.htm	EX-21.1	10756
6	EXHIBIT 23.1	a10-kexhibit2312019.htm	EX-23.1	8071
7	EXHIBIT 31.1	a10-kexhibit3112019.htm	EX-31.1	15675
8	EXHIBIT 31.2	a10-kexhibit3122019.htm	EX-31.2	15711
9	EXHIBIT 32.1	a10-kexhibit3212019.htm	EX-32.1	15195
15		chart-06f17b4ddd9352dc8dc.jpg	GRAPHIC	79662
16		g66145g66i38.jpg	GRAPHIC	10963
	Complete submission text file	0000320193-19-000119.txt		12861616

Figure 3. Example Artifact Page

Step 4. Getting Clean HTML

After clicking on the link in Step 3, the html rendered uses inline javascript to populate contents. To load the 10-K report with all data populated inline, click “Open as HTML” in the menu on the top left.



UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

FORM 10-K

NNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934
For the fiscal year ended September 28, 2019
or
☐ TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934
For the transition period from ______ to ______.
Commission File Number: 001-36743

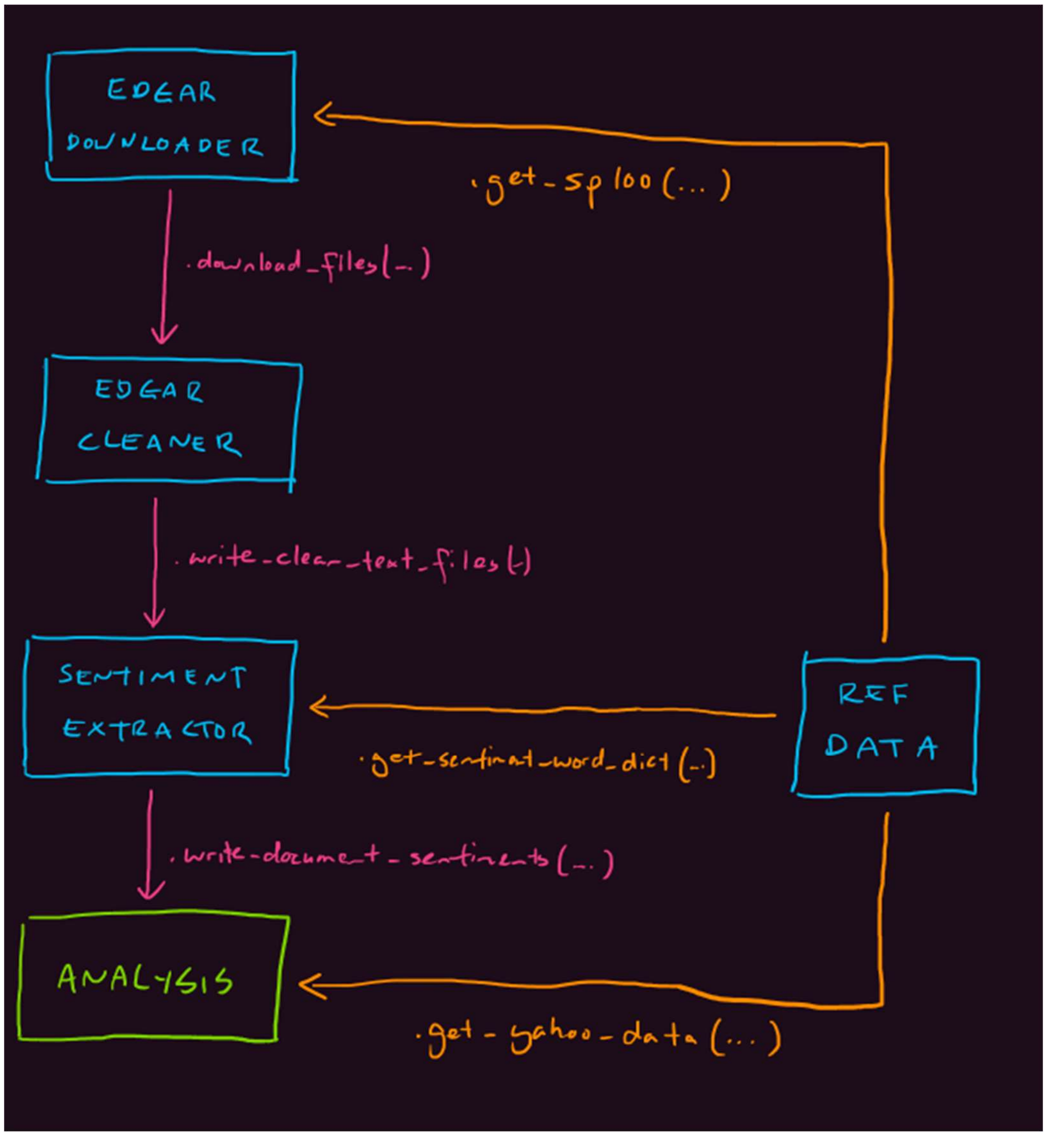
Apple Inc.
(Exact name of Registrant as specified in its charter)

Congratulations - you now know how to navigate to a 10-K report for a given company and year!

Pipeline Overview

The high level components are as follows:

Component	Description
EDGAR Downloader	Downloads raw 10-k filings from the SEC EDGAR website.
EDGAR Cleaner	Given raw 10-k html files, perform text cleaning.
Sentiment Extractor	Given a corpus of clean 10-k documents, count the number of negative sentiment words for the document. Repeat for words classified as positive, uncertain, litigious, etc.
Ref Data	Module used for retrieving data such as tickers in the S&P100, sentiment word lists and stock market data.



Part 1. Data Ingestion

Task Description

Create a module called `edgar_downloader`. The module should contain the functions:

- `write_page(url, file_path)`
  Takes in the URL and writes the html file to the path specified.
- `download_files_10k(ticker, dest_folder)`
  Downloads all the html 10-k files for the given ticker into the destination folder. The files downloaded should the named using the following convention:
  `<ticker>_10-k_<filing_date>.html`

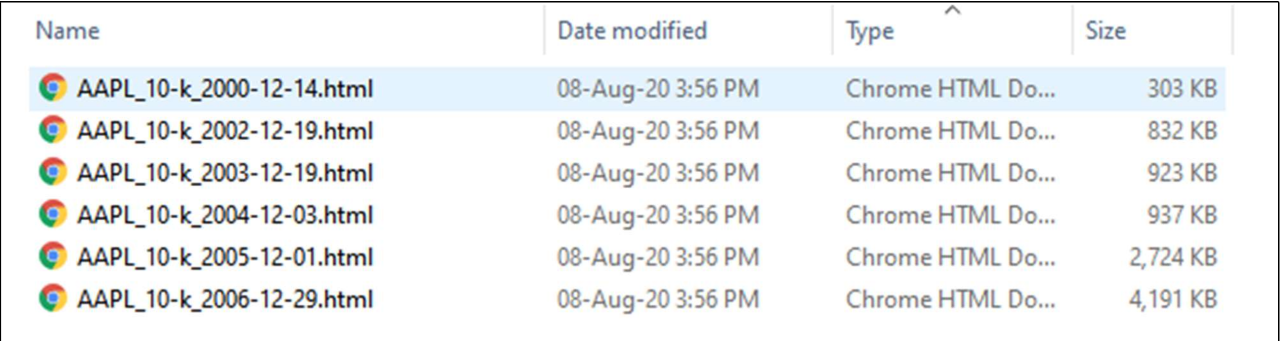
After you have unit tested your code, use your module to download filings for stocks in the S&P100.

Running your code

A user should be able to import your module and call the download_files_10k function to download files. For example, running the following will download all of Apple's filings into my C:/10k_filings_raw directory:

```
import edgar_downloader
edgar_downloader.download_files_10k('AAPL', 'C:/10k_filings_raw')
```

Running the code above should download reports to the destination folder. Example below.



Name	Date modified	Type	Size
AAPL_10-k_2000-12-14.html	08-Aug-20 3:56 PM	Chrome HTML Do...	303 KB
AAPL_10-k_2002-12-19.html	08-Aug-20 3:56 PM	Chrome HTML Do...	832 KB
AAPL_10-k_2003-12-19.html	08-Aug-20 3:56 PM	Chrome HTML Do...	923 KB
AAPL_10-k_2004-12-03.html	08-Aug-20 3:56 PM	Chrome HTML Do...	937 KB
AAPL_10-k_2005-12-01.html	08-Aug-20 3:56 PM	Chrome HTML Do...	2,724 KB
AAPL_10-k_2006-12-29.html	08-Aug-20 3:56 PM	Chrome HTML Do...	4,191 KB

Figure 4. Example of Downloaded Files in Folder

Part 2. Data Preparation / Cleaning

Task Description

Create a module called `edgar_cleaner`. The module should contain the following functions:

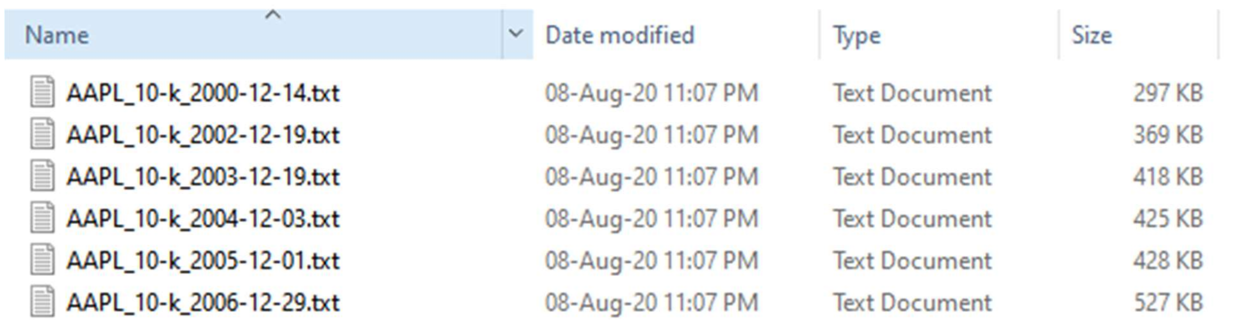
- `clean_html_text(html_text)`
  Takes in a html text string and removes tags and special characters. Returns result as a string.
- `write_clean_html_text_files(input_folder, dest_folder)`
  Takes all the html 10-k files in the input folder, cleans them using the clean_html_text function and writes them into the destination folder as text files. Files downloaded should follow the naming convention: `<ticker>_10-k_<filing_date>.txt`.

After unit testing your code, use your module to convert the html filings into clean text.

Running your code

Running the code below will read the previously written html 10-k filings and save them as text files with tags and special characters removed.

```
import edgar_cleaner
edgar_downloader.download_files_10k('C:/10k_filings_raw',
'C:/10k_filings_clean')
```



Name	Date modified	Type	Size
AAPL_10-k_2000-12-14.txt	08-Aug-20 11:07 PM	Text Document	297 KB
AAPL_10-k_2002-12-19.txt	08-Aug-20 11:07 PM	Text Document	369 KB
AAPL_10-k_2003-12-19.txt	08-Aug-20 11:07 PM	Text Document	418 KB
AAPL_10-k_2004-12-03.txt	08-Aug-20 11:07 PM	Text Document	425 KB
AAPL_10-k_2005-12-01.txt	08-Aug-20 11:07 PM	Text Document	428 KB
AAPL_10-k_2006-12-29.txt	08-Aug-20 11:07 PM	Text Document	527 KB

Figure 5. Example of Cleaned Files in Folder

Part 3A. Reference Data: S&P100 Data

Task Description

Create a module called `ref_data`. The module should contain the following functions:

- `get_sp100()`
  Returns a list of all tickers in the S&P100. Note that this can be a snapshot of the current constituents of the S&P100 and does not need to be updated live.

Running your code

Running the code below will give a list of all tickers in the S&P100.

```
import ref_data as edgar_data
tickers_sp100 = edgar_data. get_sp100()
print(tickers_sp100)
```

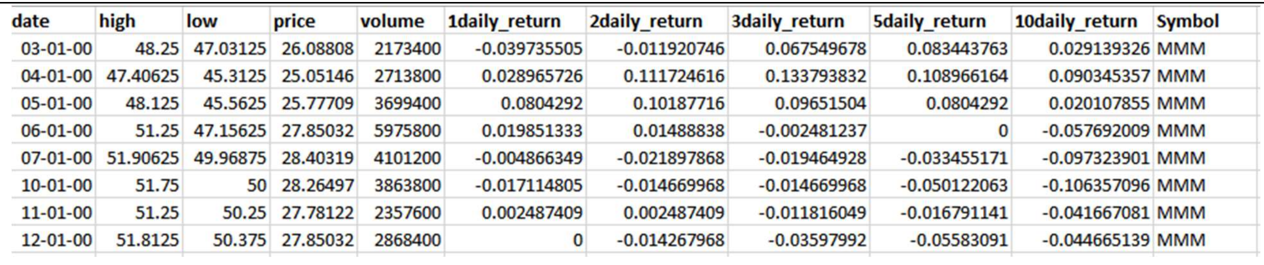
Part 3B. Reference Data: Yahoo Finance

Task Description

Update the `ref_data` module. The module should contain the following added function:

- `get_yahoo_data(start_date, end_date, tickers)`
  Downloads yahoo finance data and consolidates all data into one table. In addition, returns should be calculated for 1, 2, 3, 5 and 10 business day time horizons. For example, the 1-day return is the return made if the stock was bought today and sold the next day. Returns a dataframe.

An example of the output and the required columns is shown below:



date	high	low	price	volume	1daily_return	2daily_return	3daily_return	5daily_return	10daily_return	Symbol
03-01-00	48.25	47.03125	26.08808	2173400	-0.039735505	-0.011920746	0.067549678	0.083443763	0.029139326	MMM
04-01-00	47.40625	45.3125	25.05146	2713800	0.028965726	0.111724616	0.133793832	0.108966164	0.090345357	MMM
05-01-00	48.125	45.5625	25.77709	3699400	0.0804292	0.10187716	0.09651504	0.0804292	0.020107855	MMM
06-01-00	51.25	47.15625	27.85032	5975800	0.019851333	0.01488838	-0.002481237	0	-0.057692009	MMM
07-01-00	51.90625	49.96875	28.40319	4101200	-0.004866349	-0.021897868	-0.019464928	-0.033455171	-0.097323901	MMM
10-01-00	51.75	50	28.26497	3863800	-0.017114805	-0.014669968	-0.014669968	-0.050122063	-0.106357096	MMM
11-01-00	51.25	50.25	27.78122	2357600	0.002487409	0.002487409	-0.011816049	-0.016791141	-0.041667081	MMM
12-01-00	51.8125	50.375	27.85032	2868400	0	-0.014267968	-0.03597992	-0.05583091	-0.044665139	MMM

Figure 6. Snippet of the Yahoo Finance Output File

Use your module to create the yahoo finance reference data for stocks in the S&P100.

Running your code

Running the code below will write the yahoo finance reference data to file.

```
import ref_data as edgar_data
df_returns = edgar_data.get_yahoo_data('2000-01-01', '2020-08-01', tickers, 'daily')
df_returns.to_csv(‘C:/stock_returns_daily.csv, index=False)
```

Part 3C. Reference Data: Loughran-McDonald Sentiment Words

Task Description

Update the `ref_data` module. The module should contain the following added function:

- `get_sentiment_word_dict()`
  Returns a dictionary containing the LM sentiment words. The keys for the dictionary are the sentiments, and the values will be a list of words associated with that particular sentiment. For example, get_sentiment_word_dict()['Negative'] will return a list of words associated with negative sentiment.

Running your code

Running the code below will return the sentiment word dictionary.

```
import ref_data as edgar_data
sentiment_dict = edgar_data.get_sentiment_word_dict()
print(sentiment_dict)
```

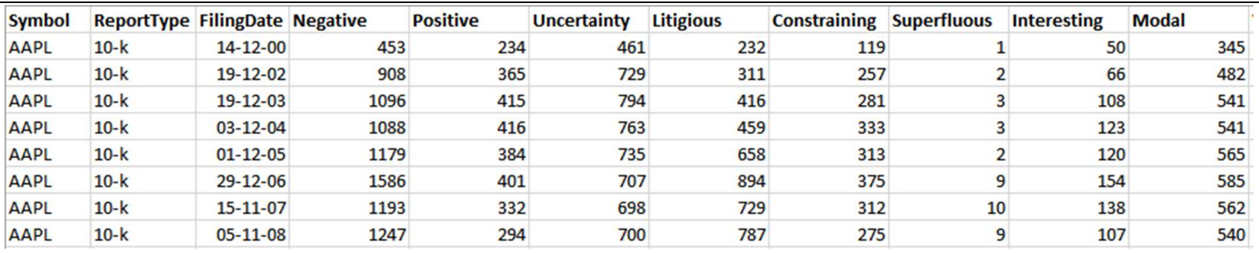
Part 4. Sentiment Word Counts

Task Description

Create a module called `edgar_sentiment_wordcount`. The module should contain the following function:

- `write_document_sentiments(input_folder, output_file)`
  Takes all the clean text 10-k files in the input folder, counts the number of words in the document belonging to a particular sentiment and outputs the results to the output file.

An example of the output and the required columns is shown below:



Symbol	ReportType	FilingDate	Negative	Positive	Uncertainty	Litigious	Constraining	Superfluous	Interesting	Modal
AAPL	10-k	14-12-00	453	234	461	232	119	1	50	345
AAPL	10-k	19-12-02	908	365	729	311	257	2	66	482
AAPL	10-k	19-12-03	1096	415	794	416	281	3	108	541
AAPL	10-k	03-12-04	1088	416	763	459	333	3	123	541
AAPL	10-k	01-12-05	1179	384	735	658	313	2	120	565
AAPL	10-k	29-12-06	1586	401	707	894	375	9	154	585
AAPL	10-k	15-11-07	1193	332	698	729	312	10	138	562
AAPL	10-k	05-11-08	1247	294	700	787	275	9	107	540

Figure 7. Snippet of the Sentiment Wordcount Output File

After unit testing your code, use your module to produce a sentiment word count summary for your clean 10-k filings.

Running your code

Running the code below will read the previously written clean text 10-k filings and write a table summarizing the sentiment word count for each document.

```
import edgar_sentiment_wordcount as edgar_sentiment
edgar_sentiment.write_document_sentiments(‘C:/10k_filings_clean’,
‘C:/sentiment_factors.csv)
```

Part 4B. Sentiment Word Counts

Task Description

Insert your results from part 4 into a SQL database so that it can easily be queried by others.

Part 5. Sentiment Analysis

Task Summary

Join the yahoo financials data (part 3B) with the sentiment word counts data (part 4) to determine whether there is a link between negative word counts and stock price returns.

This part has left intentionally open. Try plotting out the relationship between negative word counts and volume or returns. You can try to think of ways to improve your analysis and adjust previous steps where necessary. Be creative!