Scraping EDGAR with Python

1 author:

Rasha Ashraf
Georgia State University
10 PUBLICATIONS **42** CITATIONS

Scraping EDGAR with Python

Rasha Ashraf
J. Mack Robinson College of Business
Georgia State University
35 Broad St NW
Atlanta, GA 30303
Phone: (1) 404 413 7348
Email: rashraf@gsu.edu

ABSTRACT

This paper presents Python codes that can be used to extract data from SEC filings. The Python program web crawls to obtain URL paths for company filings of required reports, such as 10-K. The program then performs a textual analysis and counts the number of occurrences of words in the filing that reflect, for example, uncertainty (or any other quality specified by the researcher). The program can be easily modified to conduct other searches by changing the word list, company names, or SEC filings. The Python program could be used in an introductory graduate data analytics course in finance that has a web crawling or textual analysis component.

Scraping EDGAR with Python

1. Introduction

Scraping the SEC's EDGAR (Electronic Data Gathering, Analysis, and Retrieval system) filings using programs like Python, R, or SAS has become a widely used tool for researchers and practitioners to extract data and other information that are not readily available in a databases such as COMPUSTAT, Execucomp, or SDC. Garcia and Norli (2012) describe using the Pearl programming language to crawl EDGAR. Engelberg and Sankaraguruswamy (2007) document how to crawl EDGAR using SAS. In this paper I document and explain basic Python codes that are designed to extract data and information from SEC filings. I provide the steps to obtain the URL address for a filing (e.g., 10-K) by a company from the SEC EDGAR Master Index. To illustrate how to obtain information from any filing I show the steps to find key words that reflect uncertainty, following Loughran and McDonald's (2011) financial dictionary.

Textual analysis is becoming a primary research field in many areas, including finance and accounting. Studies have documented, for example, whether negative words used by popular news outlets affect individual stocks and aggregate market returns (Tetlock, 2007; Tetlock, Saar-Tsechansky, and Macskassy, 2008). Loughran and McDonald (2011) argue that some positive and negative words from the Harvard Psycho-sociological Dictionary, which is a commonly used source for word classifications, lead to misleading interpretations in the financial context.[1] The authors create a list of words related to finance in six categories: negative, positive, uncertainty, litigious, strong modal, and weak modal. They show a significant relation between the words and 10-K filing date return, trading volume, subsequent return volatility, unexpected earnings, and

---

[1] See Loughran and McDonald (2016) for a survey on textual analysis.

fraud. As a simple illustration of how to extract data from SEC filings using the Python program, I select ten words from the Loughran and McDonald (2011) financial sentiment dictionary representing uncertainty and use those to scrape through the 10-K filings for sixteen firms.

I present algorithms to extract textual data from SEC EDGAR filings using Python programs. First, I present how to use Python program to count number of occurrences of the selected words signaling uncertainty in a 10-K filing for a given company in year when the URL address of the filing is known. Next, I present algorithm and Python program in case the URL address of the company filing is not known in advance and has to be extracted from SEC Master Index. The SEC's computer system uses a Central Index Key (CIK) to identify companies that have filed a disclosure with the SEC. Using the CIK, the Python program searches in the EDGAR Master Index for a specific quarter in which a company filed the 10-K, obtains the URL path for the file, and then opens and loads the file to compute the frequency of the chosen words indicating uncertainty. Finally, I present the algorithm of Python program that performs the above task for a list of firms for corresponding years for which 10-K filings are extracted and a word count is performed for each company. As an example, I search in the 10-K filings of Microsoft and its fifteen peers in 2015 for the words indicating uncertainty following Loughran and McDonald's (2011) financial dictionary and report the results. I do not perform any analysis with the words extracted. The main objective of this paper is to explain to researchers and practitioners how to use Python codes, which they can later change to address their own questions.

2. Extracting Uncertainty Words from 10-K filings

There is an abundance of information in SEC filings such as the 10-K and 8-K. One can investigate whether a positive outlook or sentiment, or a company's risk-taking attitude is expressed predominantly in annual reports, affects the future performance of the firm, returns to its stock, the growth of the firm, or its future risk-taking behavior. In this paper I do not present any hypotheses related to textual analysis with EDGAR data. I only explain how Python code can be used to count the frequency of certain words that appear in SEC filings. The code can be expanded to extract other key words relevant to any research project.

I present a simple Python code to extract key words from an SEC filing in .txt format. Using the URL for the filing, for an SEC filing in .txt format, this program will search each line of the filing looking for key words provided by the researcher. The code is titled Program1.py, and the details of the code are provided in the Appendix. The algorithm of the code Program1.py is as follows:

a. Open the page and load using the URL for the 10-K filing of a company in a year.

b. The program will search each line of the page and split each line into elements (words) with whitespace as a separator.

c. For a given word list, the program will search each element in each line of the page and count the frequency of the specified words.

d. The program provides a final count of the list of specified words. It also outputs the CIK, year, and the file path URL address. CIK is a unique identifier for each company (or any other entity that files with SEC) in the EDGAR system.

The URL path name of the SEC filing may not be known in advance but needs to be searched from the SEC EDGAR Master Index. In the next section I describe how to search for the URL path in the Master Index and then use it in the logic in Program1.py.

3. EDGAR Indexes

The U.S. Securities and Exchange Commission's HTTPS file system allows comprehensive access to the SEC's EDGAR filings by corporations, funds, and individuals. EDGAR Full Indexes list all public SEC filings for each quarter starting in the third quarter of 1994 to the present.[2]

There are four types of EDGAR indexes that list company filings for all four quarters of each year. Three of the main indexes sort on the basis of (1) company name, (2) form type, and (3) CIK number. For each filing the EDGAR indexes provide CIK, Company Name, Form Type, Date Filed, and File Path (URL address). Using the link below one can access the EDGAR Full Index listed by year: https://www.sec.gov/Archives/edgar/full-index/. Figure 1 shows archives of the SEC EDGAR Full Index year by year from 1993 to the current year. After selecting a specific year, such as 2013, the following link will take one to the index of the four quarters for the year: https://www.sec.gov/Archives/edgar/full-index/2013/. Figure 2 shows the SEC filing indexes for the four quarters of 2014.

[Insert Figure 1 about here.]

[Insert Figure 2 about here.]

---

[2] Source: https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm

Choosing a particular quarter will direct you to four different types of indexes: Company, Form, Master, and XBRL, as shown in Figure 3. The first three indexes show the same information, sorted by company name, form type, and CIK number respectively. XBRL includes voluntary filer program submissions and is sorted by CIK number. Our program will search the Master Index to obtain the path for a company's filing. A typical Master Index is shown in Figure 4.

[Insert Figure 3 about here.]

[Insert Figure 4 about here.]

4. Opening the SEC Filing for a Company using the Master Index

Say we want to find the URL path for a particular SEC filing, such as a 10-K, for a given CIK in a quarter-year. For example: for AMAZON COM INC. (CIK: 1018724) in the first quarter of 2013, I want to search the Master Index for the URL address of the 10-K filing. I record the URL of the filing and open it to perform a textual analysis as described in section 2. I use Program2.py (provided in the Appendix) to perform this task following the algorithm below:

a. Access the Master Index for a given quarter-year using the URL from the SEC Edgar Archives. For example, for access to the Master Index for the third quarter of 2013, the URL is as follows:
   url1 = https://www.sec.gov/Archives/edgar/full-index/2013/QTR3/master.idx

b. Open and load the page with the url1

c. Go through each line of the Master Index and find the CIK (1018724) and filing (e.g., 10-K), extract the text file path highlighted in Figure 5, and store it as follows:
   element4='edgar/data/1018724/0001193125-13-028520.txt'

d. The path of the 10-K filing is stored as url2 = 'https://www.sec.gov/Archives/'+element4

url2=https://www.sec.gov/Archives/edgar/data/1018724/0001193125-13-028520.txt

e. Open the page at url2, then perform word count using Program1.py.

[Insert Figure 5 about here.]

If the CIK of the company is not known, the above program is modified to incorporate a search in the Master Index for the CIK for a specific company name (discussed in the next section).

5. Search the SEC Filings of Multiple Companies

It is possible to search for SEC filings by multiple companies with a single program and perform the word count for each company filing. This search uses the following logic:[3]

a) This program first opens a file that contains the names of the companies and the corresponding year for which the filing (e.g., 10-K) needs to be searched. The filing name (10-K) is hard coded in the program but can be changed according to the needs of the researcher.

b) After reading the company names and corresponding years, the program then searches the Master Index for all quarters for the CIK for each company and the quarter in which the filing is available.

c) Then the program opens the Master Index again for the selected quarter and looks for the CIK and filing (e.g., 10-K) and obtains the URL path for the filing (similar to the logic in Program2.py).

d) Then the page of the filing (10-K) is loaded using the URL obtained in step (c).

---

[3] Due to the length of Program 3, these codes are not provided in the paper and are available upon request.

e) The program then goes over each line of the filing page (.txt format) looking for the key words and counting the frequency of the words (similar to the logic in Program1.py).

f) For each company the program outputs the number of occurrences of each key word. If a filing for the company is not found, then the code reports an empty word count.

6. Results

I present a simple exercise to illustrate how to access and extract information or data from EDGAR filings. I construct a list of ten words that reflect uncertainty following Loughran and McDonald's (2011) financial dictionary. The sample list of ten key words is: anticipate, believe, depend, fluctuate, indefinite, likelihood, possible, predict, risk, uncertain. These words are hard coded in the program. The list can be expanded or modified to omit some words or include others.

In Program1.py (presented in the Appendix), using the URL path for the filing, I measure the frequency of the words listed above in the 10-K filing for Amazon Com. in 2013.   The program prints out the frequency of the "uncertainty" word list. If the URL path of the 10-K filing is not known, Program2.py (presented in the Appendix) can crawl through the EDGAR Master Index and obtain it before performing the task word count. In this program the inputs are CIK, year, and the SEC file to be retrieved. The output of the program is the URL path and the word count in Amazon's 10-K filing in 2013. These outputs are shown in the Appendix after running each program.

Next, I expand the above programs to search for SEC filings of multiple companies and perform the word count for each company filing.[4] I look for the words indicating uncertainty in

---

[4] These codes are not provided in the paper and are available upon request.

the 10-K filings of Microsoft and its peers in 2015 as reported in the proxy statement of Microsoft (SEC filing DEF 14A). Table 1 provides the name of the companies and the corresponding years for which the program searches in the corresponding 10-K filings for the key words. . One limitation is that the names must be provided in the exact format in which they appear in the SEC filing in order to be matched with the company names.

[Insert Table1 about here.]

The output of the program is constructed in the format of Table 2, where it reports the CIK, the quarter for which the 10-K is obtained in 2015, and the number of occurrences of each key word. If the company does not report filing a 10-K for the specified year for any of the four quarters, the program will output the symbol (*) in the corresponding cell(s). For example, HEWLETT PACKARD CO. did not report a 10-K filing in 2015 for any of the four quarters, so the report outputs (*) for each quarter. The program will also output (*) if it is unable to detect any company with the specified name.

[Insert Table 2 about here.]

7. Summary

With the advancement of computational power and machine learning tools, gathering, processing, and analyzing data efficiently has become one of the essential elements for advancement both in research and in practice. Traditionally, researchers and practitioners would rely on data from COMPUSTAT, Execucomp, or SDC to get valuable data on companies' performance metrics, executive compensation, and M&A deals. With programs like Python and R, now it is easy to web crawl and gather specific or unique data from SEC EDGAR filings, news articles, and social media like twitter and analyze or predict economic, social, or consumer

behavior. This paper describes Python programs for beginners that allow data to be scraped from SEC filings for further analysis. The codes can be modified to meet the needs of the researcher to obtain data from any SEC filings. The techniques described in this paper could be used in a graduate-level course on data analytics that has a web crawling component and to explore the abundance of data in SEC filings.

References

Garcia, D., and Norli., O. (2012). Crawling EDGAR. *The Spanish Review of Financial Economics*, 10(1), 1–10.

Engelberg, J., and Sankaraguruswamy, S. (2007). How to Gather Data Using a Web Crawler: An Application Using SAS to Search EDGAR. Working paper, Northwestern University and National University of Singapore.

Loughran, T., and McDonald, B. (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), 35–65.

Loughran, T., and McDonald, B. (2016). Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research*, 54(4), 1187–1230.

Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *Journal of Finance,* 62, 1139–1168.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More Than Words: Quantifying Language to Measure Firms' Fundamentals. *The Journal of Finance,* 63, 1437–1467.

Appendix

Program 1

```
# Given a URL path for EDGAR 10-K file in .txt format for a company
(CIK) in a year this code will perform word count

import urllib2
import time
import csv
import sys

CIK = '0001018724'
Year = '2013'

string_match1 = 'edgar/data/1018724/0001193125-13-028520.txt'

url3 = 'https://www.sec.gov/Archives/'+string_match1
#load the page in for the given url in response3
#urllib2 is a Python module that can be used for fetching URLs

response3 = urllib2.urlopen(url3)

#words: list of uncertainty words from Loughran and McDonald (2011)

words = ['anticipate', 'believe', 'depend', 'fluctuate', 'indefinite',
'likelihood', 'possible', 'predict', 'risk', 'uncertain']

count={} # is a dictionary data structure in Python
for elem in words:
    count[elem] = 0

#The method split() returns a list of all the words in the string
#The split function splits a single string into a string array using
#the separator defined.
#If no separator is defined, whitespace is used.

for line in response3:
    elements = line.split()
    for word in words:
        count[word] = count[word]+elements.count(word)

print (CIK)
print (Year)
print (url3)
print (count)
```

Program1.py output:

0001018724 #CIK of AMAZON COM
2013   #Given YEAR

URL address of 10-K filing
https://www.sec.gov/Archives/edgar/data/1018724/0001193125-13-028520.txt

Number of occurrences of uncertainty words
{'believe': 28, 'depend': 2, 'risk': 25, 'predict': 5, 'likelihood': 15, 'anticipate': 0, 'possible': 24, 'uncertain': 10, 'indefinite': 0, 'fluctuate': 3}

Program 2

```
# Given a Master Index URL of EDGAR find the path of raw text filing
#(10-K) for AMAZON COM (CIK 1018724) in year 2013

import urllib2
import time
import csv
import sys

CIK = '1018724' #AMAZON COM
Year = '2013'   #GIVEN
FILE='10-K'     #GIVEN


#####Get the Master Index File for the given Year


url='https://www.sec.gov/Archives/edgar/full-index/%s/QTR1/master.idx'
%(Year)
response = urllib2.urlopen(url)

string_match1 = 'edgar/data/'
element2 = None
element3 = None
element4 = None


###Go through each line of the master index file and find given CIK #and
FILE (10-K) and extract the text file path

for line in response:
    if CIK in line and FILE in line:
        for element in line.split(' '):
            if string_match1 in element:
                element2=element.split('|')
                for element3 in element2:
```

```
                     if string_match1 in element3:
                           element4=element3


# The path of the 10-K filing
url3 = 'https://www.sec.gov/Archives/'+element4
response3 = urllib2.urlopen(url3)


words = ['anticipate', 'believe', 'depend', 'fluctuate', 'indefinite',
'likelihood', 'possible', 'predict', 'risk', 'uncertain']

count={}
for elem in words:
    count[elem] = 0

for line in response3:
    elements = line.split()
    for word in words:
        count[word] = count[word]+elements.count(word)

print (CIK)
print (Year)
print (url3)
print (count)
```

Program2.py output:

0001018724 #CIK of AMAZON COM
2013   #Given YEAR

URL address of 10-K filing
https://www.sec.gov/Archives/edgar/data/1018724/0001193125-13-028520.txt

Number of occurrences of uncertainty words
{'believe': 28, 'depend': 2, 'risk': 25, 'predict': 5, 'likelihood': 15, 'anticipate': 0, 'possible': 24, 'uncertain': 10, 'indefinite': 0, 'fluctuate': 3}

Table 1: List of Company Names and Corresponding Years
The table reports Microsoft Corp. and its 2015 technology peer groups as reported in proxy statement (DEF 14A SEC Filing). This provides list of company names and corresponding years for which I look for company CIK and then the URL path for the 10-K filing.

Name	Year
MICROSOFT CORP	2015
Accenture plc	2015
ADOBE SYSTEMS	2015
AMAZON COM	2015
APPLE INC	2015
CISCO SYSTEMS	2015
EMC CORP	2015
Facebook Inc	2015
Google Inc	2015
HEWLETT PACKARD CO	2015
INTERNATIONAL BUSINESS MACHINES CORP	2015
INTEL CORP	2015
ORACLE CORP	2015
QUALCOMM INC	2015
SYMANTEC CORP	2015
YAHOO INC	2015

Table 2: Count of Uncertainty Words Extracted from the 10-K filings

The table displays the count of uncertainty words provided in Section 6 appeared in 10-K filings of Microsoft Corp. and its 2015 technology peer groups as reported in proxy statement (DEF 14A SEC Filing). The table also provides the CIK number of the companies and the quarter in which 10-K filing is available for respective companies in 2015.

Name	CIK	Quarter	Anticipate	Believe	Depend	Fluctuate	Indefinite	Likelihood	Possible	Predict	Risk	Uncertain
MICROSOFT CORP	789019	3	12	31	4	0	0	2	9	1	53	16
Accenture plc	1467373	4	2	23	6	1	6	0	21	4	55	24
ADOBE SYSTEMS	796343	1	14	80	3	5	1	5	27	6	92	9
AMAZON COM	1018724	1	0	25	2	4	0	15	19	2	31	11
APPLE INC	320193	4	3	7	11	1	12	16	14	0	113	11
CISCO SYSTEMS	858877	3	5	47	7	3	24	9	30	6	110	20
EMC CORP	790070	1	4	46	2	7	0	14	9	6	82	10
Facebook Inc	1326801	1	11	76	5	3	2	6	30	3	25	23
Google Inc	1288776	1	6	37	2	7	0	6	36	3	66	10
HEWLETT PACKARD CO	*	*	*	*	*	*	*	*	*	*	*	*
INTERNATIONAL BUSINESS MACHINES CORP	51143	1	3	3	6	0	0	11	29	11	148	11
INTEL CORP	50863	1	2	46	0	6	2	15	20	0	131	25
ORACLE CORP	1341439	2	4	64	13	3	20	13	44	2	74	44
QUALCOMM INC	804328	4	5	27	8	2	2	21	22	20	68	14
SYMANTEC CORP	849399	2	3	26	1	5	1	22	43	1	37	29
YAHOO INC	1011006	1	8	26	3	3	3	0	39	1	113	34

Figure 1: Archives of SEC Edgar Full Index for 1993 till current

**Figure 2:** SEC Edgar Index for 4 quarters in 2014

Figure 3: SEC Edgar archives for four different indexes: Company, Form, Master and XBRL in the third quarter of 2014

```
Description:           Master Index of EDGAR Dissemination Feed
Last Data Received:    March 31, 2013
Comments:              webmaster@sec.gov
Anonymous FTP:         ftp://ftp.sec.gov/edgar/




CIK|Company Name|Form Type|Date Filed|Filename
--------------------------------------------------------------------------
1000032|BINCH JAMES G|4|2013-03-04|edgar/data/1000032/0001181431-13-014069.txt
1000045|NICHOLAS FINANCIAL INC|10-Q|2013-02-08|edgar/data/1000045/0001193125-13-046001.txt
1000045|NICHOLAS FINANCIAL INC|4|2013-01-09|edgar/data/1000045/0001000045-13-000001.txt
1000045|NICHOLAS FINANCIAL INC|8-K|2013-01-30|edgar/data/1000045/0001193125-13-029629.txt
1000045|NICHOLAS FINANCIAL INC|8-K|2013-02-19|edgar/data/1000045/0001193125-13-064817.txt
1000045|NICHOLAS FINANCIAL INC|8-K|2013-03-20|edgar/data/1000045/0001193125-13-117362.txt
1000045|NICHOLAS FINANCIAL INC|SC 13G|2013-02-14|edgar/data/1000045/0000846087-13-000024.txt
1000045|NICHOLAS FINANCIAL INC|SC 13G|2013-02-14|edgar/data/1000045/0000902219-13-000333.txt
1000069|EMPIRIC FUNDS, INC|485BPOS|2013-01-15|edgar/data/1000069/0000894189-13-000203.txt
1000069|EMPIRIC FUNDS, INC|497J|2013-01-04|edgar/data/1000069/0000894189-13-000047.txt
1000069|EMPIRIC FUNDS, INC|DEF 14A|2013-02-08|edgar/data/1000069/0000894189-13-000769.txt
1000069|EMPIRIC FUNDS, INC|N-Q|2013-02-26|edgar/data/1000069/0000894189-13-001066.txt
1000069|EMPIRIC FUNDS, INC|PRE 14A|2013-01-08|edgar/data/1000069/0000894189-13-000099.txt
1000097|KINGDON CAPITAL MANAGEMENT, L.L.C.|13F-HR|2013-02-14|edgar/data/1000097/0000919574-13-001835.txt
1000097|KINGDON CAPITAL MANAGEMENT, L.L.C.|SC 13G/A|2013-02-14|edgar/data/1000097/0000919574-13-001837.txt
1000097|KINGDON CAPITAL MANAGEMENT, L.L.C.|SC 13G/A|2013-02-14|edgar/data/1000097/0000919574-13-001845.txt
1000097|KINGDON CAPITAL MANAGEMENT, L.L.C.|SC 13G|2013-02-20|edgar/data/1000097/0000919574-13-001964.txt
1000148|LEGACY CAPITAL FUND, INC.|X-17A-5|2013-03-01|edgar/data/1000148/9999999997-13-003516.txt
1000151|GREENWICH FINANCIAL SERVICES, L.L.C.|FOCUSN|2013-03-01|edgar/data/1000151/9999999997-13-006250.txt
1000151|GREENWICH FINANCIAL SERVICES, L.L.C.|X-17A-5|2013-03-01|edgar/data/1000151/9999999997-13-003350.txt
1000152|WESTERN INTERNATIONAL SECURITIES, INC.|FOCUSN|2013-02-27|edgar/data/1000152/9999999997-13-005488.txt
1000152|WESTERN INTERNATIONAL SECURITIES, INC.|X-17A-5|2013-02-27|edgar/data/1000152/9999999997-13-001772.txt
1000177|NORDIC AMERICAN TANKERS Ltd|20-F|2013-03-19|edgar/data/1000177/0001193125-13-113445.txt
```

Figure 4: A typical Master Index File
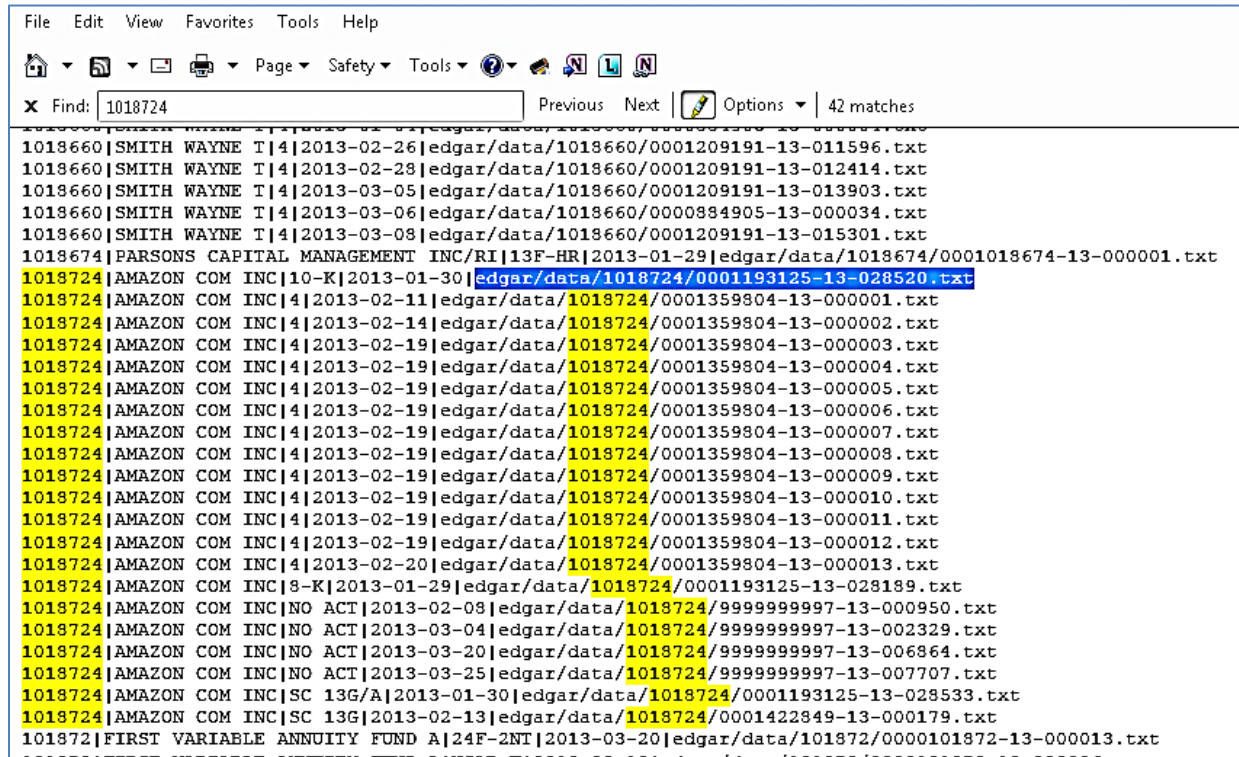
Figure 5: URL for 10-K filing is obtained from the Maser Index for AMAZON COM in first quarter of 2013.