Usage_and_Implementation_Notes

January 17, 2019

1 Usage Examples and Implementation Notes

This Notebook is meant to demonstrate a few usage examples of the application I developed for edgar debt scraping, as well as, provide some details and context around implementation.

A high level system diagram of the application is shown below. The system was designed so that individual files could be processed in streaming fashion, meaning that applicable 10Qs would be lazily located and processed into final results in iterative fashion.

1.1 Section 1: Imports

Make sure that the root directory is on the python path. May have to modify this depending upon where notebook is run from

```
In [1]: import sys
        sys.path.append("../") #make sure root edgarScraper directory is on pythonpath
        from edgarScraper.edgarDebtScraper import EdgarDebtScraper
```

1.2 Section 2: Usage Examples

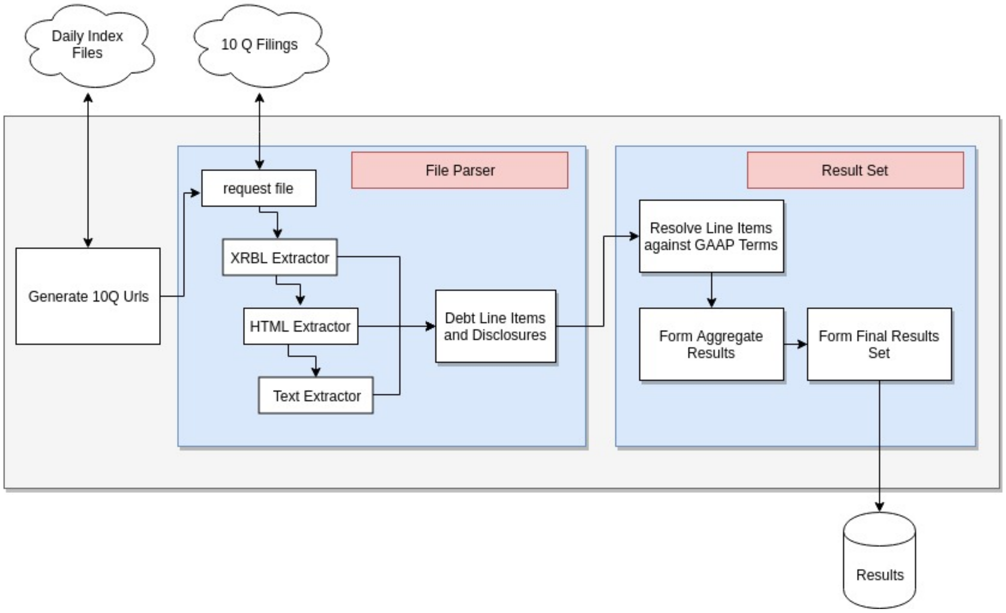
The main application is the EdgarDebtScraper object. It exposes a method called `runJob` which supports single and multiprocessing. Results can be built in memory and returned as dataFrames, or streamed to disk. Function docstring supplies more information.

```
In [2]: eds = EdgarDebtScraper()
        ? eds.runJob()
```

1.2.1 Example 1 - get 10Qs by year

The following cell shows an example job for processing the first 400 10Q files from the year 2010 using 4 processes. It should take on the order of 1-2 minutes to run and return two dataFrames. The first contains extracted line item information, the second contains free text debt disclosures. Logging output can be suppressed by changing the logging level set in `edgarScraper.config.log.py`

```
In [4]: #sample job - get results for the first 400 10Qs in 2010.  Takes ~ 2mins.
        debtDf, disclosureDf = eds.runJob(
            years = [2010],
            maxFiles = 400,
            nScraperProcesses=8
        )
```



title

```
2019-01-15 00:03:42,876 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-15 00:03:44,222 dailyIndLogger INFO Generated 25 10-Qs
2019-01-15 00:03:44,618 dailyIndLogger INFO Generated 50 10-Qs
2019-01-15 00:03:44,988 dailyIndLogger INFO Generated 75 10-Qs
2019-01-15 00:03:44,990 dailyIndLogger INFO Generated 100 10-Qs
2019-01-15 00:03:45,431 dailyIndLogger INFO Generated 125 10-Qs
2019-01-15 00:03:45,830 dailyIndLogger INFO Generated 150 10-Qs
2019-01-15 00:03:54,139 dailyIndLogger INFO Generated 175 10-Qs
2019-01-15 00:03:55,979 dailyIndLogger INFO Generated 200 10-Qs
2019-01-15 00:03:57,797 dailyIndLogger INFO Generated 225 10-Qs
2019-01-15 00:04:02,290 edgarScraperLog INFO finished consuming file 100
2019-01-15 00:04:05,708 dailyIndLogger INFO Generated 250 10-Qs
2019-01-15 00:04:08,929 dailyIndLogger INFO Generated 275 10-Qs
2019-01-15 00:04:14,067 dailyIndLogger INFO Generated 300 10-Qs
2019-01-15 00:04:17,058 edgarScraperLog INFO finished consuming file 200
2019-01-15 00:04:18,307 dailyIndLogger INFO Generated 325 10-Qs
2019-01-15 00:04:19,510 dailyIndLogger INFO Generated 350 10-Qs
2019-01-15 00:04:25,055 dailyIndLogger INFO Generated 375 10-Qs
2019-01-15 00:04:28,014 dailyIndLogger INFO Generated 400 10-Qs
2019-01-15 00:04:36,021 edgarScraperLog INFO finished consuming file 300
2019-01-15 00:04:49,768 edgarScraperLog INFO finished consuming file 400
2019-01-15 00:05:22,178 edgarScraperLog INFO Job Finished
```

1.2.2 Example 2 - Search by CIK

Another sample job with specific ciks. This changes the behavior of the scraper slightly. Since the daily index files provided by edgar aren't indexed by company name or cik, a distributed search is first conducted to find relavent 10Q filings. This list is eagerly evaluated and then passed for further file processing. Note: this is a consequence of my decision to not store the raw 10Q text files locally. There are pros and cons to this. A pro is that I don't have to do any raw data management - a definite advantage considering this is a prototype and there are GBs of data. A con is that the raw data can't be re-indexed for different use cases.

```
In [3]: # search for specific cik for years 2010-2018 inclusive.  Takes about 2-3 mins to find r
        debtDf, disclosureDf = eds.runJob(
            years = list(range(2010,2019)),
            ciks = [1062822],
            nScraperProcesses=4
        )

        debtDf.head()
```

```
2019-01-14 23:54:00,234 dailyIndLogger INFO CIKS specified, calling distributed search routine,i
2019-01-14 23:54:00,764 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:01,029 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:01,270 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:01,526 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:02,059 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:02,392 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:02,624 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:02,970 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:03,450 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:03,665 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:04,030 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:04,341 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:04,460 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:04,932 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:08,245 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:11,711 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:15,852 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:19,769 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:31,429 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:32,266 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:32,552 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:33,366 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:35,296 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:36,041 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:36,386 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:38,732 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:39,017 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:41,192 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:54:44,232 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:54:49,169 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:00,267 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:00,803 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
```

```
2019-01-14 23:55:08,023 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:10,724 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:15,697 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:16,336 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:26,690 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:32,587 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:34,517 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:34,894 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:35,440 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:39,827 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:40,291 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:40,664 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:42,063 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:48,606 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:49,748 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:49,860 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:50,385 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:53,772 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:55:54,665 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:55:59,891 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:56:00,216 dailyIndLogger INFO Searching for 10Qs in https://www.sec.gov/Archives/e
2019-01-14 23:56:01,897 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:56:07,690 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:56:12,176 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:56:14,558 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:56:19,822 dailyIndLogger INFO found 1062822 in https://www.sec.gov/Archives/edgar/
2019-01-14 23:56:30,674 dailyIndLogger INFO Found 22 filings for selected ciks
2019-01-14 23:56:38,329 edgarScraperLog INFO Job Finished
```

```
Out[3]:    ACCOUNTSPAYABLEANDACCRUEDLIABILITIESCURRENT  ACCOUNTSPAYABLECURRENT  \
        0                                   79416000.0              66553000.0
        1                                   63316000.0              50451000.0
        2                                       9832.0                  3547.0
        3                                      10141.0                  4773.0
        4                                       2984.0                  3550.0

           ACCOUNTSPAYABLEOTHERCURRENT  ACCOUNTSPAYABLERELATEDPARTIESCURRENT  \
        0                          NaN                                   NaN
        1                          NaN                                   NaN
        2                          NaN                                   NaN
        3                          NaN                                   NaN
        4                          NaN                                   NaN

           ACCOUNTSPAYABLETRADECURRENT  ACCRUEDLIABILITIESCURRENT  BANKOVERDRAFTS  \
        0                          NaN                 12863000.0             NaN
        1                          NaN                 12865000.0             NaN
        2                          NaN                     6285.0             NaN
```

```
3                          NaN                     5368.0             NaN
4                          NaN                     7437.0             NaN

   BRIDGELOAN  CAPITALLEASEOBLIGATIONSCURRENT  \
0         NaN                             NaN
1         NaN                             NaN
2         NaN                             NaN
3         NaN                             NaN
4         NaN                             NaN

   CAPITALLEASEOBLIGATIONSNONCURRENT              ...               \
0                                NaN              ...
1                                NaN              ...
2                                NaN              ...
3                                NaN              ...
4                                NaN              ...

  SENIORLONGTERMNOTES  SENIORNOTESCURRENT  SHORTTERMBANKLOANSANDNOTESPAYABLE  \
0                 NaN                 NaN                                NaN
1                 NaN                 NaN                                NaN
2                 NaN                 NaN                                NaN
3                 NaN                 NaN                                NaN
4                 NaN                 NaN                                NaN

   SHORTTERMBORROWINGS  SHORTTERMNONBANKLOANSANDNOTESPAYABLE  \
0                  NaN                                   NaN
1                  NaN                                   NaN
2                  NaN                                   NaN
3                  NaN                                   NaN
4                  NaN                                   NaN

   SUBORDINATEDDEBTCURRENT  SUBORDINATEDLONGTERMDEBT  UNSECUREDDEBTCURRENT  \
0                      NaN                       NaN                   NaN
1                      NaN                       NaN                   NaN
2                      NaN                       NaN                   NaN
3                      NaN                       NaN                   NaN
4                      NaN                       NaN                   NaN

   UNSECUREDLONGTERMDEBT  WAREHOUSEAGREEMENTBORROWINGS
0                    NaN                           NaN
1                    NaN                           NaN
2                    NaN                           NaN
3                    NaN                           NaN
4                    NaN                           NaN

[5 rows x 77 columns]
```

title

1.3 Section 3 Data Heirachy

The application attempts to find relavent debt-information for 71 different fields. These fields and the accompanying taxonomy are taken from information found on https://xbrl.us/.

Final short and long term debt levels are calculated based upon the following strategy:

1) If values exist for key fields like `LONGTERMDEBTNONCURRENT` or `DEBTCURRENT` return these values as the final long and short-term debt levels.

2) Else, attempt to form final results by aggregating up component subfields.

3) Finally, if the first two approaches fail, attempt to form results by taking values from parent-fields (usually total current / noncurrent liabilities) and subtracting "sibling-level" fields where applicable.

For more details on this aggregation logic please see the source code contained in `edgarScraper.pipelineIO.resultset.py`

As with other implementation decisions, there are pros and cons to the approach I took. A pro is that the logic employed closely matches standard GAAP Taxonomies and allows for a robust and systematic way of determining overall debt levels. Indeed, using this approach I was able to get viable values for close to 90% of all 10-Q filings from 1994-2018.

The disdavantage to this approach is that it can sometimes lead to apples-to-oranges type comparisons. For instance, for a given 10-Q, the only short term debt field recovered may be a company's total current liabilities - either because the company provided little information or extraction faired poorly. This value will likely overstate the company's short term debt (as it can include things like payroll and taxes). For a different 10-Q, a more granular short-term debt field may be the only one resolved. Under the scheme advanced, both values will appear as final short term debt levels.

1.3.1 Data Field Groupings and Hierarchy