INTELLIGENT PORTFOLIO MANAGEMENT VIA NLP ANALYSIS OF FINANCIAL 10-K STATEMENTS

Purva Singh[1]

[1]School of Computer Science and Engineering, VIT University, Vellore
singhpurva2906@gmail.com

ABSTRACT

The paper attempts to analyze if the sentiment stability of financial 10-K reports over time can determine the company's future mean returns. A diverse portfolio of stocks was selected to test this hypothesis. The proposed framework downloads 10-K reports of the companies from SEC's EDGAR database. It passes them through the preprocessing pipeline to extract critical sections of the filings to perform NLP analysis. Using Loughran and McDonald [6] sentiment word list, the framework generates sentiment TF-IDF from the 10-K documents to calculate the cosine similarity between two consecutive 10-K reports and proposes to leverage this cosine similarity as the alpha factor. For analyzing the effectiveness of our alpha factor at predicting future returns, the framework uses the alphalens library to perform factor return analysis, turnover analysis, and for comparing the Sharpe ratio of potential alpha factors. The results show that there exists a strong correlation between the sentiment stability of our portfolio's 10-K statements and its future mean returns. For the benefit of the research community, the code and Jupyter notebooks related to this paper have been open-sourced on Github[1].

KEYWORDS

Natural language Processing, stock market, textual analysis, TF-IDF, cosine similarity, intelligent portfolio management.

1. INTRODUCTION

Natural language processing (NLP) is an intersection of computation, intelligence, and statistics whose primary goal is to automate the understanding of semi-structured language leveraged by humans. Computers have utilized NLP to use human language to generate conversational text, extract meaningful words, summarize long documents, build intelligent question-answering systems, detect spam emails, and leverage standard intelligence technologies [1]. But the scope of NLP has expanded beyond its use for text-generation and document summarization. There are multiple use-cases of NLP evolving in the financial sector and bots to provide financial advice, predict stock movements, personalized finance advisory through recommendation systems, and many more [2]. One primary application of NLP in the FinTech domain is analyzing various financial articles, daily reports on the stock market, and social media posts or tweets to predict a particular stock's future returns or an entire portfolio.

A 10-K document is a comprehensive report filed annually by a publicly-traded company about its financial performance [20]. A typical 10-K financial document contains five distinct sections: business, risk factors, selected financial data, management's discussion, analysis, financial statements, and supplementary data. The SEC requires this report to keep investors

[1] https://github.com/purvasingh96/StockGram-Intelligent-Portfolio-Manager

aware of a company's financial condition and allow them to have enough information before they buy or sell shares in the corporation, or before investing in the firm's corporate bonds [20].

Various fundamental analysts investigate all relevant information regarding the stocks that they follow, including financial statements required by the regulatory authorities, for example, 10-K documents published annually or 10-Q papers published quarterly. A quantitative analyst who covers approximately 30-40 stocks might have to go through a lot of paperwork for their respective research work. In this paper, the proposed framework targets to mitigate this pain point of a quantitative analyst. The framework leverages NLP techniques using which a quantitative analyst can quickly analyze selective sections from 10-Ks and 10-Qs reports. Automated textual analysis and information retrieval using NLP techniques would also help analysts process large volumes of financial data in less time and cost. NLP techniques can also help quantitative analysts extract latent information in these documents that might predict the company's future mean returns.

To leverage text in financial statements as an input for developing trading strategies, we need to convert the financial 10-K reportings into quantitative data. We condense the selected 10-K filings into a bag of words and categorize each word using the *Loughran McDonald sentiment word list,* specially curated for performing textual analysis in finance [6]. Now to shape the importance of repeated observations within a single document, while reducing the impact of common words among a collection of documents, the framework leverages Term Frequency - Inverse Document Frequency (TF-IDF) [7]. TF-IDF converts our 10-K reports into a group of document vectors for further processing.

A significant change in the financial document from one period to the next might indicate changes in the market or company structure. It could lead to additional risk for underlying stocks. Thus, it would be beneficial to compare the 10-K documents over time using similarity metrics [8]. This similarity metrics would convert two papers into a number representing a degree of similarity. In this paper, we have leveraged cosine similarity to calculate the similarity score between two 10-K documents over a time period and use it as an alpha factor that has the potential to predict future mean returns of our portfolio. The framework utilizes Alphalens, an open-source tool developed by the Quantopian Research Platform, to analyze alpha factors' predictive power [21]. Upon performing factor return analysis, turnover analysis, and comparing the Sharpe ratio of the potential alpha factors, the paper demonstrates that the documents that were expressive of a *stable positive* sentiment yielded maximum returns compared to forms that expressed *negative*, *litigious*, or *constraining* sentiment.

The organization of the rest of the article is as follows. Section II gives an overview of empirical models that have leveraged financial reporting documents for performing textual analysis to generate trading strategies and documents that utilize sentiment analysis for predicting future mean returns of the stock portfolio. Literature Survey summarizes the general forms of the model and their advantages and disadvantages. Section III delineates basic terminologioes related to this paper and Section IV describes the problem statement, which this paper aims to mitigate. The proposed framework uses a specific dataset as its input, described in Section V,

followed by Section VI, defining the framework's methodology. Section VII evaluates our hypothesis by conducting factor return analysis, turnover analysis, and comparing the Sharpe ratio of potential alpha factors. Section VIII provides the final summary and essential findings of the paper. It also provides suggestions for improving the framework's performance and future directions.

2. Literature Survey

Textual analysis is blooming in the field of finance. This section briefly performs a literature survey on content analysis models that leverage the most popular methods and discuss papers on sentiment analysis relevant to our study. The most commonly leveraged approach for performing textual analysis on financial reports consists of two steps. The first step is to categorize each word in the preprocessed financial statements into positive or negative sentiment [12]. Colm Kearney and Sha Liu (2013) have conducted an extensive survey on literature that leverages textual analysis to extract sentiment from sources such as social media, corporate disclosures, and media articles. They have also reviewed how each source of data and the chosen linguistic method has idiosyncratic advantages and disadvantages that drive research focus [10]. Tetlock et al. (2008) have leveraged Harvard Psychosociological Dictionary to classify financial words as positive or negative [13]. However, Loughran and McDonald (2016) have published a detailed review of textual analysis, where they have introduced five new word lists that recognizes the tone of a financial document. This word list is better than the widely used Harvard Dictionary, which categorized more than three-fourth of the words as *negative* that were typically not considered *negative* in the economic context [6]. Feng Li, Russell Lundholm, and Michael Minnis (2013) leverage financial 10-K reports to measure a firm's competitive environment by counting the term competition's frequency in its financial reporting document [11].

The second step in the textual analysis is to allocate appropriate weights present in the chosen sentiment word list, enabling the quantitative algorithm to convert textual content from these financial reports into a quantitative score. Jegadeesh and Wu (2013) proposed a novel approach to leverage market reactions to 10-K filings to determine each word's term weights and tone. Their system does not depend on word list choice as they recorded similar results after omitting half of the words in word lists [12]. Loughran and McDonald (2011) have shown that term-weighting schemes that assign different weights on terms in a document can improve the model's fitment [6].

Performing sentiment analysis on financial statements has seen a surge in its usage to extract information related to stock prices, firm fundamentals, and overall stock price movement [14]. Smailović *et al.* utilized supervised machine learning to create a sentiment classifier to categorize financial tweets from the Twitter platform into three categories: positive, negative, and neutral. They leveraged public opinions on the stock market made online on the Twitter platform to analyze the correlation between positive sentiment movements and the corresponding closing price of the stock [15]. Tushar Rao *et al.* have leveraged the Naive Bayes classifier to categorize tweets related to financial data into positive and negative sentiments. They analyze the correlation between a stock's short-term performance and a broad set of public

opinions on financial data available on Twitter [16]. Pagolu *et al.* have leveraged Word2Vec representations of public opinion expressed related to financial news on the Twitter platform and combined those representations with Random Forest Classifier for creating their sentiment analysis engine. Their experiments have shown a strong correlation between the movement of stock prices and public opinion of that stock expressed through tweets [17].

3. BASIC TERMINOLOGIES

This section deliniates the basic terminologies that will be used further in the paper to explain the methodology and evaluation results.

3.1. FACTORS AND ALPHA FACTORS

Individual investors employ various trading strategies such as momentum trading strategy, mean-reversion strategy, signals from social media, etc to test their hypothesis in predicting the stock price movement. *Factors* are the data that can provide predictive information about the future movement of stocks [3]. In other words, factors are signals that help us suggest where to invest in a portfolio of stocks and tell the relative magnitude of those investments [4]. Factors that are significantly predictive of the mean of the return distribution are called *alpha factors.*

One example of a former alpha factor is the *market cap of a stock,* i.e., small-cap stocks are more inclined to produce higher returns than large-cap stocks [5]. In this paper, we will be leveraging the sentiment stability of a financial 10-K report over time as an alpha factor. A significant change in the financial document from one period to the next might indicate changes in the market or company structure. It could lead to additional risk for underlying stocks. Thus, it would be beneficial to compare the 10-K documents over time using similarity metrics. In this paper, we predict that the company that endures minimal change in its 10-K reports will eventually generate the highest one-year forward returns. Our hypothesis is keeping a track of how the sentiment of 10-K document is changing from one period to the next.

For example, if company A's 10-K report for the year 2019 is very similar (indicated by high cosine similarity) to its 10-K report published in 2018, then company A is expected to have higher future mean returns as compared to company B that indicates a low cosine similarity between its consecutive 10-K reports. In order to generate this alpha factor, we calculate the cosine similarity between two consecutive 10-K statements in time. The reason for chosing sentiment words from Loughran and McDonald word dictionary is because instead of using a very large vocabulary, we are limiting ourselves to words which are meant to carry a specific sentiment in the particular domain of financial markets.

3.2. ALPHALENS

Alphalens, an open-source tool developed by the Quantopian Research Platform, to analyze alpha factors' predictive power [21]. Alphalens gives us a spectrum of analytics features from returns, information analysis to turnover analysis, which provides us with a sense of predictive power of an individual factor [21]. Using Alphalens, we calculate and analyze the potential returns of our alpha factors *(factor returns)* and also the factors' stability over time *(factor rank*

autocorrelation) [4]. Both these quantities combined give us a sense of how often an individual factor value switches up.

3.3. Factor Return Analysis

Alphalens gives us a full battery of tests from future returns, information analysis to turnover analysis that provides us with a sense of the predictive power of an individual factor [21]. One useful parameter is the return of our factor, which is called the *factor return*. In other words, factor returns are a way to directly measure the returns of our portfolio if their weights were determined purely by the alpha factor [4]. Alphalens requires two mandatory arguments to predict future mean returns: *factors* and *prices*. In this paper, we consider cosine similarity between two consecutive 10-K documents as the factor data and year-end adjusted closing prices of the stocks in our portfolio as pricing data to run against our factor data.

After generating the factor data frame and setting the pricing data, we pass both the arguments in the alphalens' method called '*get_clean_factor_and_forward_returns',* which accepts factor data, pricing data, quantiles, bins, and periods [21]. This function generates a multi-indexed merged data frame that is indexed by date at level 0 and followed by stock/asset at level 1. This data frame contains the values for a single alpha factor, forward returns for each period, and quantile/bin in which the signal belongs. [21]. Finally, we pass the generated dataframe to alphalens' function *'factor_returns*'which generates period-wise factor returns.

3.4. Turnover Analysis

Since liquidity and transaction costs are dependent upon market conditions at the time of the trade, it is challenging to simulate actual transaction costs when evaluating an Alpha factor. So a useful proxy for these real-world constraints is to measure the *turnover* [4]. The turnover analysis estimates the fraction of the portfolio's total value getting traded in a period [3]. One of the ways to measure turnover is *factor rank autocorrelation*. Factor rank autocorrelation is a way to measure how stable are the ranked alpha factors. In this case, stability refers to the fact that alpha ranks do not change much from period to period. Since trading is costly, we would always prefer other factors to be the same, i.e., the alpha factor's ranks do not change significantly per period [21].

A high factor rank autocorrelation is an indication that the turnover is lower. A low or even a negative autocorrelation is a proxy to indicate a higher turnover. If two alpha factors have similar quintile performance and similar factor returns, we would prefer the one with lower turnover [4]. The reason for choosing alpha factor with lower turnover is that it makes it possible for us to execute trades if we have liquid stocks and reduce transaction costs. Excessive turnover could imply that our Alpha factor is only catching noise [3].

3.5. Sharpe Ratio

The *Sharpe Ratio* or risk-adjusted return is a critical metric in evaluating alpha factors. It is the measure of excess portfolio return over the risk-free rate relative to its standard deviation.

$$SharpeRatio = \frac{r_{risky\ portfolio} - r_{riskfree}}{\sigma_{excess\ return}} \quad (4)$$

Sharpe ratio helps us to compare the relative performance of alpha factors. One important thing to note is that the Sharpe ratio is the key and not the magnitude of factor returns [4].

4. Problem Statement

The growth of unstructured corporate data has been exponential. This plethora of information masks itself as a burden on quantitative analysts as an information overload [22]. There are high chances that even the best analysts could overlook the latent information. The financial information available is so comprehensive that the feasibility of promptly analyzing such detailed documents is doubtable. The proposed framework leverages NLP analysis that mitigates the issue of information overload. Automated textual analysis and information retrieval would help analysts process large volumes of financial data in less time and cost. NLP analysis also helps to filter out relevant sections based on our use-case from 10-K documents.

Individual analysts might study the small parts of financial documents in isolation due to which they might overlook latent data and patterns in large datasets [22]. NLP analysis in the proposed framework helps extract such hidden patterns that could be present in unstructured financial reports, which might indicate future returns.

The existing papers that have performed sentiment analysis on financial statements have relied on methods such as word counts. The extant pieces have leveraged textual feed from social media platforms for performing sentiment analysis, which is not a rich source of financial textual data. For a better NLP analysis, the proposed framework uses of corporate reports filed by the companies with the SEC, containing detailed textual information of the company's financial state compared to social media feed. Apart from this, existing papers have categorized financial tweets into a maximum of three categories: positive, negative, and neutral. The proposed framework leverages *Loughran and McDonald's word list* specially curated to recognize financial sentiment and categorizes financial data into seven different views: positive, negative, uncertainty, litigious, constraining, superfluous, and modal.

5. Dataset

The proposed framework leverages the *Loughran and McDonald Financial Sentiment Dictionary* of Loughran and McDonald [6] to perform NLP analysis. The framework uses the following subset of sentiments mentioned in this dictionary:

1. Negative
2. Positive
3. Litigious
4. Constraining

There are two main reasons for choosing this particular word list. The extant pieces have leveraged textual feed from social media platforms for performing sentiment analysis, which is not a rich source of financial textual data. The second reason for choosing this dictionary was its level of comprehensiveness. Loughran and McDonald's word list consists of 2355 negative words and 354 positive words, making the most effective sentiment list for our use-case [18].

S. No.	Company Name	Ticker	CIK Number	Sector
1.	Amazon.com Inc.	AMZN	0001018724	E-commerce
2.	Eli Lilly and Company	LLY	0000059478	Pharmaceuticals
3.	CenterPoint Energy Inc.	CNP	0001130310	Energy
4.	Exxon Mobil Corporation	XOM	0000034088	Energy
5.	Nike Inc.	NKE	0000320187	Apparels/Footwears
6.	Federal Realty Investment Trust	FRT	0000034903	Real estate investments
7.	Boeing Co.	BA	0000012927	Aerospace

Table I: Portfolio details

The proposed framework creates a portfolio of 7 stocks, from diverse sectors, to perform NLP analysis on their respective 10-K statements. With the help of NLP, we can extract information that can be useful for trading. The SEC has created a website called Electronic Data Gathering Analysis Retrieval (EDGAR) that gives us straightforward access to all the available financial statements and is leveraged by the proposed framework to extract 10-K reports of the selected stocks [23].

The framework leverages a python package called *Alphalens,* which requires two mandatory arguments to predict future mean returns: *factor data* and *pricing data* [21]. Factor data are the numerical values, one for each stock per period, potentially predictive of the chosen stocks' future returns. We note the stock price at the time of computing factor values, which we use as the pricing data. Since we are considering financial 10-K statements for NLP analysis that companies file annually, we will consider the *year-end adjusted closing price* of stocks to run against our factor data. To obtain this stock price, we have used the *End of Day US Stock Prices* database by sourcing from *Quotemedia* through *Quandl's* premium subscription [19]. We have chosen an annual adjusted close price because it is this price that reflects the actual value of the company and accounts for all corporate actions such as stock splits, distributions, and dividends [3].

6. Methodology

6.1. Fetching 10-K Financial Reports

When companies file their 10-K reports to the SEC, it is gathered in the EDGAR database and is publicly available for investors to download [6]. To search for company-wise filing reports, we need to submit an HTTPS request to the following REST Url:

```
https://www.sec.gov/cgi-bin/browse-edgar
```

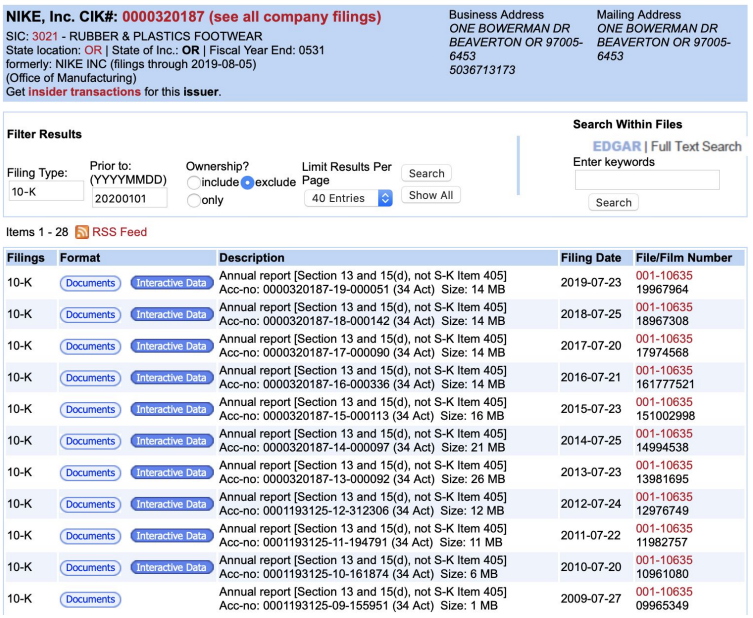
To specify the details of the report in which we are specifically interested in, we need to pass the following query parameters [24]:

1. *CIK number* (CIK): a unique numerical identifier assigned by the EDGAR system.
2. *Report type* (type): type of financial report that we wish to query. Example 10-K, 10-Q, 14-K.
3. *Prior-to date* (dateb): EDGAR accepts a prior-to date that identifies the latest date in which we are interested.
4. *The number of reports* (count): this quantity describes the number of filings up to the prior-to date.
5. *Ownership* (owner): The SEC requires filings from individuals who own significant amounts of the company's stock. Setting the *owner* parameter to *exclude,* EDGAR won't provide reports related to its director or officer ownership [24].

As an example, to download Nike's annual report before 2020, where Nike's CIK number (fetched from Table 1) is 0000320187, and 10-K denotes the type of annual reports, we would form the EDGAR Url as follows :

```
https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany
&CIK=0000320187&type=10-K&dateb=20200101&count=60&owner=exclude
```

Hitting the EDGAR REST Url that we formed above would redirect us to a web page that contains tabular data related to the company's type of filing document, document description, filing date, and file number. Figure 1 shows the EDGAR's search result dashboard after hitting the REST Url formed above.



NIKE, Inc. CIK#: 0000320187 (see all company filings)
SIC: 3021 - RUBBER & PLASTICS FOOTWEAR
State location: OR | State of Inc.: **OR** | Fiscal Year End: 0531
formerly: NIKE INC (filings through 2019-08-05)
(Office of Manufacturing)
Get insider transactions for this **issuer**.

Business Address
ONE BOWERMAN DR
BEAVERTON OR 97005-6453
5036713173

Mailing Address
ONE BOWERMAN DR
BEAVERTON OR 97005-6453

Filter Results

Filing Type: 10-K
Prior to: (YYYYMMDD) 20200101
Ownership? include exclude only
Limit Results Per Page: 40 Entries
Search | Show All

Search Within Files
EDGAR | Full Text Search
Enter keywords
Search

Items 1 - 28 RSS Feed

Filings	Format	Description	Filing Date	File/Film Number
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-19-000051 (34 Act) Size: 14 MB	2019-07-23	001-10635 19967964
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-18-000142 (34 Act) Size: 14 MB	2018-07-25	001-10635 18967308
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-17-000090 (34 Act) Size: 14 MB	2017-07-20	001-10635 17974568
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-16-000336 (34 Act) Size: 14 MB	2016-07-21	001-10635 161777521
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-15-000113 (34 Act) Size: 16 MB	2015-07-23	001-10635 151002998
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-14-000097 (34 Act) Size: 21 MB	2014-07-25	001-10635 14994538
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0000320187-13-000092 (34 Act) Size: 26 MB	2013-07-23	001-10635 13981695
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-12-312306 (34 Act) Size: 12 MB	2012-07-24	001-10635 12976749
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-11-194791 (34 Act) Size: 11 MB	2011-07-22	001-10635 11982757
10-K	Documents, Interactive Data	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-10-161874 (34 Act) Size: 6 MB	2010-07-20	001-10635 10961080
10-K	Documents	Annual report [Section 13 and 15(d), not S-K Item 405] Acc-no: 0001193125-09-155951 (34 Act) Size: 1 MB	2009-07-27	001-10635 09965349

Figure 1: EDGAR search results for Nike's 10-K documents prior-to 2020-01-01.

After fetching the web page as a response, we can perform web scraping with Python by leveraging the *BeautifulSoup* library and access the links that would help us download the 10-K filing reports [25]. These document links will help us download the pure HTML version of the desired 10-K document, which we store in a dictionary against the corresponding stock's CIK number, as depicted in Figure 2.

```
[
  {
    CIK: '0000320187'
    ten_k: '\n<TYPE>10-K\n<SEQUENCE>1\n<FILENAME>nke-5312017x...
    file_date: '2017-07-20'},
  {
    CIK: '0000320187'
    ten_k: '\n<TYPE>10-K\n<SEQUENCE>1\n<FILENAME>nke-5312016x...
    file_date: '2016-07-21'},
  {
    CIK: '0000320187'
    ten_k: '\n<TYPE>10-K\n<SEQUENCE>1\n<FILENAME>nke-5312015x...
    file_date: '2015-07-23'},
  {
    CIK: '0000320187'
    ten_k: '\n<TYPE>10-K\n<SEQUENCE>1\n<FILENAME>nke-5312014x...
    file_date: '2014-07-25'},
  {
    CIK: '0000320187'
    ten_k: '\n<TYPE>10-K\n<SEQUENCE>1\n<FILENAME>nke-5312013x...
    file_date: '2013-07-23'},
]
```

Figure 2: Storing HTML version of 10-K filing reports in a list.

6.2. Preprocessing 10-K Filing Reports

From Figure 2, we can see that the HTML version of 10-K documents stored in the list is very messy. We need to preprocess this data before we can leverage it for performing NLP analysis. Preprocessing of 10-K filing reports is done in the following steps:

1. Remove HTML tags and convert textual data into lower-case.
2. Distill the verbs in 10-K documents by returning the base form of the words using lemmatization.
3. Finally, filter out the stop-words from 10-K reportings.

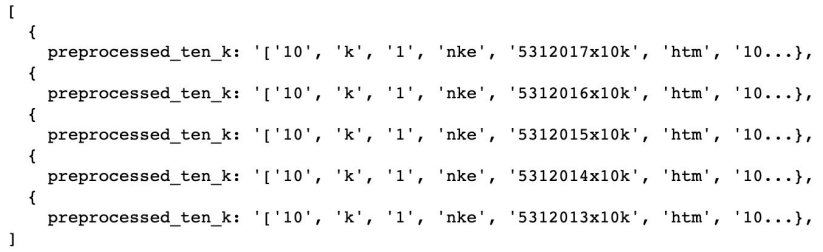


Figure 3: Preprocessed 10-K filings.

6.3. Computing TF-IDF and Cosine Similarity

Once we have our 10-K documents ready in a clean and normalized form, we need to transform them into features for modeling purposes. In section 5.2, we converted the HTML version of 10-K reportings into a preprocessed set of words. But separating these sets of terms is very inefficient as they are of different sizes, contain varied words, and are hard to compare. Instead, we can collect all the words present in both 10-K filings as well as the Loughran McDonald sentiment word list to form the vocabulary [6]. To compensate for the words that frequently occur within a corpus, we can compute the *document frequency* by counting the number of documents in which each word in the column appears. We then divide the term frequencies in the data frame by document frequency of that term [26]. This metric is known as *Term Frequency - Inverse Document Frequency (TF-IDF),* and Equation 1 represents the mathematical version of the TF-IDF metric.

$$W_{i,j} = tf_{i,j} \times log(N/df_i) \tag{1}$$

$tf_{i,j}$ = Number of occurrences of term i in document j

df_i = Number of documents containing i

N = total number of documents

Using the Loughran McDonald sentiment word list, we will select only those words from 10-K documents present in the sentiment word list and then generate sentiment TF-IDF from the 10-K documents. Now, using the calculated TF-IDF, we will calculate the *cosine similarity* between consective financial 10-K reports. Equation 3 calculates the cosine similarity between two vectors.

Figure 4: Cosine similarity between consecutive 10-K statements against time for each stock in our portfolio.

If two documents are to be represented in their vectorized format as follows:

$$\vec{d}_1 = (w_{d10}, w_{d11}, \dots, w_{d1k}) \text{ and } \vec{d}_2 = (w_{d20}, w_{d21}, \dots, w_{d2k}) \quad (2)$$

Here, w_{d1i} and w_{d2i} $(0 \leq i \leq k)$ indicates the frequency of each term inside a document. Now to calculate the cosine resemblance between the two documents, the proposed framework leverages Equation 3.

$$\cos(\mathbf{d_1}, \mathbf{d_2}) = \frac{\mathbf{d_1}\mathbf{d_2}}{\|\mathbf{d_1}\|\|\mathbf{d_2}\|} = \frac{\sum_{i=1}^{n} \mathbf{d_{1}}_i \mathbf{d_{2}}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{d_{1}}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{d_{2}}_i)^2}} \tag{3}$$

To analyze the pattern of cosine similarity between each tick in time for the selected companies in our portfolio, Figure 4 plots the cosine similarity between consecutive 10-K documents for each company against time. The cosine similarity plots indicate the stability of individual sentiments conveyed by our portfolio with respect to time. Here, we have to analyze, which sentiment is maintaining the highest cosine similarity for all the stocks in our portfolio. It is this sentiment, which has the highest stability and therefore, according to our hypothesis, will yield the maximum returns.

7. Results and Evaluations

7.1. Factor Returns

Figure 5 shows a plot between 1 year factor returns and time. Here, the alphalens library generates one-year forward returns as our input data was yearly adjusted close price of the stocks in our portfolio. As we can see from the graph, 10-K financial reports expressing a *positive sentiment,* yield the maximum returns. On the other hand, the forms that convey *constraining*, *negative*, and *litigious* resulted in the lowest returns. The following observation proves the existence of a strong correlation between the stability of the sentiment indicated by our portfolio and its future mean returns.

Figure 5: Factor returns of our portfolio over time

7.2 Turnover Analysis

A high factor rank autocorrelation is an indication that the turnover is lower. A low or even a negative autocorrelation is a proxy to indicate a higher turnover. If two alpha factors have similar quintile performance and similar factor returns, we would prefer the one with lower turnover [4]. The reason for choosing alpha factor with lower turnover is that it makes it

possible for us to execute trades if we have liquid stocks and reduce transaction costs. Excessive turnover could imply that our Alpha factor is only catching noise [3].

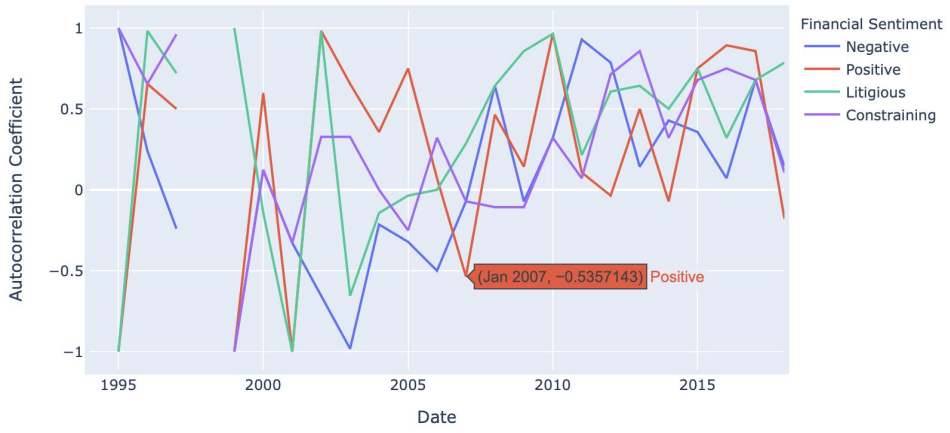


Figure 6: Factor rank autocorrelation against time

7.3 Sharpe Ratio

Table 2 shows the Sharpe ratio of our alpha factors. Usually, a ratio under 1.0 is considered sub-optimal. Sharpe ratio greater than 1.0 is *acceptable to good* by investors. A Sharpe ratio higher than 2.0 is good, and investors deem a 3.0 or higher Sharpe ratio excellent [3]. Looking at the Sharpe ratio of our alpha factors, we can see that the 10-K filing reports that convey the the most stable *positive* sentiment achieves the highest Sharpe ratio of 1.02.

Factors	Sharpe Ratio
Positive	1.02000000
Litigious	0.89000000
Negative	-0.24000000
Constraining	-2.98000000

Table 2: Factors and their corresponding Sharpe ratio

8. Conclusion and Future Work

This paper has proposed a framework that leverages the Loughran McDonald's word list to perform NLP analysis on financial 10-K reportings of 7 companies belonging to diverse sectors ranging from e-commerce, pharmaceuticals, energy, footwears, and aerospace. The framework extracts and downloads HTML versions of 10-K filings from the SEC's EDGAR database and preprocesses them for further NLP analysis. After cleaning and normalizing the 10-K documents, the proposed framework converts the filings into their corresponding numeric vector representations using TF-IDF that indicate the frequency of each Loughran McDonald

sentiment word present in that 10-K document. The framework then calculates cosine similaity between consecutive 10-K reports using TF-IDF values. This cosine similarity is used as the alpha factor to calculate 1-year future mean returns using the alphalens library. Upon evaluation, we found that 10-K documents indicating *positive* sentiment, that had maintained the highest cosine similarity against time, had the highest factor return. In conclusion, our extensive analysis shows that the sentiment stability of financial 10-K reports over time could be a potential alpha factor for predicting future mean returns.

We can leverage even more categories of sentiment for future work and analyze their correlation with future mean returns. We can also create a separate portfolio to include data from other companies that can further validate our hypothesis. We can also test our hypothesis for a non-diversified portfolio. In future, this work could further be validated by performing back-testing.

REFERENCES

[1] Ranjan, N., Mundada, K., Phaltane, K., and Ahmad, S., 2016. A Survey on Techniques in NLP. *International Journal of Computer Applications*, *134*(8), pp.6-9.

[2] Musto, C., Semeraro, G., Lops, P., De Gemmis, M., and Lekkas, G., 2015. Personalized finance advisory through case-based recommender systems and diversification strategies. *Decision Support Systems*, *77*, pp.100-111.

[3] Investopedia documentations, https://www.investopedia.com

[4] AI for Trading, Nanodegree Course by Udacity

[5] Rompotis, G., 2019. Large-cap vs. small-cap portfolio performance: new empirical evidence from ETFs. *Review of Accounting and Finance*.

[6] 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), pp.35-65.

[7] Ramos, J., 2003, December. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, pp. 133-142).

[8] Viswanathan, K., 2010. Text-based similarity metrics and delta for semantic web graphs.

[9] Azimi, M., and Agrawal, A., 2019. Is positive sentiment in corporate annual reports informative? Evidence from deep learning. *Evidence from Deep Learning (July 2019)*.

[10] Kearney, C., and Liu, S., 2014. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, *33*, pp.171-185.

[11] Li, F., Lundholm, R., and Minnis, M., 2013. A measure of competition based on 10-K filings. *Journal of Accounting Research*, *51*(2), pp.399-436.

[12] Jegadeesh, N., and Wu, D., 2013. Word power: A new approach to content analysis. *Journal of financial economics*, *110*(3), pp.712-729.

[13] Tetlock, P.C., Saar-Tsechansky, M., and Macskassy, S., 2008. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, *63*(3), pp.1437-1467.

[14] Azimi, M., and Agrawal, A., 2019. Is positive sentiment in corporate annual reports informative? Evidence from deep learning. *Evidence from Deep Learning (July 2019)*.

[15] Smailović, J., Grčar, M., Lavrač, N., and Žnidaršič, M., 2013, July. Predictive sentiment analysis of tweets: A stock market application. In *International Workshop on Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data* (pp. 77-88). Springer, Berlin, Heidelberg.

[16] Rao, T., and Srivastava, S., 2012. Analyzing stock market movements using Twitter sentiment analysis.

[17] Pagolu, V.S., Reddy, K.N., Panda, G., and Majhi, B., 2016, October. Sentiment analysis of Twitter data for predicting stock market movements. In *2016 international conference on signal processing, communication, power, and embedded system (SCOPES)* (pp. 1345-1350). IEEE.
[18] Kandrup, J., Johansen, T.R., Mølhave, J.C. and Plenborg, T., 2018. Sentiment Analysis of 10-K Filings.
[19] Quandl end of day US Stock Prices database, Accessed: 2020-10 https://www.quandl.com/data/EOD-End-of-Day-US-Stock-Prices/documentation/product-overview.
[20] 10-K document by Investopedia. https://www.investopedia.com/terms/1/10-k.asp
[21] Alphalens official documentation, https://quantopian.github.io/alphalens/
[22] Lewis, C, and Young, S., 2019. Fad or future? Automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, *49*(5), pp.587-615.
[23] The U.S. Securities and Exchange Commission website, https://www.sec.gov/edgar/searchedgar/companysearch.html
[24] Accessing EDGAR data via SEC API, https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm
[25] Beautiful Soup: Build a Web Scraper With Python, https://realpython.com/beautiful-soup-web-scraper-python/
[26] TF-IDF Python Example, Medium article by Cory Maklin, https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76

AUTHOR

Purva Singh has completed her B.Tech in Computer Science and Engineering from Vellore Institute of Technology, Vellore. She has been exploring the field of Artificial Intelligence and Deep Learning for the past two years. She is passionate about Natural Language Processing, Deep Reinforcement Learning, and Big Data Analytics.

Email: singhpurva2906@gmail.com