Information Retrieval on 10-K Filings

Anuneet Anand
Indraprastha Institute of Information Technology Delhi, India
anuneet18022@iiitd.ac.in

Isha Gupta
Indraprastha Institute of Information Technology Delhi, India
isha18040@iiitd.ac.in

Osheen Sachdev
Indraprastha Institute of Information Technology Delhi, India
osheen18059@iiitd.ac.in

Paras Mehan
Indraprastha Institute of Information Technology Delhi, India
paras18062@iiitd.ac.in

Uday Narayan Goel
Indraprastha Institute of Information Technology Delhi, India
uday18369@iiitd.ac.in

ABSTRACT

The financial statements of a company provide a lot of valuable insights about it. However, extracting crucial information from these statements is time-consuming and often quite cumbersome. We propose an interactive information retrieval system that can query the financial information of a company, summarise SEC 10-K filings and visualise a company's performance over the past few years. Our proposed system can be used by, but not limited to, investors, traders, analysts, auditors, and corporate executives.

1. INTRODUCTION

1.1 Motivation

In 2020, the world witnessed various new trends and technologies due to lockdowns and social distancing norms. People moved away from physical interactions to virtual meetings, digital tools, and platforms to carry out their daily tasks. One trend that caught our attention, in particular, is the rise in retail trading [10].

Today, retail traders contribute a significant share in trading volumes and are no longer controlled by institutions, banks, or big fund houses [9]. With rising retail participation and market volatility, there is also a rise of rumors and misinformation that can lead to people making wrong decisions. To help people make better decisions and prevent them from acting on erroneous information, the Securities and Exchange Commission of the United States (SEC) keeps an oversight on companies. It makes sure that they make proper disclosures regarding their activities. For this purpose, they maintain the "Electronic Data Gathering, Analysis, and Retrieval system" popularly referred to as the EDGAR database, which houses all the reports and disclosures a company makes with them.

This paper tries to make a novel attempt to retrieve essential and actionable insights from this database based on user queries to provide them with clean and single window access to all the vital information they may need about a company.

We will be working specifically on the dataset of **Form 10-K** that a company must file with the SEC. The annual report on Form 10-K (Fig. 5) provides a comprehensive overview of the company's business and financial condition and includes audited financial statements [1]. The 10-K follows a standard format specified by the SEC (Fig. 8) and typically includes more detailed information than the annual report to shareholders. They are supposed to describe their business, risk factors, give details of their financial statements, legal proceedings, stockholders matters, exposure to market risks, and supplementary information [2].

1.2 Problem Statement

Corporations working in the United States are supposed to file regularly with the SEC. Any information or notification of a company must be shared with the SEC. The SEC started maintaining a database of these documents in digital format ever since 1990. The database maintained by the SEC is publicly available, and frequently updated, and contains millions of documents filed to date. Investors can go to their website and lookup for the required documents(Fig. 6 & Fig. 7). However, accessing these documents is not user-friendly, as various forms filled with the SEC serve different purposes. Knowing the correct form is essential for navigating the database. Moreover, the texts are long, and not all information is relevant to the user and exists for just compliance purposes. The documents consist of texts and XBRL(for financial statements). The SEC has built a full-text search engine that uses HTML tags to navigate the fillings. However, the use of HTML tags is inconsistent across companies, and HTML does not support the semantics of structured retrieval and querying because it does not reveal the location of where the search word appears [3].

1.3 Proposed Solution

We intend to build an information retrieval system to identify and extract critical information from the 10-K document filings. The retrieved information will then be used to summarise different companies' financial performance over

five years. We would build an interactive dashboard for the same. The dashboard would also contain a natural language querying system to help the user analyse the company better. The major challenge in this task is to handle the extensive documents and ensure that queries can run in a reasonable time.

1.4 Dataset Description

We will be using the Electronic Data Gathering, Analysis, and Retrieval(EDGAR) system maintained by the Securities and Exchange Commission(SEC) of the United States. The corpus of documents comprises Form 10-K filled by top 100 companies by Market Capitalization. We have collected the records for the top 100 companies for the past five years, giving us a corpus of 500 documents. Each document comprises texts, images, financial reports in XBRL (eXtensible Business Reporting Language)(Fig. 9). The total database size is 14GB and the average document size being 28MB.

2. LITERATURE REVIEW

Our proposed system can be broken down into various individual components. Thus, our literature review aims to explore the current research in each of these components.

2.1 Segmentation and Information Extraction

Text segmentation is the task of dividing a document into contiguous segments based on its semantic structure. Information extraction is the task of automatically extracting structured information from unstructured and semi-structured machine-readable documents and other electronically represented sources. [3] discusses the study of text segmentation on 10-K Sec financial documents using a genetic algorithm without relying on a file's markup.

Since HTML markup is optional, not all files have a markup, and for even the ones that do, the markup is unreliable. As described earlier, each file follows specific guidelines regarding the sections to be present in a 10-K filing. However, these sections' names may vary slightly from file to file, and the order of these sections may also vary. This paper describes its problem statement as splitting the text in a filing into the sections described in the guidelines. The paper solves this problem by writing a genetic algorithm to segment the document into different sections listed in the guidelines. It defines the fitness function as the sum of the L2 norm of each section's size and a label similarity component (similarity of the current candidate section titles to those described in the guidelines). The approach used is extremely useful because this approach doesn't require any human annotations and the segmentation process is completely automated. However, the limitation of this paper is that it does not perform comparatively well in detecting all sections. For e.g., it's performance in detecting Part II and Item 11 is much lower than the performance on other sections of the 10-K filings.

2.2 Generating Word Embeddings

Word Embeddings from text corpus play an essential role in natural language processing. [11] discusses the generation of word embeddings from a text corpus of 10-K filings. More than 183k 10-K filings from the period 1993-2018 were processed for this task. The text corpus primarily focused on sections related to Risk Factors, Quantitative and Qualitative Disclosures about Market Risks, Management's Discussion, and Analysis. Skip-Gram Word2Vec model was implemented to generate the embeddings.

The quality of word embeddings was determined by checking their performance on sentiment word list of Loughran-McDonald [7]. The word list of Loughran-McDonald is one of the most popular financial lexicon and often used as a reference. The word embeddings were able to differentiate various types of sentiment words used in the financial domain. Cosine similarity was used to analyse relationships between financial words. A low cosine similarity for pairs of words, which have opposite meanings in the financial context, confirmed that word embeddings were generated correctly. However, the major limitation of this approach was that Word2Vec model generated embeddings which were context-independent and models like BERT might perform better as they generate context-dependent embeddings. Also, the embeddings were generated only for some specific sections of 10-K documents. There is a need to generate embeddings for other important sections too.

2.3 NLP to Extract Information

Financial statement metrics provide limited insights. It happens because, at times, there is important information contained in verbal discussions which are not captured by standard metrics. NLP is therefore used because (1) it has a low cost of processing large data contents (2) it detects latent features in data that are difficult to spot by manual analysis. [6] uses multiple NLP techniques to extract important pieces of information:

- **Key word searches and counts:** The authors parse a set of documents and get the word counts. Hoberg and Moon (2017) identify keywords related to offshore activities in 10-K filings to determine whether their frequencies are related to operational hedging.
- **Attribute dictionaries:** Possible misclassifications encouraged Loughran-McDonald to develop finance-specific dictionaries that are calibrated to the research question [7]. This had to be done because some words like liability are common financial terms but imply a negative connotation in the standard English dictionary. It is used to capture the tone of a statement and identify whether it leads to something positive or negative.
- **Naive Bayes classification:** It measures how likely the words in a corpus of documents convey a specific attribute. However, Naive Bayes classification's main limitations are that the building of training and testing data sets is costly and time-consuming.
- **Cosine similarity:** It uses the inner product to calculate the similarity between two documents. Hoberg and Lewis (2017) show that it is possible to extend the cosine similarity framework to create a measure of document similarity relative to a specific attribute.

2.4 Question Answering systems

Question Answering (QA) aims to provide precise answers in response to the user's natural language questions. An

Open Domain QA (OpenQA) is a system that tries to answer questions without any specified context. A traditional OpenQA system follows three stages:

1. Question Analysis: Reformulation of Question into search queries.
2. Document Retrieval: Searching for relevant documents and passages
3. Answer Extraction: Extracting answer from these documents

[12] gives a review of the various algorithms and techniques implemented in the three stages of OpenQA and how deep neural networks have been incorporated into these techniques. The paper also describes the modern QA system in a Retriever-Reader model, where the Retriever is an IR system. At the same time, Reader aims at inferring the final answer from the retrieved documents.
The Retriever can be of three types:

- Sparse retriever implementing classical IR techniques such as TF-IDF(Term Frequency-Inverse Document Frequency) and BM-25(Best Match 25).
- Dense retriever employing deep learning techniques which attain high accuracy and efficiency using interactions between tokens of questions and answers.
- Iterative Reader where the search for documents is conducted in multiple steps.

The Reader can be extraction based (predicting answer span from the retrieved documents) or generation based (generating answers in natural language).

The research reviewed in this paper gives us a fairly good understanding of how a question answering system is built and what all techniques can be applied to improve its performance. It highlights the limitations of existing Question answering system such as term mismatch and noisy retrieval and suggests techniques to overcome these limitations. The techniques described in this paper include query expansion and paraphrasing, question classification, changing the granularity of indexing (sentence level, paragraph level, document level) and training a dense retrieval model as a binary classification problem.

2.5 Analysis of 10-K Forms

[5] studied the correlation between text patterns of specific sections of 10-k forms of companies and respective sales patterns. The authors use text and data-mining on 10-K Forms describing risk factors in annual reports to investigate the relationship between the item's wording and the Compound Annual Growth Rate (CAGR) of revenue in the last three to five years. They wanted to investigate that if a company's financial performance is good, then are there any specific patterns in the business text of the 10-K Form. The authors used the entire text of Item 1A. Risk Factors and the revenue information of Item 6 Selected Financial Data of the companies' 10-K Form for the analysis. Using the forms, they tested the following four hypotheses:

1. Category 7370 containing Google, Facebook, Twitter, etc. is better than category 7373 containing Yahoo in revenue performance.
2. (a) Companies with low revenue performance tend to write a shorter "Risk Factors" item section, to skip over risks.
   (b) Companies with high revenue performance tend to write a shorter "Risk Factors" item section because they think there are fewer risks in their business.
3. (a) In item Risk Factors, a positive/negative tone correlates with sales performance.
   (b) Using the positive/negative text analysis results in the Risk Factors item section, we can group companies by sales performance.
4. The occurrence patterns of words in Risk Factors are correlated with sales performance.

They used sentiment analysis, correlation analysis, TF-IDF, and term-document matrix (TDM) for testing the four hypotheses. The authors verified Hypothesis 1 and hypothesis 2, they found some evidence of a correlation between sales performance and text statistics, but they were not able to verify the same. The authors rejected hypotheses 3 and 4 since they could not find any significant correlation in both cases, one of their main limitations. They couldn't prove the relation between the "Risk Factors" section and revenue performance. But still, the authors concluded that companies with good financial performance and bad financial performance often use different words. Therefore, it is essential to analyze the words that appear predominantly in the business text in predicting a company's future sales performance.

3. METHODOLOGY

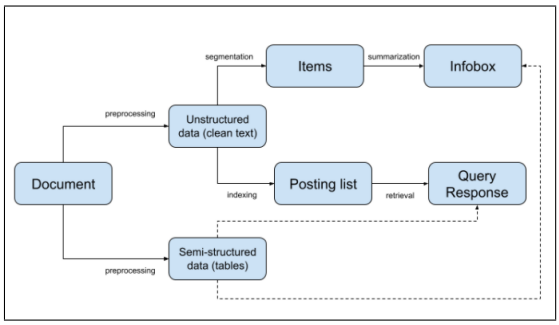


Figure 1: Brief description of Methodology

Our methodology is given in Fig. 1. A detailed explanation of the same is as follows:

- **Dataset Creation:** The corpus of documents comprises Form 10-K filled by top 100 companies by Market Capitalization for 5 consecutive years.It was scraped from the Electronic Data Gathering, Analysis, and Retrieval(EDGAR) system maintained by the Securities and Exchange Commission(SEC) of the United States. We have collected the records for the top 100 companies for the past five years, giving us a corpus of 500 documents. Each document comprises texts, images, financial reports in XBRL (eXtensible Business Reporting Language)(Fig. 9). The total database size is 14GB and the average document size being 28MB.

- **Preprocessing:** Each document had all the forms that the company has filed with the SEC attached to it. So the first thing we did was using the Beautiful Soup parser, we extracted the 10-K document from each document and saved it as a separate HTML file. Then we extracted the text from the HTML.

- **Segmentation:** Since each document is over 100 pages, we had to identify important sections and segment them. We decided to segment documents based on items present in the document using rule-based methods such as regular expression matching.

- **Indexing:** An inverted index is created from the preprocessed data, considering each 10-K filing as a document. The index stored the term frequency of each term in each document in which it occurs to provide support for TF-IDF. The generated index is pickled for usage.

- **Retrieval:** We tried three different approaches for retrieval : BM25, TF-IDF and log(TF)-IDF. Since TF-IDF performed the best among the three, we incorporated it in the final system. We use the document-level index to retrieve the most relevant documents for the user's query based on TF-IDF scores. Once we have a shortlisted set of documents, we retrieve the most relevant sentences to the query from these documents. Figure 2 shows the an example of the search results on the application.

- **Analysis:** Figure 3 shows the analysis we generate for a company from the 5 year filings of that company. Figure 4 shows the tables extracted from the documents.

  - **Data Analysis:** We have extracted information such as Balance Sheets, Cash Flows and Income Statements from each document. For each company we have provided a 7 year analysis and comparison of Net Income, Assets, Liabilities, Equities, etc for the last 7 years of that company.
  - **Sentiment Analysis:** We attempted sentiment analysis for Item 7 to determine how positive is the company about its progress in the coming years. We used the Python library TextBlob to find Polarity and Subjectivity of text from textual data present in Item 7. Polarity is a float in [-1,1] where 1 means a statement with positive sentiments and -1 means a statement with negative sentiment. Subjectivity is a float in [0,1], where a high value indicates that the statement is affected by emotions and personal opinions. The 5 year sentiment analysis graphs would help a market analyst get a summary of the company's position in the market without actually reading the reports completely.
  - **Ratio Analysis:** From the financial reports retrieved, we calculate assets, liabilities, equity, net income, return on assets and return on equity. We also provide a short summary for these metrics which consists of a concise and insightful year-on-year comparison.

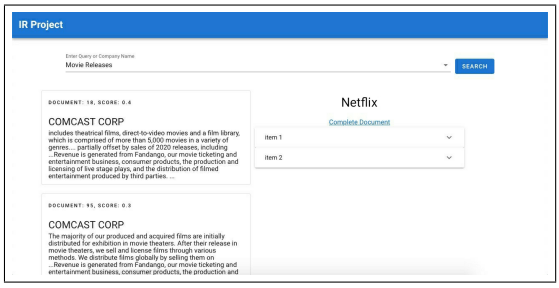
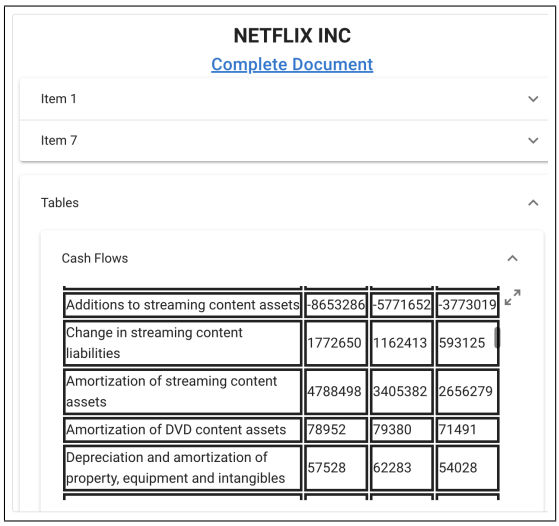


Figure 2: Frontend UI - Search

Figure 3: Frontend UI - Graphs



Additions to streaming content assets	-8653286	-5771652	-3773019
Change in streaming content liabilities	1772650	1162413	593125
Amortization of streaming content assets	4788498	3405382	2656279
Amortization of DVD content assets	78952	79380	71491
Depreciation and amortization of property, equipment and intangibles	57528	62283	54028

Figure 4: Frontend UI - Table

- **Summarization:** We generate summaries for 'Item 1' and 'Item 7' of each document. For generating the summaries we use a pre-trained BERT model and trained an unsupervised extractive document summarization like [8] and [4]. Since the pre-trained BERT model performed better, it is being used in the final system.

- **Application Development:** The application has a Client-Server architecture. A Python based backend is deployed to run the models and the IR system. The frontend is made in Vue.js which makes HTTP API requests to the python backend, fetches the data and renders it for the user in an accessible manner.

Table 1: Evaluation of IR System

Method	Precision@3	Precision@5	Precision@10
TF-IDF	0.78	0.74	0.54
log(TF)-IDF	0.48	0.46	0.41
BM-25	0.66	0.65	0.51

4. EVALUATION

Evaluation of our system has been performed on Form 10-K filled by top 100 companies by Market Capitalization for 5 consecutive years. We chose 20 queries, out of which 15 were generic queries, and 5 were company-specific queries. A list of queries is provided in the appendix. For evaluation, we chose Precision@K as our evaluating method. We manually examine the output of each query for each retrieval system and check whether the document is relevant or not. We check the relevance of the top 10 documents and calculate the precision at 3,5 and 10. We provide the results in Table 1.Based on the evaluation, we chose TF-IDF as our main method as it gave highest precision values.

The system is able to process natural language queries in the finance domain. We can also search for a specific company by its name. Proper error checks have been implemented so that the system does not crash if some data is missing in a document. Link to the original SEC 10-K is also provided, in case someone wants to read more about a company.

5. CONCLUSION

We were able to successfully implement a query and analysis system for SEC 10-K Filings based on TF-IDF. We believe that this system would help investors, traders, analysts, auditors, and corporate executives. The system can be improved further by using a more robust pre-processing pipeline and trying different retrieval models.

The system currently operates on an index made for 500 10-K documents and is easily deploy-able. The response time of the system is decent on the local machine. We have maintained a pre-processing and indexing pipeline to handle any newly filed 10-K document. Thus, We can easily scale the system by re-indexing and allocating more computational resources for it.

6. CODE

The application has been developed using version control with Github. Link to the repositories:

- Frontend
- Backend

7. REFERENCES

[1] How to Read a 10-K/10-Q | Investor.gov.

[2] SEC.gov | Current EDGAR Filer Manual (Volumes I - II).

[3] J. Carroll and T. Y. Lee. A genetic algorithm for segmentation and information retrieval of sec regulatory filings. In *DG. O*, pages 44–52. Citeseer, 2008.

[4] A. Esteva, A. Kale, R. Paulus, K. Hashimoto, W. Yin, D. Radev, and R. Socher. Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization, 2020.

[5] B. Lee, J.-H. Park, L. Kwon, Y.-H. Moon, Y. Shin, G. Kim, and H.-j. Kim. About relationship between business text patterns and financial performance in corporate data. *Journal of Open Innovation: Technology, Market, and Complexity*, 4(1):3, 2018.

[6] C. Lewis and S. Young. Fad or future? automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, 49(5):587–615, 2019.

[7] T. Loughran and B. McDonald. The use of word lists in textual analysis. *Journal of Behavioral Finance*, 16(1):1–11, 2015.

[8] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, 2016.

[9] B. Pisani. Trading volume for electronic brokers doubled last quarter and shows no signs of letting up, May 2020.

[10] P. Sarah and G. Katherine. The blistering rise of retail trading, buyings tocks how robinhood, e-trade, schwab and others are changing markets. *Bloomberg.com*, October 2020.

[11] S. Sehrawat. Learning word embeddings from 10-k filings for financial nlp tasks. *Available at SSRN 3480902*, 2019.

[12] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua. Retrieving and reading: A comprehensive survey on open-domain question answering, 2021.

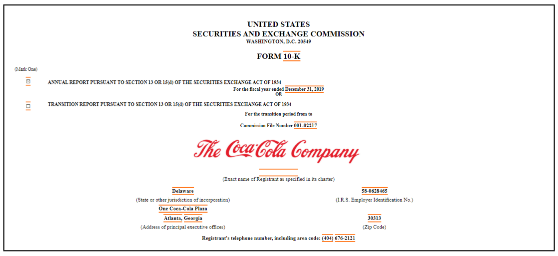
APPENDIX

A. EVALUATION QUERIES

1. Movie Releases
2. affected by covid-19
3. competitors of amazon aws
4. electric vehicle sales
5. presence in asia
6. sales in europe
7. SNEAKERS
8. oil exploration companies
9. chip manufacturers
10. cloud revenue
11. facebook daily users
12. chemical and fertilizers
13. class 8 truck sales
14. greater china sales
15. iphone Sales
16. car insurance premiums

17. berkshire premium receipts
18. coca cola bottling revenue
19. lockheed martin revenue segments
20. morgan stanley trading revenue

B. FIGURES



UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
WASHINGTON, D.C. 20549
FORM 10-K

Figure 5: Form 10-K of The Coca-Cola Company

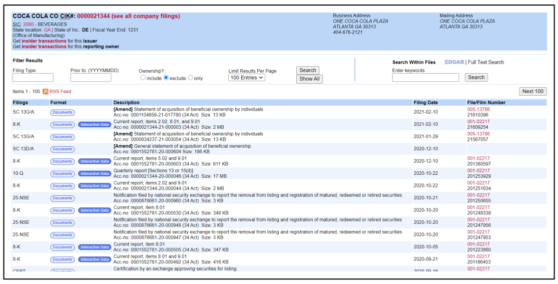


Figure 6: Snapshots of documents submitted by The Coca-Cola Company

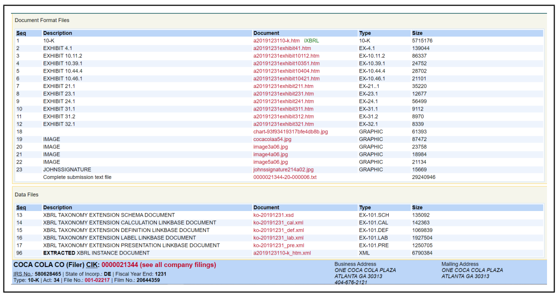


Figure 7: Submission File of Form 10-K

Figure 8: Contents of Form 10-K

```
<SEC-DOCUMENT>0000021344-20-000006.txt : 20200224
<SEC-HEADER>0000021344-20-000006.hdr.sgml : 20200224
<ACCEPTANCE-DATETIME>20200224160817
ACCESSION NUMBER:               0000021344-20-000006
CONFORMED SUBMISSION TYPE:      10-K
PUBLIC DOCUMENT COUNT:          126
CONFORMED PERIOD OF REPORT:     20191231
FILED AS OF DATE:               20200224
DATE AS OF CHANGE:              20200224

FILER:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 COCA COLA CO
                CENTRAL INDEX KEY:                      0000021344
                STANDARD INDUSTRIAL CLASSIFICATION:     BEVERAGES [2080]
                IRS NUMBER:                             580628465
                STATE OF INCORPORATION:                 DE
                FISCAL YEAR END:                        1231

        FILING VALUES:
                FORM TYPE:              10-K
                SEC ACT:                1934 Act
                SEC FILE NUMBER:        001-02217
                FILM NUMBER:            20644359

        BUSINESS ADDRESS:
                STREET 1:               ONE COCA COLA PLAZA
                CITY:                   ATLANTA
                STATE:                  GA
                ZIP:                    30313
                BUSINESS PHONE:         404-676-2121

        MAIL ADDRESS:
                STREET 1:               ONE COCA COLA PLAZA
                CITY:                   ATLANTA
                STATE:                  GA
                ZIP:                    30313
</SEC-HEADER>
<DOCUMENT>
<TYPE>10-K
<SEQUENCE>1
<FILENAME>a2019123110-k.htm
```

Figure 9: Contents of Form 10-K in Text Format