Data crawling and knowledge construction

of EDGAR data

Project Instructor:

Dr. Kewei Sha

Project Mentor:

Catherine Yin

Group Members:

Amey Parab

Baljeet Singh

Priyanka Shah

Vishal Deshmukh

# **Acknowledgment**

We owe our profound gratitude towards Dr. Kewei Sha, our capstone advisor who has given us guidance and direction amid the term of our project. He held weekly meetings with us on the status of our undertaking. During our weekly meeting, he explored our task status and gave feedbacks with respect to technical, documenting and deliverable aspects of the project.

We additionally need to thank our mentor Catherine Yin who gave us the thought to our project and gave the appropriate directions and technological guidance to complete the project. We had successive meetings with him and his timely response and our agile method of development helped us to finish our undertaking in time.

Contents

[Acknowledgment 2](#_Toc481527931)

[Abstract 4](#_Toc481527932)

[Introduction 5](#_Toc481527933)

[Background 5](#_Toc481527934)

[Problem Definition 7](#_Toc481527935)

[Motivation 7](#_Toc481527936)

[Design 7](#_Toc481527937)

[1. Data Crawling and Downloading 8](#_Toc481527938)

[2. Creating Knowledge base 9](#_Toc481527939)

[3. Analysis 10](#_Toc481527940)

[Implementation 12](#_Toc481527941)

[1. Web crawler 12](#_Toc481527942)

[2. Sectionalization: 15](#_Toc481527943)

[3. Analysis 19](#_Toc481527944)

[Challenges 21](#_Toc481527945)

[Future Work 22](#_Toc481527946)

[Conclusion 22](#_Toc481527947)

[References 23](#_Toc481527948)

[Appendices 24](#_Toc481527949)

[Project Schedule 24](#_Toc481527950)

[Software and Hardware requirements 25](#_Toc481527951)

[Hardware Requirements: 25](#_Toc481527952)

[Software Requirements: 25](#_Toc481527953)

[Lessons Learned 25](#_Toc481527954)

# **Abstract**

It has become an era where everything is digitalized, which has increased ever more chances of utilizing such information in numerous ways.  Since most available data is in textual format, analysis of such data is quite challenging. SEC website has such EDGAR data which serves to enhance the productivity and fairness of the security market for the benefit of investors and corporations. All this information should be used properly to construct the knowledge base which will be utilized to analyze the data to build out report on the corporation. This will help in reducing the time to study about the firms, and will help the beneficiaries and law firms to take the important decisions without losing their valuable time.

The goal of this project is to crawl the archived EDGAR data using Python, and store it to MongoDB – a NoSQL database. This data will be further analyzed using Natural Language Processing techniques to construct the knowledge base. This will be helpful in scrutinizing the two major factors of company profile namely Risk Factors & Regulations. System will suggest missing or potential Risk factors and Regulations which should be considered in such aspects as business model/industry/jurisdiction/offering type. We will also be analyzing the Business model of each company to acquire the helpful knowledge for forthcoming companies which will happen to share similar industry domain.

# **Introduction**

Publicly owned companies, their officers and major investors are required to file regular disclosures with the Securities and Exchange Commission (SEC). To improve accessibility to these public documents, the SEC began developed the EDGAR (Electronic Data Gathering, Analysis and Retrieval) electronic disclosure system. This system provides ready, free access to all electronic filings made since 1994.

The recent trend for both the public and private sectors is to make information web-accessible. Putting data on-line leverages the universality of the Internet, improves user access, speeds the dissemination of information, and reduces costs for both the provider and user. The Securities and Exchange Commission (SEC), through its EDGAR (Electronic Data Gathering, Analysis and Retrieval) database initiative, was an early innovator in this area. The importance of the EDGAR database rests in the scope of the data it contains—disclosures of financial and operational performance of all publicly traded companies. It has been argued that under the Freedom of Information Act mandate, the Commission has an obligation to both promote and provide ready access to these documents [1,2].

EDGAR has become a valuable resource for both investors and the securities markets over the time. The data and information contained by EDGAR, is an asset itself. If properly accessed and studied, it can help various business to grow and can pace up the IPO filling process. Although access has been greatly improved, the ability to automatically analyze these filings is limited due to the semi-structured nature of the documents.

The insights of these documents always play an important role in IPO filing process. Most of the law firm experts spend their time in reading these documents. The data is vast and, reading this data is always a time-consuming task. This results in loss of valuable time of law experts.

Our project aims at automating this process which will aid the law firm employees to identify the major Risks associated with Business facts. These things are needed to be clearly mentioned in IPO filling document to increase the acceptance rate. The output of project will be an analyzed report, which will focus on Risks Factor related to the business based on the archived data from the similar businesses industry. Thus, it will help to reduce the time consumption and pace up the IPO filing process.

# **Background**

Since its inception in the mid-1930s, the primary mission of the SEC has been to protect investors and maintain the integrity of securities markets. As part of this effort, domestic, publicly held companies are required to disclose complete and accurate information about their operations, as well as any event that could materially impact them [3]. This required information is extensive. The SEC receives 12 million pages of documents annually [4]. Manual processing of this much information is both expensive and time consuming. Having to physically handle paper filings also limits the timely access to this important, public information.

1. SEC’s EDGAR database

‘‘The laws and rules that govern the securities industry in the United States derive from a simple and straightforward concept: all investors, whether large institutions or private individuals, should have access to certain basic facts about an investment prior to buying it. To achieve this, the SEC requires public companies to disclose meaningful financial and other information to the public, which provides a common pool of knowledge for all investors to use to judge for themselves if a company’s securities are a good investment.’’ [3].

To improve access to this information, the SEC developed the EDGAR system, currently in its 8th revision [6,5]. It has evolved to the point that it automates ‘‘the collection, validation, indexing, acceptance, and forwarding of submissions by companies and others who are required by law to file forms with the U.S. Securities and Exchange Commission (SEC). Its primary purpose is to increase the efficiency and fairness of the securities market for the benefit of investors, corporations, and the economy by accelerating the receipt, acceptance, dissemination, and analysis of time-sensitive corporate information filed with the agency’’ [7].

SEC maintains several types of forms as mentioned in the below table, these forms performs several tasks and serves many important function, for the investors as well as the companies.

Table 1: Common SEC Forms accessible through EDGAR

| **Form Usage** | **Form Name** |
| --- | --- |
| Annual Reports | 10K, 10-KSB, 10-K405 |
| Quarterly Reports | 10Q, 10-QSB |
| Special Reports | 8-K, 6-K |
| Proxy Filings | DEF 14A, PRE 14A |
| Insider Trading | 144, 3, 4, 5 |
| IPO Fillings | S-1, SB-1, F-1, 424B, SB-2 |
| Tender Offers | 14D-1 |
| Response to Tender Offers | 14D-9 |
| Mutual Fund Filings | N-1A, N-30D, 497 |
| Mergers and Acquisitions | 13D, 14D-1, 14D-9, S-4 |
| Employee Benefit Plans | S-8 |
| Secondary Stock Offering | S-2, F-2, S-3, F-3 |
| REITS (Real Estate Investment Trusts) | S-11 |
| Small Caps | SB-1, 10-KSB, 10-QSB |
| Registration Statements | S-3, 424B |
| Going Private | 13E3, 13E4 |

Out of the above-mentioned forms in the table, the below listed forms have been used for this project.

* **Form 8-K**: A report of unscheduled material events or corporate changes which could be of importance to the shareholders or to the SEC. It also notifies the public of events reported including acquisition, bankruptcy, resignation of directors or a change in the fiscal year.
* **Form 10-K**: A 10-K is a comprehensive summary report of a company's performance that must be submitted annually to the Securities and Exchange Commission. Typically, the 10-K contains much more detail than the annual report. It includes information such as company history, organizational structure, equity, holdings, earnings per share, subsidiaries, etc.
* **424B3**: A form of prospectus that reflects facts or events that constitute a substantive change from or addition to the information set forth in the last form of prospectus filed with the SEC.
* **424B4**: This form discloses information, facts or events covered in both forms 424B1 and 424B3.
* **424B5**: Discloses information, facts or events covered in both forms 424B2 and 424B3.

b. Database Access

A part from in-room data reading, EDGAR provides multiple ways to access this database. The SEC provides daily and quarterly index files called Master.idx file which contains the list of all filings, company names and link to actual forms for a quarter.

The column-wise information maintained in Master file is as follows:

Company name: Name of the Company

Form type: 8-K, 10-K, 424B3, 424B4, 424B5

CIK (Central Index Key): to uniquely identify the submitting company

Date filed: Date of the filing form

URL: Link where the full text of the filing can be obtained.

In this project, we have used quarterly based master.idx files to download the above-mentioned form types.

# **Problem Definition**

The vast amount of data within each filing, is very hard to read and is a time-consuming process. As the time passes, EDGAR data keeps increasing.

EDGAR’s filings, if studied properly might give important insights into the present financial conditions, for the respective industry sector, and companies. These insights can drive many of the important future decisions for a business, which can help the business to grow. These insights will also help the law firms in reducing processing time, which in result may help the respective stakeholders, customers, and the industries themselves.

# **Motivation**

As the technology is growing into various aspects of human life, it has always proved itself to help reduce human load. The drastic increase in usage of artificial intelligence, and machine learning has reduced human efforts.

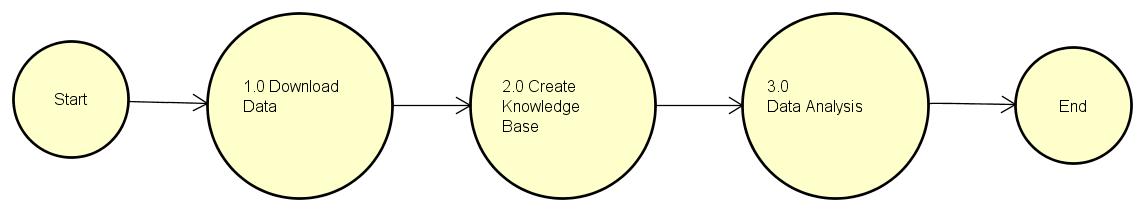
With our problem in view, the data is in textual format. With natural language processing and machine learning in mind, our problem can be solved. Using the various techniques discussed later, a system can be automated to solve this problem, by properly providing it a training dataset.

# **Design**

As most of the efforts spent on IPO process are mainly focused on SEC filling and creating EDGAR documents, law experts’ advice plays a big role in this process. Our product will aid the law firm employees in identifying the major Risks and Regulations that needs to be clearly mentioned in IPO filling document to increase the acceptance rate.

The output of project will be an analyzed report, which will focus on Risks Factor, and Regulations related to the business based on the archived data from the similar businesses industry. This project will also be helpful to understand the ongoing business of certain company, as well as will reverse engineer the company's profile based on the risk factors.

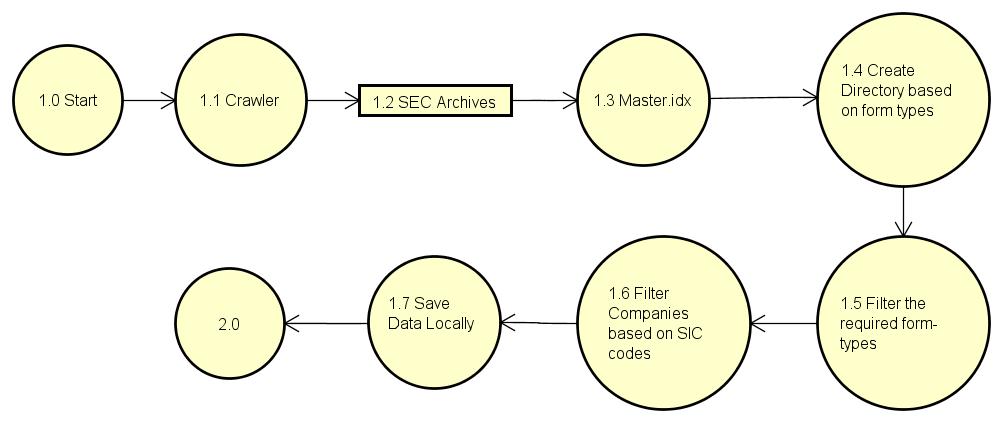
To get all these insights, we designed an approach by building from the foundation. This includes data downloading, followed by data extraction and then analysis as shown in the figure 1.

Figure 1. Process Flow

## Data Crawling and Downloading

To build the knowledge base, raw data is the most important thing needed. Gathering this raw data is the primary step for this project. As mentioned earlier, we used the FTP Access provided by SEC, to gather the data, which includes downloading of the above-mentioned forms viz. 8-K, 10-K, 424B3, 424B4, 424B5. These forms contain a lot of information, and are mostly in html format.

SEC has an archive of these forms and can be easily accessed by the master.idx file for each year and each quarter within the respective year. These master.idx files have company’s details for that quarter. These details consist of company’s CIK (Central Index Key) code, which is assigned by SEC uniquely, also SIC (Standard Industrial Classification) code, which helps in classification based of industry type. It also consists of form types, date filed, and company name, file name. All this fields are pipe ‘|’ separated.

Figure 2. Data Crawling and Downloading Process

We achieved this by designing a python based crawler, which sends the request to SEC website to connect to SEC archives. It then fetches the master.idx file for the defined year and quarter. For this project, we have gathered data till 2010.

The crawler reads the master.idx file for required form type. The crawler then creates the directory of form type for the quarter and places it in the respective year, thus creating nested directories. This data is saved locally.

The Crawler takes the following parameters as input:

* **Year**: This is one of the three input parameters used in data crawling. Master.idx files of this year or all four quarters of this year will be downloaded. Link to all forms are present in this master.idx file with the company code, company name, link to form, form type and data on which form is filed.
* **Form Type**: Downloading and analysis must be done on 5 form types: 424B3, 424B4, 424B5, 8-k and 10-k forms.
* **Industry type**: This is a parameter which specifies for which industry type download must be done. Currently, we are focusing on Biopharma and Coal industries.

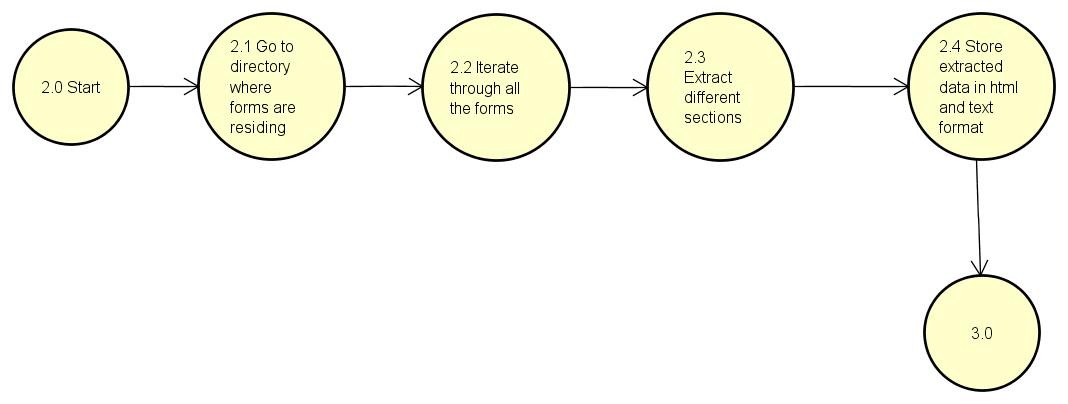
The crawler uses Python’s following libraries:

* Request: It sends HTML request to the SEC website
* BeautifulSoup: This library parses the HTML pages
* OS: This library is used for creating nested directories to store files per years and quarters locally.

## Creating Knowledge base

Creating a knowledge base from the raw data is always a challenging task. As discussed earlier, EDGAR has a lot of unstructured data, which acts as a biggest challenge while creating the knowledge base.

The second process is the crucial one, as cleaning the raw data and extracting the useful data is a challenging task. The Figure 3 shows the flow of this process.

Figure 3. Creating Knowledge Base Process

To achieve this, we designed python script, which takes the downloaded forms as input. From these forms, we have extracted:

* Risk Factors
* Regulations
* Business Section

Identifying these sections is a challenging part, as the forms of same type and same industry follows different HTML structure. One way of extracting Risk Factors is by considering the start of heading that begins with Risk factors and ends before the next section starts. “Regulation” has different term for different forms, like “Environmental and Other Regulatory Matters” and “Government Regulation and Product Approval”. All different terms should be considered for sectionalizing these sections.

As these forms are in HTML format, we extracted both the HTML as well as textual format for the sections. These extracted sections are then stored in MongoDB, which provides document like structure apart from common rows and columns database. So, our database contains all the Bold statements which are gist of the paragraphs followed, as well as the paragraph itself.

## Analysis

The extracted information from the raw data, serves as an input for this phase. We followed an approach by training a system, to automate this process. Where we used Machine learning algorithm, which will classify the different categories, and will provide the evaluation of test set. This approach required us to identify different categories, which we did with our understanding, as well as expert domain knowledge.

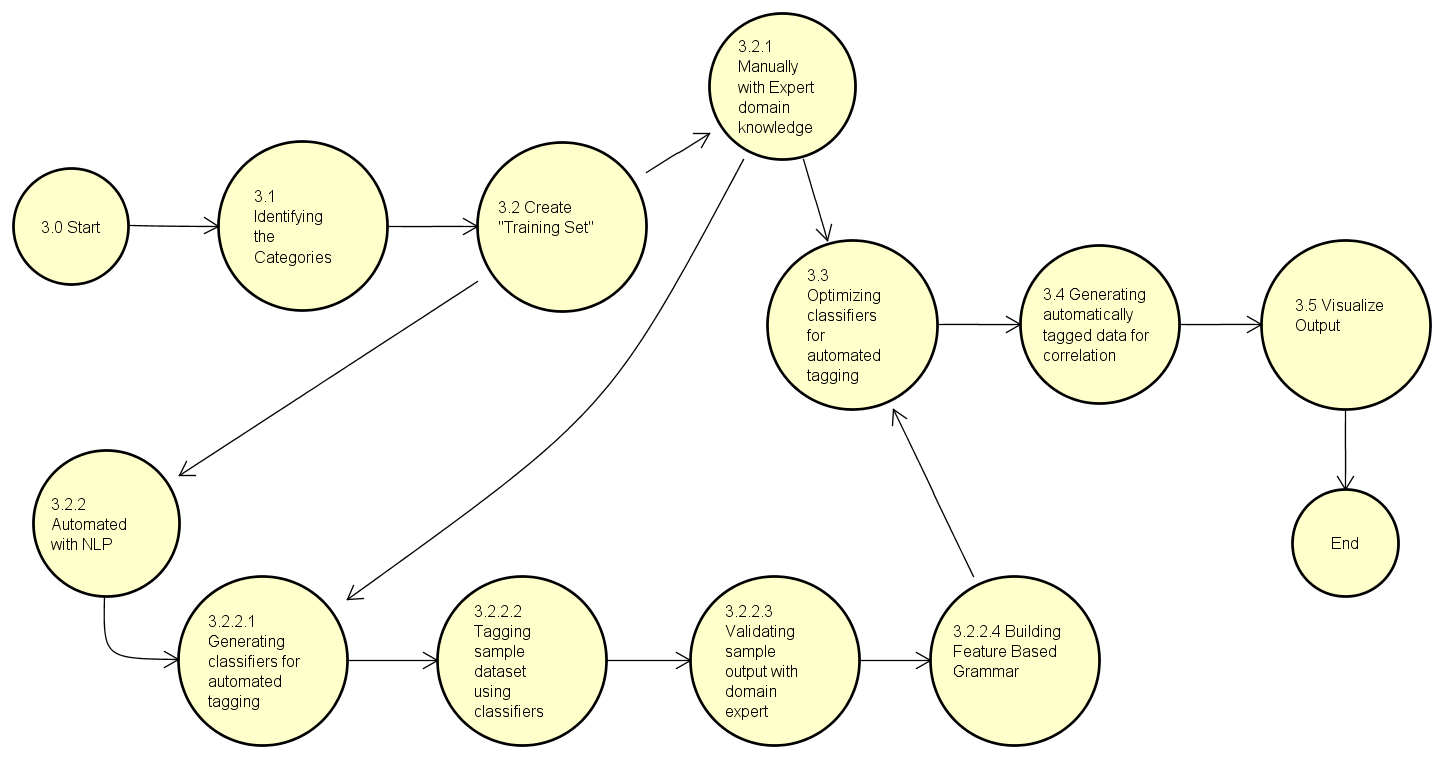


Figure 4. Analysis

In this approach we created a ‘training set’, which includes the extracted information, and tagged with the different categories which it belongs to. To create ‘training set’, we followed two ways: one included automating it with Natural language processing, and the other included doing in manually, using the expert domain knowledge.

The first way was further used to generate the classifiers for the automatically tagged data. These classifiers were further used to tag the ‘training set’. The sample output was validated using the expert domain knowledge. This tagged set was then used to build the feature based grammar. The grammar was used to optimize the classifiers for automated tagging, also we optimized the classifiers using expert domain knowledge.

Using supervised learning, optimized classifiers were then used to generate automatically tagged data for correlation.

# 

# **Implementation**

To implement this design for automating a system, we implemented it in four different modules by following Agile Methodology. The modules are as follows:

1. **Web crawler**
2. **Sectionalizing Forms**
3. **Form 8-K**
4. **Form 10-K**
5. **Form 424BX**
6. **Analysis**

We used python language, to implement these modules. Python is easy to implement analytical programs and tools. It also provides flexibility in using different kind of data structures. Python provides easy access to different API’s and have vast number of libraries for data mining, natural language processing, web scraping, machine learning, etc. which are necessary for this project.

As this project is based on a lot of unstructured data we used MongoDB, a NoSQL database. It is a document-oriented database and is horizontally scalable. It provides dynamic schema for storing semi structured or unstructured data, as the data involved in this project is mostly unstructured.

## **1. Web crawler**

Web crawler aims at downloading all the available forms on SEC website based on the input provided. These inputs include SEC code, Year, Quarter, Form Type and local folder path where forms are to be saved. SEC codes are basically the different types of industries, for example BioPharma industries has SIC code 2834. The year range can be starting from 1990 till 2017. Form type can be 10-K, 8-K, 424B3, 424B4 and 424B5.

Once the above inputs are provided to Web Crawler, it downloads the Master.idx file for each year and quarter. Master file contains information about all forms filed for a year and quarter. Each line in Master file contains these informations:

* CIK
* Company Name
* Form Type
* Date Filed
* Filename

CIK is the unique identity of companies that do their filings with SEC. Form Type specifies which form the company has filed. Filename is the link to form where form is hosted on SEC website.

Web crawler reads the Master file line by line and evaluates if this form should be downloaded. It basically checks if the form type of a CIK belongs to desired SEC code. If it belongs to the desired SIC code, then the form is downloaded locally on the system. Master file doesn’t contain information about the SIC codes. There is a link where we send HTTP request to the URL with CIK as parameter, then the information about the SIC code can be retrieved. This SIC code helps in determining about the company for which industry types it belongs to. The URL used for this purpose is https://www.sec.gov/cgi-bin/browse-edgar?CIK=PARAMETER.

If CIK belongs to desired industry type, then form can be downloaded locally by the help of Filename. This filename when appended with https://www.sec.gov/Archives/edgar gives you the actual link from where the form can be downloaded. We send HTTP request to this constructed URL and get response as HTML content. This HTML content is saved in HTML file with following naming convention:

SIC\_CIK\_Company Name\_Date Of Filing\_Form Type

local-path/2016/Q3/10-K/2834\_57725\_LANNETT CO INC\_2016-08-29\_10-K.html

This naming convention is followed to maintain the consistency and easier to identify which file belongs to any year, quarter, SIC, CIK and form type.

Technically, following steps are performed to achieve Web crawling in the python script:

* Input given as Year range, SIC codes, form types.
* For each year and quarter, Master index file is downloaded locally to the system. HTTP request is sent to this URL with year and quarter as a parameter: https://www.sec.gov/Archives/edgar/full-index/. The response of Master index file is saved with. idx extension. URLLIB2.request library is used for this purpose.
* Read the above downloaded master file line by line. Each line contains information about CIK, Form type and link to actual form. Before downloading the form, it is required to validate that this form belongs to SIC code. This can be done by sending HTTP request to https://www.sec.gov/cgi-bin/browse-edgar?CIK=PARAMETER. This gives the output as HTML content, which is parsed to check the SIC code from the HTML content.
* If the above step returns true, then the form can be downloaded locally. This can be done by sending HTTP request to https://www.sec.gov/Archives/edgar appended with file name. The request is sent via URLLIB2.request library and response is stored as HTML content. BeautifulSoup, html2text, re, os are some other libraries used for crawling.

These forms are stored in Nested directories based on year, quarter and form types:

2016

Q1

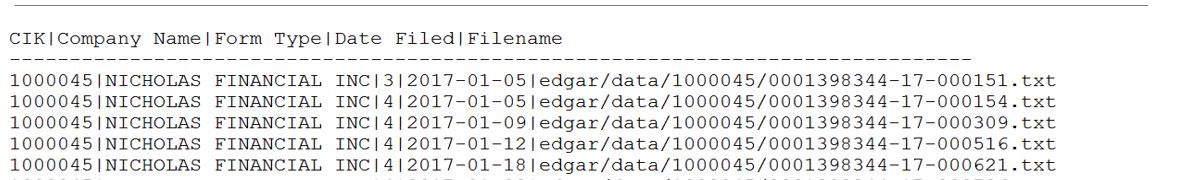
10-K

8-K

424B3

424B4

424B5

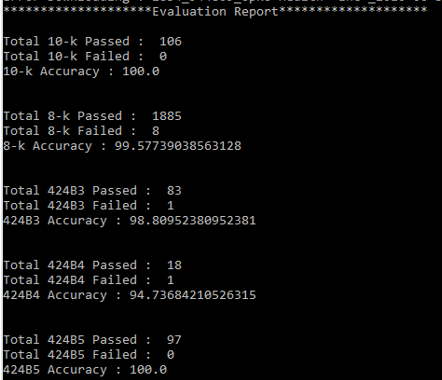
Master.idx file:

CIK-SIC mapping:



**Web Crawling accuracy:**

* For 10-k : 100%
* For 8-k : 99.57%
* For 424B3 : 98.80%
* For 424B4 : 94.73%
* For 424B5 : 100%

Note: When we cross verified for the forms that were not downloaded successfully, it was because of the broken link provided in Master file. 

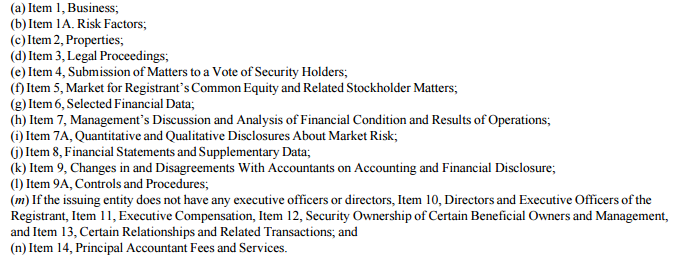
## **2. Sectionalization**

Data collected so far will be used for Analysis purpose. There are lots of sections in each form, but for analysis purpose, we just need:

* “Risk factors” and “Business” sections from 10-K
* “Risk factors” section from 424B3
* “Risk factors” section from 424B4
* “Risk factors” section from 424B5
* “Item no.” and “Exhibit no.” from 8-K

**Sectionalization 10-K:**

Typical structure of 10-K form:



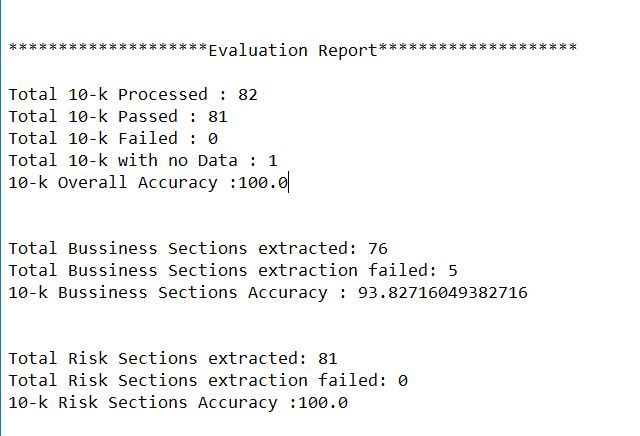
A Form 10-K is an annual report required by SEC which gives a comprehensive summary of a company's financial performance. The 10-K includes information such as company history, organizational structure, executive compensation, equity, subsidiaries, and audited financial statements. Companies with more than $10 million in assets and a class of equity securities that is held by more than 2000 owners must file annual and other periodic reports, regardless of whether the securities are publicly or privately traded. If a shareholder requests a company’s Form 10-K, the company must provide a copy. In addition, most large companies must disclose on Form 10-K whether the company makes it’s periodic and current reports available, free of charge, on its website. Form 10-K, as well as other SEC filings may be searched at the EDGAR database on the SEC's website.

From 10-K forms, we need “Risk factors” and “Business” sections for later analysis purpose. “Risk factor” section describes all possible and potential risks in investing in company’s stocks to customers and corporations. “Business” sections describe the company’s business model, organizational structure and financial circumstances of companies. These are the two very important sections in 10-k forms which are read by customers before they make any investments in the stock.

From all the forms that are downloaded locally, we will first look for 10-K forms. Then these forms are opened for reading the HTML content, parsed throughout to check if the line starts with “Item 1a Risk factors” or “Business”. Following steps are performed in the python script:

* 10-k Sectionalizer locates the 10-k directories in any year/quarter directories.
* It iterates through HTML content line by line
* It checks the starting of each HTML line, if the line starts with “Item 1a Risk factors” or “Business”, start index is stored.
* It proceeds further to check the start of each line to check “Risk factors” and “Unresolved Staff Comments” and store it as start index for Business and Risk Factor section respectively.
* HTML content for Business and Risk factor is extracted with the help of start and end indices.
* These HTML content is then considered for extracting bolds and plain text associated with bolds.
* All the above information is stored in MongoDB with the following attributes:
  + SIC
  + CIK
  + Company Name
  + Date
  + Form Type
  + HTML content of whole form
  + Risk Factor HTML content
  + Risk Factor Plain text
  + Risk Factor Bold Tags
  + Risk Factor Paragraphs
  + Business HTML content
  + Business Plain text
  + Business Bold Tags
  + Business Paragraphs

**10-K Sectionalization Accuracy:**

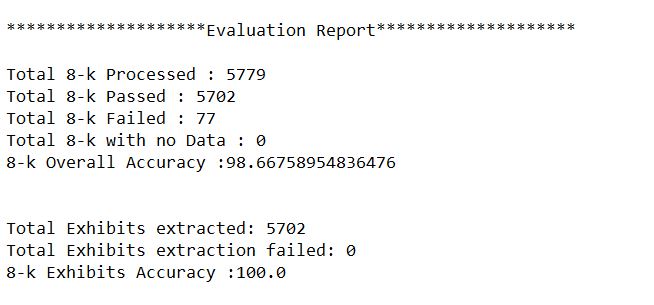


**Sectionalization 8-K:**

* Unlike 10-K forms, 8-K don’t have Risk factor and Business sections in it. They have items and exhibits in it. In addition to filing annual reports on Form 10-K and quarterly reports on Form 10-Q, public companies must report certain material corporate events on a more current basis. Form 8-K is the “current report” companies must file with the SEC to announce major events that shareholders should know about. 10-k Sectionalizer locates the 10-k directories in any year/quarter directories.

From all the forms that are downloaded locally, 8-K Sectionalizer looks for 8-K forms. Then these forms are opened for reading the HTML content, parsed throughout to check if the line starts with “Item” or “Exhibit”. Following steps are performed in the python script:

* It iterates through HTML content line by line
* Retrieve the text starting with the line having “item” until the next item found.
* If item number is “9.01”, then search for exhibit number and its description under item 9.01 section.
* The above process will continue until all item information found.
* Store all above information in MongoDB documents :
* All the above information is stored in MongoDB with the following attributes:
  + SIC
  + CIK
  + Company Name
  + Date
  + Form Type
  + HTML content of whole form
  + Item Number title and description
  + Exhibit information

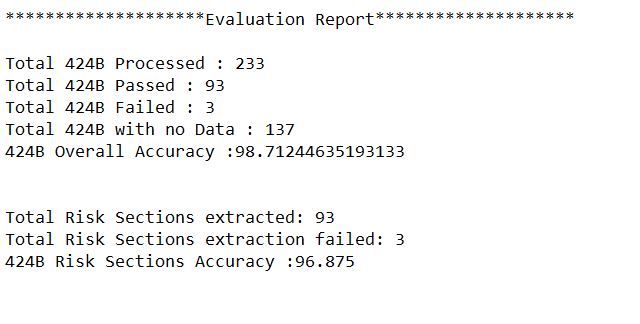
**8-K Sectionalization Accuracy:**

**Sectionalization 424B:**

The prospectus form that a company must file if it is making a primary offering of securities on a delayed basis. SEC form 424Bs must include information about the security being offered, including the price set for the public offering and the method of distribution.

From all the forms that are downloaded locally, 424B Sectionalizer looks for 424B (3, 4, 5) forms. Then these forms are opened for reading the HTML content, parsed throughout to check if the line starts with “Item” or “Exhibit”. Following steps are performed in the python script:

* Iterate through HTML content line by line
* Parse through Table of Contents and get the title of next section of the Risk Factors.
* Iterate through HTML content to extract data between Risk Factors and Title stored in the previous step.
* Separating all lines having font-Bold and text under each bold line for Risk Factors.
* Store all above information in MongoDB documents:
  + SIC
  + CIK
  + Company Name
  + Date
  + Form Type
  + HTML content of whole form
  + Risk Factor HTML content
  + Risk Factor Plain text
  + Risk Factor Bold Tags
  + Risk Factor Paragraphs



**Fact-Risk separation:**

Risk factor section of 10-K and 424Bs contains Risks and facts associated with those risks. Fact\_risk\_separation.py reads the Risk factors from the files and divides them into two categories, i.e. Risks and Facts based on the meaning of the sentences. Risks mentioned in these forms are the future possibilities of any unwanted circumstances that can impose risk of financial losses on customer’s investments. This is done mainly by the python’s NLTK package.

Risks are usually the future statements and contains the context of “possibility”. The sentence is tagged with Part of speech and then identified for Modals. A modal is a type of auxiliary (helping) verb that is used to express: ability, possibility, permission or obligation. Modal phrases (or semi-modals) are used to express the same things as modals, but are a combination of auxiliary verbs and the preposition to. These words are Can/could/be able to, May/might, Shall/should, Must/have to, Will/would.

We have utilized this feature to differentiate between Risks and Facts. If any sentence contains the Modal, then it is considered as Risk, otherwise fact. Furthermore, we identified the polarity of the risk on the scale of 0 to 1, 0 being the neutral risk and 1 being the most negative risk. These polarities can be used by customer which risks to consider before they make any investments.

**Key phrase extraction:**

It’s difficult to analyze the whole text of risk and facts from these forms and whole text is not important for text analysis. Key phrase extractor analyses the text and extracts the important key phrases which will later be used for analysis purpose. There are series of steps performed for text analysis, for example, removing all stop words from the text, lower the case of text and bring all words to their root form. We have used two libraries for this process: RAKE (Rapid Automatic Keyword Extraction) and IBM Watson library.

**Mongo to Excel:**

This script helps to export the data from MongoDB to excel files for visualization and manually tagging the text to its categories.

## **3. Analysis**

By this time, we have all the data which is important for analysis in MongoDB. Now, data can be pulled from MongoDB and analysis can be performed on it. Firstly, categories are identified for classification of EDGAR data. All EDGAR data will fall under one or more than one categories. The classification of the EDGAR data will be done with the help of automatic tagging algorithm. This will help in identifying the Business Facts in the individual risk factors. It’s important to identify the correlated Business and Risk factor text. Risk factor and Business text will said to be co related if they fall under same categories. This analyzed correlation will be utilized to optimize the future IPO filing process.

For analyzing the test data, natural language processing techniques have been used throughout the analysis. These techniques help in pointing out the potential risks in investment to any stock of a company. This is also helpful in analyzing risks between two companies.

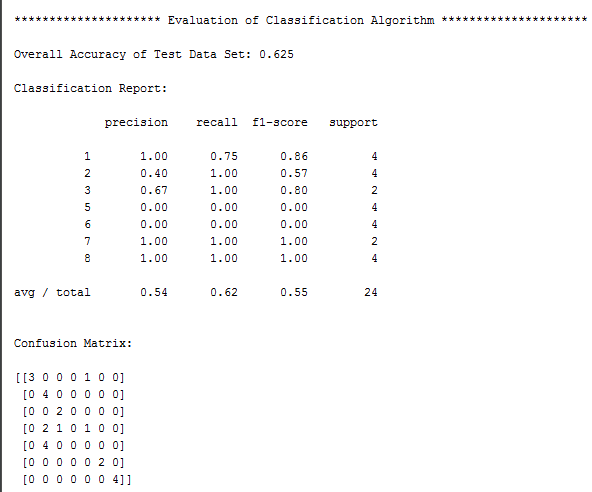
For identifying categories, domain knowledge is required of IPO filing process. The base training set is developed based on the training set manually. Following categories are considered for analysis purpose:

* Financial circumstances
* FDA processes
* Other INDUSTRY-RELATED regulatory measures
* Measures outside of the US jurisdiction/market
* GENERAL BUSINESS
* PRODUCTS
* GENERIC PUBLIC COMPANY LANGUAGE – ALMOST ALWAYS APPEAR IN EVERY COMPANY’S FILING – AND SUCH STRUCTURE
* INTELLECTUAL PROPERTY

Analysis of this risks factor is done by Supervised Classification Algorithm. The results from this classification algorithm are verified and the classifier keeps on improving in iterative manner. Main library which is used for Text classification is sciki-learn.

Steps performed for data analysis:

* Firstly, the manually tagged data is loaded to the program as the training set. This is present in excel files.
* For each category in training set, separate folders are generated by program, and the text which belongs to those categories are inserted in those categories.
* Above mentioned text is tokenized with scikit-learn and it convert occurrences to frequencies.
* Sklearn Naïve Bayes classifier is created and fed the training set.
* Tf-idf (Term frequency – inverse document frequency) is created for the above data. TF-IDF helps to show importance of a word in a category.
* Once the classifier is trained, test data is passed to get it tagged to categories provided. ‘
* Accuracy is measured against to check the percent of correct predictions made from all predictions.
* Confusion matrix is generated which helps in detailed analysis of output.
* Precision is evaluated based on the positive results divided by all results.

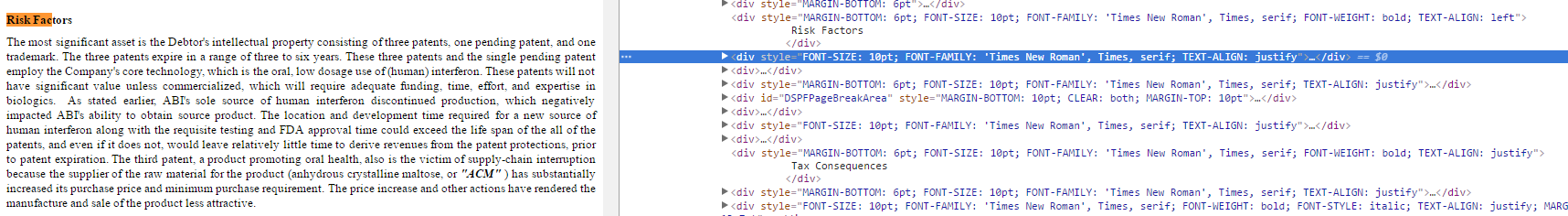


# **Challenges**

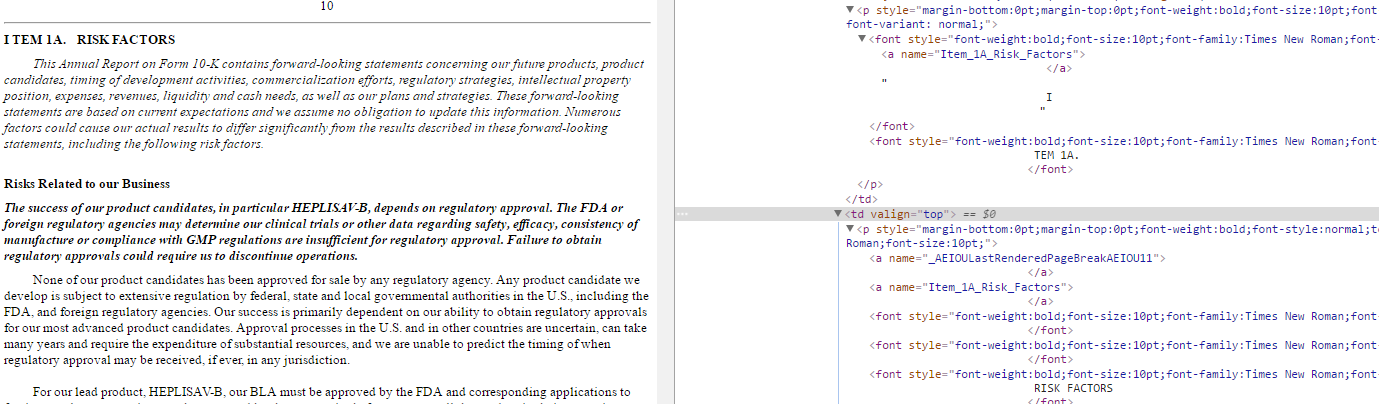
Challenges faced during Sectionalization of 10-K forms:

* Unstructured HTML data: There are millions of filings done by companies each year for 10-k forms. Different companies follows different HTML format structure. Mostly the forms don’t follow the nested structure, which would have been better for parsing. It follows the parallel division structure, that means all the paragraphs and sections are at same level which makes it difficult for extracting any particular section.

Parallel Division structure:



* Inconsistency in text of title: We are going line by line through HTML content to check for “Item 1a Risk factors” or “Business”. In some forms “Item 1a” and “Risk Factors” are in one line, while in some forms, they are in other lines. Some also have the word “item” split in different line like “I” in one and “tem” in other line. This makes very difficult to search for the titles of different sections.





# **Future Work**

This project can be further enhanced to correlate Business facts and Risk factors, which can also help the law firm experts to take important decision.

By applying the classification algorithm used for automatically tagging Risk Factors in 424B forms, correlation of different important aspects of business can be done. Correlating items and exhibits extracted from 8-K forms, can be helpful in improving the quality of 8-K forms, and so the insights.

Adapting designed modules of this project can also be used to download, sectionalize and analysis of other industries

# **Conclusion**

We developed python scripts to collect and sectionalize EDGAR data for analysis, using Natural Language Processing and Supervised Classification algorithm. We developed a system to automate risk factor tagging process. This automated tagging of risk factors will help Law firms to correlate the business of the industries to reduce the time and cost to optimize IPO filing process.

# **References**

[1] R. Nader, Statement of Ralph Nader before FOIndiana: FREEDOM OF INFORMATION, September 21, 1996, http://[www.cptech.org/govinfo/foindiana.html](http://www.cptech.org/govinfo/foindiana.html).

[2] Securities and Exchange Commission, SEC FOIA Program The Freedom of Information Act: What It Is, What It Does, October 9, 2001, <http://www.sec.gov/foia.shtml>.

[3] Securities and Exchange Commission, The Investor’s Advocate: How the SEC Protects Investors and Maintains Market Integrity, Mar. 1, 2001, <http://www.sec.gov/about/whatwedo.shtml>.

[4] Securities and Exchange Commission, Edgar Filer Manual, Release 5.10, SEC, Washington, DC, Sept. 1996.

[5] Securities and Exchange Commission, Edgar Filer Manual, Release 8.0, SEC, Washington, DC, Sept. 2001, http://[www.sec.gov/info/edgar/filermanual.htm](http://www.sec.gov/info/edgar/filermanual.htm).

[6] Securities and Exchange Commission, EDGAR Filer Manual v. 8.0, New Version: September 21, 2001, <http://www.sec.gov/info/edgar/filermanual.htm>

[7] Securities and Exchange Commission, Important Information about EDGAR, Sept. 28, 1999, <http://www.sec.gov/edgar/aboutedgar.htm>

# **Appendices**

## **Project Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Weeks** | **Tasks** | **Members** | **Date** |
| Week 1 | Project Assignment and group selection | Amey, Baljeet, Priyanka, Vishal | 01/17/2017 |
| Week 2 | Website Creation & Role Assignment | Amey, Baljeet, Priyanka, Vishal | 01/24/2017 |
| Week 3 | Requirement Understanding | Amey, Baljeet, Priyanka, Vishal | 01/31/2017 |
| Week 4 | Requirement Understanding | Amey, Baljeet, Priyanka, Vishal | 02/07/2017 |
| Week 5 | Data collection, Unit Testing, Mid-Term Presentation | Amey, Baljeet, Priyanka, Vishal | 02/14/2017 |
| Week 6 | Data Storage | Amey, Baljeet, Priyanka, Vishal | 02/21/2017 |
| Week 7 | Data Analysis | Amey, Baljeet, Priyanka, Vishal | 02/28/2017 |
| Week 8 | Data Analysis | Amey, Baljeet, Priyanka, Vishal | 03/07/2017 |
| Week 9 | Data Analysis, Unit Testing | Amey, Baljeet, Priyanka, Vishal | 03/14/2017 |
| Week 10 | Data Analysis, Unit Testing, Documentation | Amey, Baljeet, Priyanka, Vishal | 03/21/2017 |
| Week 11 | Data Analysis, Performance Testing, Documentation | Amey, Baljeet, Priyanka, Vishal | 03/28/2017 |
| Week 12 | Performance Testing, Documentation | Amey, Baljeet, Priyanka, Vishal | 04/04/2017 |
| Week 13 | User Acceptance Test, Documentation | Amey, Baljeet, Priyanka, Vishal | 04/11/2017 |
| Week 14 | Submitted Initial Draft | Amey, Baljeet, Priyanka, Vishal | 04/18/2017 |

## **Software and Hardware requirements**

### Hardware Requirements:

1. 8GB RAM
2. 500 GBStorage Space
3. I5 or Latest Processor

### Software Requirements:

1. Programming Language: Python
2. NoSQL database: MongoDB
3. MongoDB Visualization: RoboMongo 1.0
4. Python IDE: Pycharm, Anaconda Framework

## **Lessons Learned**

1. Domain Knowledge: As this domain was completely new to us, it was hard to understand the Forms, and their details. We learned about the different forms and how they work in IPO filing process.
2. We learned thoroughly the IPO filing process, and different events occurring during this process.
3. We learned about Supervised Machine Learning, classification using pythons’ machine learning library Scikit- Learn.
4. We thoroughly learned about the different python libraries such as rake, sentiment analyzer of nltk package, IBM Watson, etc., those were implemented as well as other which we considered to implement. We learned about the functionalities provided by all these libraries.
5. We learned the usage of MongoDB, how it is implemented and how it provides scalability and stores unstructured data, as compared to relational databases.