USING NLP AND DEEP LEARNING ON SEC FILINGS TO PREDICT STOCK PRICE MOVEMENTS

Yusuf Aktan
June 2018

Abstract

Company SEC filings have long been used as valuable sources of information in making investment decisions. Several papers and projects have demonstrated how natural language processing techniques could be used to extract information from SEC filings and news in order to predict stock volatility. This project recreates and builds upon past work by using GloVE embeddings and MLP, CNN, and RNN deep learning architectures to predict changes in stock prices immediately following the release of an 8-K document.

Introduction

In the financial services and banking industry, vast amounts of resources are dedicated to pouring over, analyzing, and attempting to quantify qualitative data from news and SEC mandated reporting. This problem is also constantly compounded as the news cycle shortens and reporting requirements for public companies become more onerous. Several studies have also suggested that the highest quality signals of stock price swings come not from third party news reporting, but the companies themselves and their reports to the SEC. [1][2] In addition, several papers have demonstrated the utility of neural networks in NLP[3][4], and NLP in extracting information from SEC reports to predict changes in stock prices.

In this project I attempt to demonstrate the viability of using natural language processing word embeddings on SEC 8-K documents with deep learning methods to predict stock price volatility after a company experiences a major event. I set up this project up based off a paper by Google and Stanford University ("On the Importance of Text Analysis for Stock Price Prediction"),[5] and build upon it by exploring pre trained word embeddings and deep learning neural network architectures.

[1] Content of Annual Reports as a Predictor for Long Term Stock Price ...
[2] https://careeradvancement.uchicago.edu/sites/default/files/ucib-journal/filings-volume-and-volatility.pdf
[3] https://arxiv.org/pdf/1703.03091.pdf
[4] https://arxiv.org/abs/1702.01923
[5] https://nlp.stanford.edu/pubs/lrec2014-stock.pdf

Methods

Data Collection

8-K documents were scraped from the SEC Edgar database for all companies in the S&P 500 as of May 2018, using the BeautifulSoup python package. Metadata such as the date and time a document was published and the categories of disclosures made[6], was extracted, while tables and charts were discarded. Because of the size of the data and time needed to scrape and collect it, a Google Cloud instance with 8 Intel Xeon cores and 52 GB of memory along with a Google Cloud storage bucket were eventually setup. Historical open and adjusted close price data for the same companies was gathered from the AlphaVantage API. Historical index prices for the VIX and GSPC (S&P 500) was downloaded from Yahoo Finance.

Feature Engineering

For each document release, one year, one quarter, and one month historical moving average price movements were calculated based on the time right before a document's release, and normalized by the change in the S&P 500 index. All windows refer to days that the NYSE and Nasdaq were actually open (non-holiday weekdays).

Feature	**Moving Average Window**
One Month Movement	5 days
One Quarter Movement	10 days
One Year Movement	20 days

Table 1. Moving average windows for historical price movement calculations

The target feature was calculated as the change in price of a stock right before and after a document's release, normalized by the change in the S&P 500. For example, for a company that released a document on February 5, 2018, the change in its opening and adjusted close

[6] https://www.sec.gov/fast-answers/answersform8khtm.html

price was calculated, minus the change during the same time for the S&P 500 index. The normalized changes were labeled as either “up” (> 1%), “down” (< -1%), or “stay“(between -1 and 1%).

Text Preprocessing

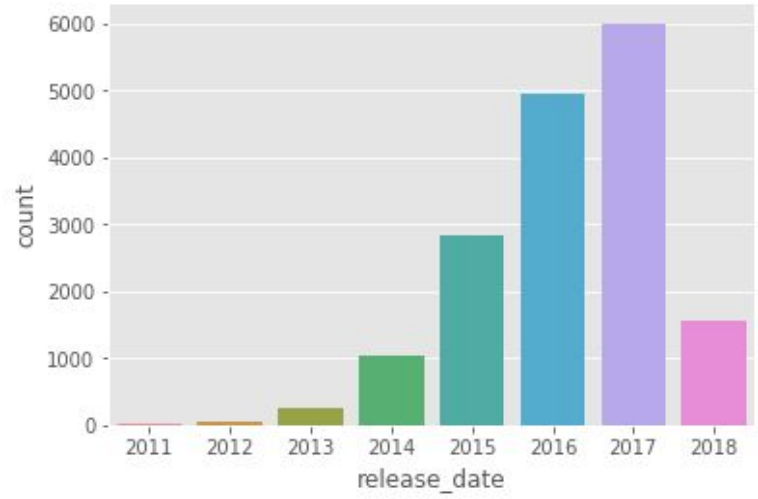
All texts were preprocessed by removing stopwords, punctuation, and numbers, lemmatization, and conversion to lower case. This was accomplished using the NLTK WordNet corpus reader in combination with Dask for a multithreading speedboost
All documents were padded with zeros to a uniform length of 34603 words. This cutoff was chosen at the 90th percentile of document lengths in order to preserve most textual information but prevent the dataset from becoming unnecessarily large. The Stanford NLP Wikipedia 2014 + Gigaword 5 100 dimension was chosen for pre trained word embeddings under the assumption that it would carry information for specialized,industry-specific words found in the texts since it was trained from the Wikipedia corpus.

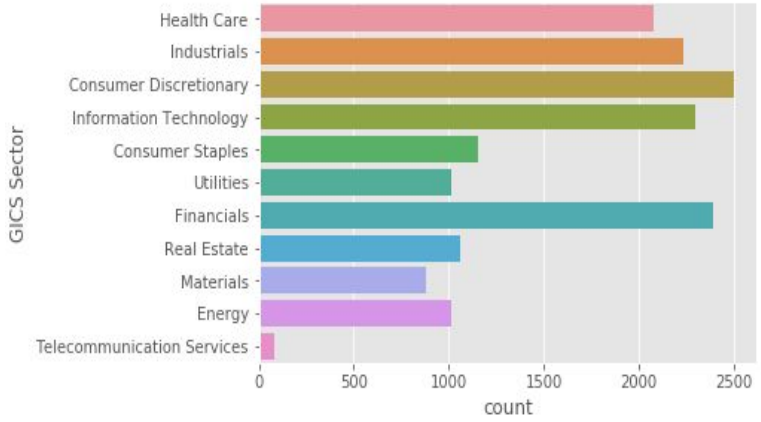
Explanatory Feature (X)	**Description**
Global Industry Classification Standard (GICS) Sector for Company	i.e technology, healthcare, industrials
VIX (Volatility Index)	Measure of market volatility expressed as standard deviation of returns over 30 days for S&P 500
Items	Item sections extracted from 8-K texts which classify category of disclosure or event, example items are bankruptcy, acquisition/disposition of assets, entry into a new material agreement, etc. One 8-K document could contain multiple items and disclosures
Recent Stock Movements (Year, Quarter,	Historical one year, one quarter, and month

Month)	moving average stock movement, normalized by movement of the S&P 500 over the same period of time
Texts	
Report Text (Corpus)	Corpus of 8-K documents with tables, numbers, stopwords, and punctuation removed, stemmed, lemmatized, and then vectorized using GloVe 100 dimension pre-trained word embeddings
Target Variable (Y)	
Trade Signal	Change in stock price for a company directly immediately before and after a document's release, normalized by the change in the S&P 500. Signals are "up","down", and "stay"

Table 2. List of all dataset features

After dropping duplicate samples, and texts for which no release date could be extracted, the final dataset consisted of about 17,000 documents for 500 companies from 2011 to 2018.





Figures 1 & 2. Dataset sample sizes for year an 8K filing was released, and operating sector company

Machine Learning

All categorical features were one hot encoded, and continuous features such as recent stock movements, and closing prices of VIX were standardized to have a mean of 0 and standard deviation of 1.

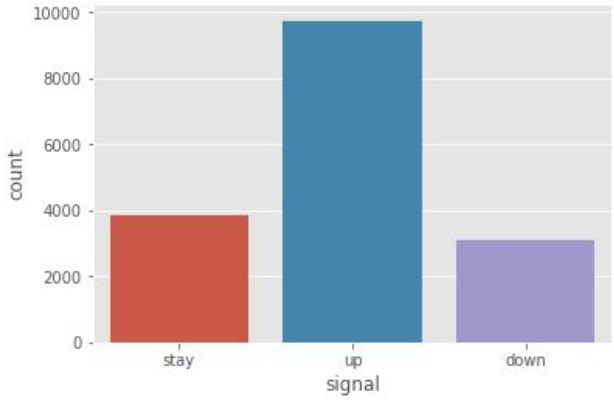


Figure 3. Count of target classes. Dataset has imbalanced classes

The dataset was then randomly shuffled and split into 80% train and 20% test data. The dataset suffered from imbalanced classes, with over 50% of samples labeled as (“up”), which intuitively makes sense considering the steady rise of the S&P 500 over the last decade. Oversampling on the training data was used to correct for this, with randomly selected samples in each class duplicated to have an equal number of samples of each of the three classes.

Four different machine learning architectures were constructed using Keras with a Tensorflow backend, consisting of two input layers (one for text documents, one for features), an embedding layer with the pretrained GloVe vectors, and (a) a multilayer perceptron fully connected network - “MLP”, (b) two 1D convolutional layers - “CNN”, (C) a bidirectional GRU layer - “RNN”, and (d) a 1D convolutional layer, followed by a GRU layer - “CNN-RNN”. Each network was trained for 10 epochs, with a batch size of 32, on two NVIDIA K80 GPUs.

	Loss	Accuracy	AUC_ROC
MLP	1.078725	0.461999	0.557087
CNN	1.073401	0.403950	0.575170
RNN	0.913587	0.629563	0.797259
CNN RNN	0.934135	0.615001	0.782580

Table 3. Metrics on validation dataset after 10 epochs of training.

The RNN and CNN-RNN networks achieved the highest accuracy and auc_roc scores on the validation dataset. However the CNN-RNN model required half of the training time of the RNN model. Training continued for the CNN-RNN model for another 15 epochs, with minimal gain in validation loss.

Results

The CNN-RNN network was able to achieve an accuracy of 64.5% and AUC_ROC of .90 on the validation data.

	Loss	Accuracy	AUC ROC
CNN_RNN	1.1853	0.6447	0.9026

Table 4. Highest performance of the CNN RNN model after 23 epochs of training

This model is a 94% improvement over a baseline random choice algorithm, as well as a 16% improvement over the Stanford and Google paper this project was modeled from. These results suggests that although word-embeddings and neural networks require more time and computational resources to build and train, the gains in accuracy are worth the effort.

Discussion

This project only touches the surface of how the latest natural language processing techniques and deep learning models could be used to extract meaningful information from SEC reporting and asses swings in a company's stock price. A more specialized set of word embeddings, or advanced techniques such as Sense2Vec[7] could be explored in order to gather more nuanced information from text.

Change in a stock market price was only measured immediately before and after a document's release, although it's quite possible markets react to different types of news at different rates. An extension of this project could be looking at price movements in the days following a disclosure.

[7] https://arxiv.org/abs/1511.06388

Finally, although CNN, RNN, and hybrid architectures are currently considered to be some of the most state-of-the-art NLP models, there are still a virtually unlimited number of combinations to explore within those models, in regards to RNN vs LSTM vs GRU units, and the depth, size, and hyperparameters regarding RNN and CNN layers, and the use of pooling layers. Unfortunately time and money restrictions (Using multiple GPUs and processing large datasets can quickly add up in Google Cloud) did not allow me to explore these possibilities. Still a 64% accuracy suggests that these efforts could be worth the while to quickly extract data from large quantities of textual data and make trading decisions, tasks that would normally require armies of specially-trained analysts.