Textual Analysis of SEC Filings

Chris Bemis
Jesse Berwald
Ali Nabizadehgangaraj
Loren Anderson

April 23, 2018

Abstract

The Securities and Exchange Commission (SEC) requires public corporations to file detailed disclosures. The stated purpose of these disclosures is "to increase the efficiency and fairness of the securities market by accelerating the receipt, acceptance, dissemination and analysis of time-sensitive corporate information". Disclosures are filed on a quarterly and annually basis via 10-Q and 10-K reports, respectively, and are available from the SEC's Electronic Data Gathering, Analysis, and Retrieval system (EDGAR). Each contain a wealth of detailed information regarding, for instance, tax liabilities, real estate investments, and supply chain strategy. Investment firms routinely leverage the wealth of information found in SEC disclosures to make intelligent, long-term investment decisions based on corporate disclosures. This project aims to automate the process of disclosure analysis. Here, we attempt to predict a company's performance through a combination of natural language understanding and machine learning. Also, we try to determine if SEC disclosures contain adequate information – predictive positive or negative sentiments – that can facilitate the construction of machine learning models to predict corporate returns.

1 Text Informativeness

The first method we consider stems from attempts to gauge the grade level and informativeness in text documents. Two common measures of grade level or ease of reading are the Fog Index and Flesch Reading score. Each considers the average sentence length and proportion of complex words determined by the number of syllables to compute a score. However, these dont necessarily translate well to investors understanding of the documents.

In October 1998, the SEC created the plain English rule to emphasize that clear writing benefits a firms least sophisticated investors. The idea is that investors, brokers, advisers, and others in the financial services industry will be more able to assess and more likely to invest in companies whose financial disclosures are not buried in legal jargon and obtuse language. Research performed by Tim Loughran and Bill McDonald shows that these scores are poor at gauging text informativeness. Specifically, syllables are found to be uncorrelated with text informativeness. Therefore, Loughran and McDonald created the plain English measure that better captures text informativeness. It contains a mixture of components such as sentence length, average word length, etc. Furthermore, they showed with this measure that more informative documents increased the trading of average investors and increased the magnitude of returns in a 0-3 day window after the filing. (note that the magnitude is not signed)

We will attempt to use these measures to predict the return of document after 1, 3, 6, and 12 months after the filing date. Research states that there are many possible explanations of why documents may be more complex or informative, but we initially hypothesize that informative documents have higher returns.

1.1 Methodology

- Get matrix containing word frequency counts of each document.
- Compute 3 scores for each document.
- Compare ranked lists to ranked return list, fixing the scoring algorithm.

1.2 Variants

1.3 Results

2 Word Tones

Another method we consider deals with the tones of words in the document. Specifically, we consider words with negative tones. The Harvard Psychosociological dictionary contains word classifications, some being positive and negative. Research by Tim Loughran and Bill McDonald shows that about of the negatively classified words in that dictionary are not negative in the financial context. For example, taxes and liabilities are not usually considered negative in the financial context. Therefore, Loughran and McDonald created their own list of 2,337 words that are supposed to have negative tones in the financial context. Examples include investigation, misstatement, and misconduct. Each one of the above methods computes a score for each document based on the proportion of negative words it classifies. The results using the financial-negative list showed that there was a significant relationship between that score and the return 0-3 days after the 10-K filing. As one would expect, the more negative a document seemed, the worse its return was. The relationship wasnt significant with the Harvard list. However, when words that occurred more frequently had their weights lowered and words that appeared in fewer documents had their weights increased, the results were comparable and the relationship was significant for both lists (Harvard and Financial Negative). This process is known as Term-Frequency Inverse Document Frequency (TFIDF).

We will attempt to build on this research by using both of these lists to attempt to predict returns for 1, 3, 6, and 12 months after the filing date of the 10-K reports.

2.1 Methodology

- Get matrix containing word frequency counts of each document.
- Compute 2 scores for each document.
- Compare ranked lists to ranked return list, fixing the scoring algorithm.

2.2 Variants

2.3 Results

3 File Size

The last method we consider comes again from the research of Tim Loughran and Bill McDonald. In one of their more recent papers published in 2014, they define readability as the effective communication of valuation relevant information. The assumption in this paper is that better written documents produce less ambiguity in the valuation, which results in less volatility immediately after the filing date.

Loughran and McDonald use file size of the complete submission text file of the 10-K report on the SECs EDGAR website. They state that effective communication of valuation relevant information to investors should focus less on style and instead encourage managers to write more concisely. Their motivation is that concisely written documents are more likely to be read, and the information from the 10-K more effectively incorporated into stock prices and analyst forecasts.

The results showed that greater file sizes (less concise documents) had significantly more volatility. In their research, file size was shown to be comparable to other methods that required parsing the text documents. Therefore, since there is no need to parse for file size and it is an easily reproducible statistic, they recommend scientists to use this in their research.

We plan to predict returns based on these file sizes. Our initial hypothesis is that larger file sizes lead to relatively worse returns.

3.1 Methodology

- Get file sizes.
- Compute score for each document.
- Compare ranked lists to ranked return list, fixing the scoring algorithm.

3.2 Variants

We may choose to remove pictures, tables, etc. from each document.

3.3 Results

4 Machine Learning Techniques

In contrast to our previous 3 methods where we compute a score using a pre-defined model using financial context, we now look to machine learning methods where we train a model using past data and test on future data. Our baseline machine learning method will use the documents as data points with their words as features and incorporate principal component analysis and multiple linear regression.

The documents contain many words that are simply noise and dont convey meaning to our problem. PCA allows us to (attempt to) remove noise from the dataset. First we find a small set of factors that are based on words in the documents. Each document can be represented as a combination of some of these factors, so we have taken data points (documents) with many words and reduced them down to combinations of a small number of factors. These factors are meant to better separate or distinguish the documents.

This allows us to more easily perform a multiple linear regression on these reduced data points. Multiple linear regression attempts to find a relationship between the an output variable and multiple input variables. The output variable in this case is the return, and the input variables are the aforementioned factors. We create the model using past data and test its accuracy on future data. Then, given a new data point, we reduce the dimensionality through PCA and determine its output in the multiple linear regression.

4.1 Methodology

- Get matrix containing word frequency counts of each document.
- Calculate the return for the company of each document for n months after 10-K release.

- Perform PCA on the matrix.
- Run linear regression with the matrix columns as explanatory variables and return values as the response variable.
- Train the algorithm on a set of documents and test on a set of documents made after the train set documents.

4.2 Variants

Word counts are a nice baseline, but there exist other well-known and popular methods that extend this idea. One possibility is using TFIDF and its variants. Another technique is Latent Dirichlet Allocation (LDA). The purpose of LDA is to learn a set of topics that are distributions on words. Then each document is a distribution of topics. After finding a set of topics through training, a new document's topics can be found by computing its (log) probability that it is a topic (over each topic). This reduces the dimensionality from a bag of words down to a few topics (10 maybe?). This is easily accomplished through the Gensim library in Python. The "story" of LDA is that one creates a document by choosing a topic with a certain probability and then choosing a word from the topic with certain probability. This is a crude approximation, yet it is more relevant to documents than a simple PCA.

Perhaps the bag-of-words model is too crude to produce meaningful results, as it removes order of the words. Consequently, we could consider a n-grams: sequences of n words in a row. These n-grams replace our words in the bag-of-words model. There are known approaches to tackle these problems in machine learning; however, we (Loren) have not read the relevant papers yet. This increases the feature space, so computational care must be taken.

We don't need to include all words in each document. There are many possible stop word lists online, and we could experiement using different stop word lists and removing each word in the list from all documents before creating the matrix.

With all of these methods, we will incorporate other classification algorithms. However, we want to determine the success of our baseline methods first before exploring in that direction.

4.3 Results

5 Conclusions