SMU Data Science Review

Volume 2 | Number 1 Article 17

2019

Use Advances in Data Science and Computing Power to Invest in Stock Market

Mustafa A. Sakarwala
Southern Methodist University, msakarwala@mail.smu.edu

Anthony Tanaydin
Southern Methodist Unversity, atanaydin@mail.smu.edu

Use Advances in Data Science and Computing Power to Invest in Stock Market

Mustafa Sakarwala and Prof. Anthony Tanaydin PhD

Masters of Science in Data Science,
Southern Methodist University,
Dallas, TX 75275 USA
{msakarwala,atanaydin}@smu.com

Abstract. As part of its overseeing of capital markets, the Securities and Exchange Commission (SEC) requires firms with publicly traded shares to issue periodic reports to shareholders. These SEC filings are part of the SEC's Electronic Data Gathering, Analysis, and Retrieval system (EDGAR), a large online database. Financial services and banking industry have armies of analysts that are dedicated to rushing over, analyzing, and attempting to quantify qualitative data from this SEC mandated reporting. We sought to prototype a predictive model to render consistent judgments on a company's prospects, based on the written textual sections of public earnings releases extracted from SEC 8-K financial reports and actual stock market performance. In this project, we leverage data from EDGAR to model the viability to predict the stock price through, natural language processing (NLP) and deep learning methods. The model used to predict the stock movement in the near futures (next few days from the release of report) by incorporating relevant financial information, such as recent stock price movement and above or below earnings, and other textual information from these financial reports. Our results will demonstrate how a deep learning model trained on text in SEC Document filings could provide a valuable signal to an investment decision maker. The results will be most important in 1-5 days (i.e. The next day after the financial event) but persist(constant or in some trend up or down) for up to five days.

1 Introduction

In the financial field, stock markets and its trends could be extremely volatile in nature. It attracts researchers to capture this volatility to predict its next behavior. Investors and market analysts study the market behavior and plan their buy or sell strategies accordingly. Mainly there are two methods for forecasting market trends. One is Technical analysis and other is Fundamental analysis [1]. Technical analysis considers past price and volume to predict the future trend, whereas on the other hand, Fundamental analysis of a business involves analyzing its financial data to get insight. The ability of both technical and fundamental analysis is disputed by the efficient-market hypothesis which states that stock market prices are essentially unpredictable[2].

Articles in the Wall Street Journal and on Bloomberg articles, corporate SEC filings, earnings-call transcripts, social media messages, etc. all contain ample information about financial markets and investor behaviors. Extracting meaningful signals from unstructured and high dimensional text data through traditional methods is not an easy task and the predictions are not dependable. However, with advancements in machine learning and computational linguistic techniques in recent years, processing and statistically analyzing textual content tasks can be accomplished. Many applications of statistical text analysis in social sciences have proven to be successful [3].

Publicly traded companies are required to file form 8-K, 10-K and other reports to the Securities and Exchange Commission (SEC) to notify its investors of any material events. Companies are required to completely disclose pertinent information about their business and corporate structure to investors. Examples of such material events include changes in management, the departure of directors, bankruptcy, and layoffs. They have to provide details like, how much are they paying the CEO, how much did they pay for the company that they acquired last year, how much is the company spending on their lease, how much cash do they have on hand? How are their European operations progressing? It's all in there.

SEC filings deliver the pure information about a company, unblemished by brokerage analysis. One can find out everything that they ever wanted to know about a company, just by skimming through the pages of their quarterly or annual reports. In the presence of a material event, it is likely that the market will react or has reacted to the event. Serious investor and financial firms, read SEC filings religiously. Before we go on, let's explain what the 10-K and 8-K are.

- 8-K : The form that is filed by companies to inform their shareholders of "unscheduled material events that are important to shareholders" like CFO leaves, SEC is launching an investigation, the company announces a new business deal, delisting notice, shutting down a plant, layoffs or bankruptcy. These are all material events that would require an 8-K to be filed. The 8-K is extremely common, and many companies will file a number of 8-K's throughout the course of a quarter.[4]

- 10-K : The annual report that is filed (yearly) by a company. This is an extremely in-depth document that contains everything that you ever wanted to know about the company. Executive compensation, audited financial statements, organization structure, etc.[5]

A thorough analysis of the investment opportunity of a company would also include a review of other companies in the industry to understand relative performance. Several papers have demonstrated the utility of machine learning, and NLP in extracting information from SEC reports to predict changes in stock prices. Heeyoung Lee et al [6] present a text analysis based approach on building predictive models with company 8-K reports. Kogan et al [7] demonstrate predicting financial volatility from a financial report. Our solution is to use these techniques to build a predictive model that could do a preliminary review of these documents more consistently and economically, allowing investment analysts to focus their follow-up analysis time more efficiently and resulting in better investment decisions.

This study will follow the Fundamental analysis technique combined with some technical data to discover the trend of a stock by considering report published by a company and tries to classify the movement of stock prices in the near future. This project is an attempt to build a model that predicts financial report polarity which may affect changes in stock trends. In other words, check the impact of financial reports on stock prices.

The exploration comprises of three parts. The first part is to read financial reports hosted by the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system operated by the US Securities and Exchange Commission (SEC) and linking them to related stock price index data. The second part is text analysis of the 8-K financial report using Natural Language Processing and the third part is using Neural Network (CNN-RNN and other deep learning architectures) to create a model for predicting stock price movement.

2 Primer on SEC, EDGAR

The Securities and Exchange Commission (SEC) was created in response to the Great Depression in 1934. As part of its overseeing of capital markets, the Securities and Exchange Commission (SEC) has required corporations to file a host of forms and documents to comply with the regulatory policy. Recognizing the anticipated growth in SEC's responsibilities, the SEC created the EDGAR system to collect and make available electronic documents that provide investors with pertinent company information in 1984.

EDGAR - Electronic Data Gathering Analysis and Retrieval system – is the electronic filing system created by the Securities and Exchange Commission (SEC) to increase the efficiency and accessibility of corporate filings. The system is used by all publicly traded companies when submitting required documents to the SEC. Corporate documents are time sensitive, and the creation of EDGAR has greatly decreased the time it takes for corporate documents to become publicly available. Since 2006, the EDGAR system has also captured information about mutual funds and variable insurance products. The system now contains more than 20 million corporate filings in its database [8].

There are four types of EDGAR indexes that list company filings for all four quarters of each year. [9]

1. Ownership: The SEC requires filings from a company's director, the company's officers, and individuals who own significant amounts of the company's stock.
2. Form Types: EDGAR provides hundreds of different types of financial reports, and you can download the PDF that describes all of them.

Form	ID	Description
Prospectus	424	Provides general information about the company
Annual report	10-K	Annual statement of a company's finances
Quarterly report	10-Q	Quarterly statement of a company's finances
Annual proxy statement	14-K	Information about company owners
Current events report	8-K	Notification of important event
Sale of securities	144	Notification of significant sale of stock
Beneficial ownership report	13-D	Identifies prominent owner (>5%) of company

Table 1. Types of Financial Reports

3. Central Index Keys (CIKs): This ten-digit identifier is called the central index key or CIK. A CIK is used to look up a company's report.
4. Report Dates and Count

To access reports, we have to submit an HTTPS request to a URL that starts with https://www.sec.gov/cgi-bin/browse-edgar?. We refer to this as the EDGAR URL.

To specify a report, we are interested in, we need to append a query string to the URL that provides the following information:

- The corporation's identifier (CIK)
- The type of the report (type)
- The prior-to date (dateb)
- The after-from date (dates)
- The number of reports (count)
- Ownership (owner)

3 Primer on Natural Language Processing

Natural Language Processing (NLP) deals with building computational algorithms to automatically analyze and represent human language. NLP based systems have enabled a wide range of applications such as Google's powerful search engine, and more recently, Amazon's voice assistant named Alexa. NLP is also useful to teach machines the ability to perform complex natural tasks such as machine translation and dialogue generation.

For a long time, the majority of methods used to study NLP problems employed shallow machine learning models and time-consuming, hand-crafted features. This led to problems such as the curse of dimensionality since the information was represented with sparse representations (high-dimensional features) [10]. However, with the recent popularity and success of word embeddings (low dimensional, distributed representations), neural-based models have achieved superior results on various language-related tasks as compared to traditional machine learning models like SVM or logistic regression [11].

A word is known by the company it keeps (FIRTH, J.R. 1957:11).Distributional vectors, also called Word Embeddings, are based on the so-called distributional hypothesis – words appearing within similar context possess similar meaning. Words embeddings are pre-trained on a task where the objective is to predict a word based on its context, typically using a shallow neural network. The figure below illustrates a neural language model proposed by Bengio and colleagues. [12] (Colleagues, n.d.)

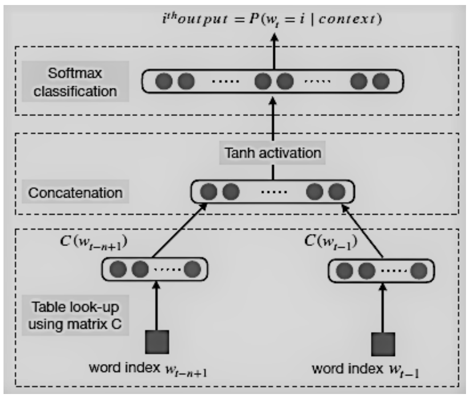

Fig. 1. Shallow Neural Network

Distributed representations gained popularity when the continuous bag of words (CBOW) and skip-gram model were introduced to the field. They were popular because they could efficiently construct high-quality word embeddings and could also be used for semantic compositionality.

Around 2013, Mikolav et al., proposed both the CBOW and skip-gram models [13]. CBOW is a neural approach to construct word embeddings and the objective is to compute the conditional probability of a target word given the context words in a given window size. On the other hand, Skip-gram is a neural approach to construct word embeddings, where the goal is to predict the surrounding context words (i.e., conditional probability) given a central target word. For both models, the word embedding dimension is determined by computing (in an unsupervised manner) the accuracy of the prediction.

One of the challenges with word embedding methods is when we want to obtain vector representations for phrases such as “hot potato” or “Boston Globe”. We can’t just simply combine the individual word vector representations since these phrases don’t represent the combination of the meaning of the individual words. And it gets even more complicated when longer phrases and sentences are considered. The other limitation with the word2vec models is that the use of smaller window sizes produce similar embeddings for contrasting words such as “good” and “bad”, which is not desirable especially for tasks where this differentiation is important such as sentiment analysis. These models also suffer models also suffer from other problems such as not taking into account polysemy and other biases that may surface from the training data.

For tasks such as parts-of-speech (POS) tagging or named-entity-recognition (NER), it is useful to look at morphological information in words such as characters and combinations thereof. Since we are analyzing the text at the character level, these type of embeddings are known as 'Character Embeddings'. They help to deal with the unknown word issue as we are no longer representing sequences with large word vocabularies that need to be reduced for efficient computation purposes.

Though both character-level and word-level embeddings have been successfully applied to various NLP tasks, their long-term impact has been questioned. It has recently been found that word vectors are limited in how well they capture the different facets of conceptual meaning behind words.

3.1 Global Vectors (GloVe)

The GloVe model stands for Global Vectors which is an unsupervised learning model which can be used to obtain dense word vectors similar to Word2Vec [14]. However the technique is different and training is performed on an aggregated global word-word co-occurrence matrix, giving us a vector space with meaningful sub-structures. This method was invented in Stanford by Pennington et al.

The basic methodology of the GloVe model is to first create a huge word-context co-occurrence matrix consisting of (word, context) pairs such that each element in this matrix represents how often a word occurs with the context (which can be a sequence of words). The idea then is to apply matrix factorization to approximate this matrix as depicted in the following figure 2.

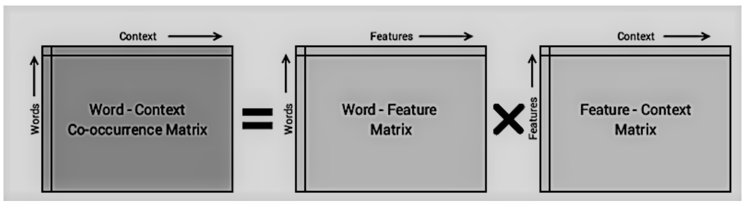

Fig. 2. Conceptual model for the GloVe model's implementation

Considering the `Word-Context (WC)` matrix, `Word-Feature (WF)` matrix and `Feature-Context (FC)` matrix, we try to factorize `WC = WF x FC`, such that we we aim to reconstruct `WC` from `WF` and `FC` by multiplying them. For this, we typically initialize `WF` and `FC` with some random weights and attempt to multiply them to get `WC'` (an approximation of WC) and measure how close it is to `WC`. We do this multiple times using Stochastic Gradient Descent (SGD) to minimize the error. Finally, the `Word-Feature matrix (WF)` gives us the word embeddings for each word where `F` can be preset to a specific number of dimensions. A very important point to remember is that both Word2Vec and GloVe models are very similar in how they work. Both of them aim to build a vector space where the position of each word is influenced by its neighboring words based on their context and semantics. Word2Vec starts with local individual examples of word co-occurrence pairs and GloVe starts with global aggregated co-occurrence statistics across all words in the corpus.

4 Primer on Deep Learning and Modelling approach

The field of artificial intelligence is essentially when machines can do tasks that typically require human intelligence. It encompasses machine learning, where machines can learn by experience and acquire skills without human involvement. Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. Similarly to how we learn from experience, the deep learning algorithm would perform a task repeatedly, each time tweaking it a little to improve the outcome. We refer to 'deep learning' because the neural networks have various (deep) layers that enable learning. Just about any problem that requires "thought" to figure out is a problem deep learning can learn to solve.

4.1 Multi Layer Perceptron (MLP)

Artificial Neural Networks have generated a lot of excitement in Machine Learning research and industry, thanks to many breakthrough results in speech recognition, computer vision and text processing. Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features

$$X = x_1, x_2, ..., x_m$$

and a target y , it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 1 shows a one hidden layer MLP with scalar output. Figure 3 shows a multi layer perceptron shows a one hidden layer MLP with scalar output. Note that all connections have weights associated with them, but only three weights (w0, w1, w2) are shown in the figure.

Input Layer: The Input layer has three nodes. The Bias node has a value of 1. The other two nodes take X1 and X2 as external inputs (which are numerical values depending upon the input dataset). As discussed above, no computation is performed in the Input layer, so the outputs from nodes in the Input layer are 1, X1 and X2 respectively, which are fed into the Hidden Layer.

Hidden Layer: The Hidden layer also has three nodes with the Bias node having an output of 1. The output of the other two nodes in the Hidden layer depends on the outputs from the Input layer (1, X1, X2) as well as the weights

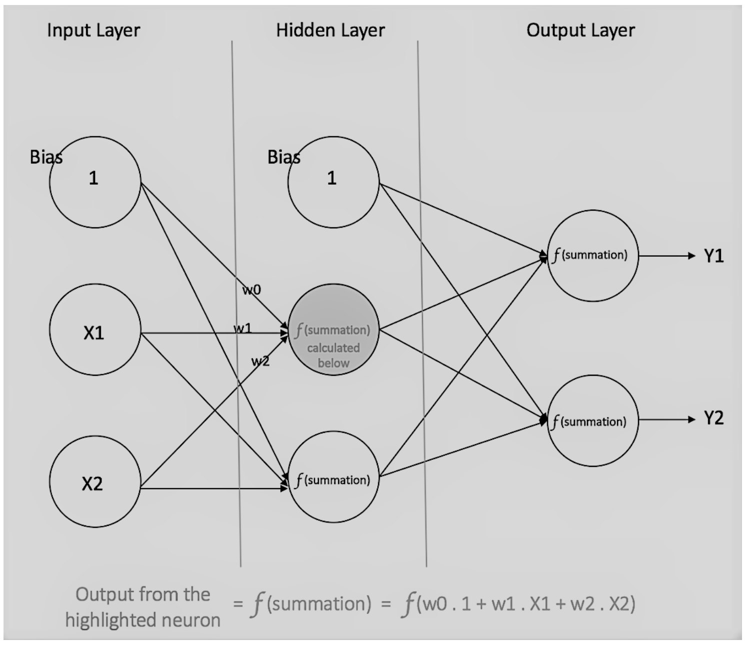
Input Layer
Hidden Layer
Output Layer
Bias
1
X1
X2
Bias
1
w0
w1
w2
f (summation) calculated below
f (summation)
f (summation)
f (summation)
Y1
Y2
Output from the highlighted neuron = f (summation) = f(w0 . 1 + w1 . X1 + w2 . X2)

Fig. 3. a multi layer perceptron having one hidden layer

associated with the connections (edges). Figure 3 shows the output calculation for one of the hidden nodes (highlighted). Similarly, the output from the other hidden node can be calculated. Remember that f refers to the activation function. These outputs are then fed to the nodes in the Output layer.

Output Layer: The Output layer has two nodes which take inputs from the Hidden layer and perform similar computations as shown for the highlighted hidden node. The values calculated (Y1 and Y2) as a result of these computations act as outputs of the Multi Layer Perceptron.

4.2 Convolutional Neural Networks (CNN)

Following the popularization of word embeddings and its ability to represent words in a distributed space, the need arose for an effective feature function that extracts higher-level features from constituting words or n-grams. These abstract features would then be used for numerous NLP tasks such as sentiment analysis, summarization, machine translation, and question answering (QA). Convolutional Neural Networks (CNN) turned out to be the natural choice given their effectiveness in computer vision tasks. A CNN is basically a neural-based approach which represents a feature function that is applied to constituting words or n-grams to extract higher-level features.

Collobert and Weston were among the first researchers to apply CNN-based frameworks to NLP tasks [15]. The goal of their method was to transform words into a vector representation via a lookup table, which resulted in a primitive word embedding approach that learns weights during the training of the network.

In order to perform sentence modeling with a basic CNN, sentences are first tokenized into words which are then transformed into a word embedding matrix (i.e. input embedding layer) of d dimension. Then, convolution filters are applied to this input embedding layer which consists of applying a filter of all possible window sizes to produce what is called a feature map. This is then followed by a max-pooling operation which applies a max operation on each filter to obtain a fixed length output and reduce the dimensionality of the output. And that procedure produces the final sentence representation. By increasing the complexity of the basic CNN model explained above, other NLP tasks such as Named Entity Recognition and POS can be achieved. This requires a window-based approach, where for each word a fixed size window of neighboring words (sub-sentence) is considered. Then a standalone CNN is applied to the sub-sentence and the training objective is to predict the word in the center of the window, also referred to as word-level classification. CNNs fail to model long-distance dependencies, which is important for various NLP tasks. To address this problem, CNNs are coupled with time-delayed neural networks (TDNN). Dynamic CNN (DCNN) is another useful type of CNN that has shown success in different NLP tasks, such as sentiment prediction and question answering.

Overall, CNNs are effective because they can mine semantic clues in contextual windows, but they struggle to preserve sequential order and model long-distance contextual information. Recurrent models (RNN) are better suited for such type of learning and they are discussed next.

4.3 Recurrent Neural Networks (RNN)

Recurrent Neural Networks are specialized neural-based approaches that are effective at processing sequential information. The term “recurrent” applies as they perform the same task over each instance of the sequence such that the output is dependent on the previous computations and results. These sequences are typically represented by a fixed-size vector of tokens which are fed sequentially (one by one) to a recurrent unit. The main strength of an RNN is its capacity to memorize the results from the previous computations and use it for the current computation. This makes RNN a good candidate for modeling context dependencies for inputs of arbitrary length. RNNs have been used to study various NLP tasks such as machine translation, image captioning, and language modeling, among others. Simple RNNs suffer from the infamous vanishing gradient problem, which makes it really hard to learn and tune the parameters of the earlier layers in the network. Long Short Term Memory (LSTM) networks, Residual Networks (ResNets) and Gated Recurrent Networks (GRU) were later introduced to overcome this limitation. An LSTM consist of three gates (input, forget, and output gates), and calculate the hidden state through a combination of the three. GRUs are similar to LSTMs but consist of only two gates and are more efficient because they are less complex. A study shows that it is difficult to say which of the gated RNNs are more effective, and they are usually picked based on the computing power available. Various LSTM-based models have been proposed for the sequence to sequence mapping (via encoder-decoder frameworks) that are suitable for machine translation, text summarization, modeling human conversations, question answering, image-based language generation, among other tasks [16].

5 Design and Solution

5.1 Reading data from the SEC EDGAR and AlphaVantage

As a first step, we download for S&P 500 companies as of November 2018 the list of 8-K document links from the SEC EDGAR website using the BeautifulSoup python library. Then for each link, we download the associated text reports from the SEC EDGAR website using BeautifulSoup and parse metadata such as the date and time of document release and the categories of disclosures made were extracted, while tables and charts were discarded. We also gather the historical stock prices (close, open, adjusted-close) using the Alpha-Vantage python api.

Next, for each document the corresponding release date, one year, one quarter and one month historical moving average price movements were calculated and normalized by the change in the S&P 500 indices. All windows refer to the days that the NYSE and Nasdaq were actually open (non-holiday weekdays). Historical index prices for the Volatility Index (VIX) and GSPC (S&P 500 Index) were downloaded from Yahoo Finance.

(VIX)CBOE Volatility Index - Created by the Chicago Board Options Exchange (CBOE), the Volatility Index, or VIX, is a real-time market index that represents the market's expectation of 30-day forward-looking volatility. Derived from the price inputs of the S&P 500 index options, it provides a measure of market risk and investors' sentiments. It is also known by other names like "Fear Gauge" or "Fear Index." Investors, research analysts and portfolio managers look to VIX values as a way to measure market risk, fear and stress before they take investment decisions (CIK) [17]

GSPC - This is the ticker symbol for the S&P 500 Index. GSPC index is calculated by taking the sum of the adjusted market capitalization of all S&P 500 stocks and then dividing it with an index divisor, which is a proprietary figure developed by Standard & Poor's.

The target feature is calculated as the change in the 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 October 11, 2018, the change in its opening and adjusted close price is calculated, minus the change during the same time for the S&P 500 indices. The normalized changes are labeled as either "up" (> 1%), "down" (< -1%), or "stay"(between -1 and 1%).

Then we combine the 8-K data with financial data obtain from Alpha-Vantage. To do this, we use the company names, tickers and central index keys

(CIK), and match for every stock index the companies with the appropriate Form 8-K.

Because of the size of the data approximately 20000 documents and time needed to scrape and collect it, we had to split them into separate parts for each stock ticker and download them separately and store them as separate csv files. Below in Fig. 4. We show the breakdown for the reports categorized by industry. In Fig. 5. Below we see the breakdown of a report by the year they were released.

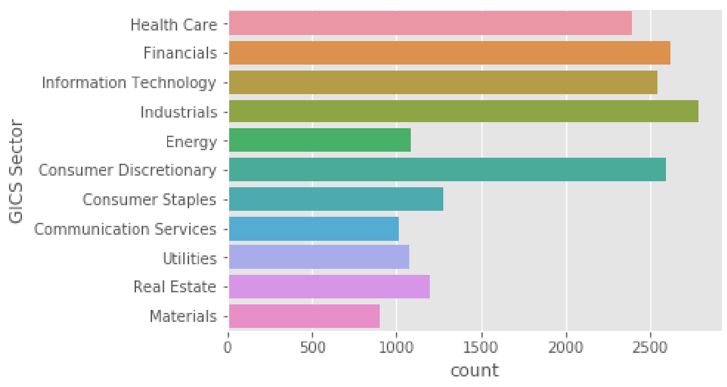

Fig. 4. Breakdown of report counts by Industry

5.2 NLP Exploration

The text is preprocessed by removing stopwords, punctuations and numbers, by lemmatizing and by converting to lower case. This is accomplished using the NLTK WordNet corpus reader in combination with Dask for a multithreading speed boost. All documents are padded with zeros to a uniform length of 33300 words. This cutoff was chosen at the 90th percentile of document lengths in order to preserve more textual information, but prevent the data set from becoming unnecessarily large. The Stanford NLP Wikipedia 2014 + Gigaword 100 dimensions were chosen for pre-trained word embedding's 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. After dropping duplicate samples, and texts for which no release date could be extracted, the final dataset consisted of about 20,000 documents for 505 companies.

5.3 Deep Learning

All categorical features are one hot encoded, and continuous features such as recent stock movements, and closing prices of VIX were standardized to have a

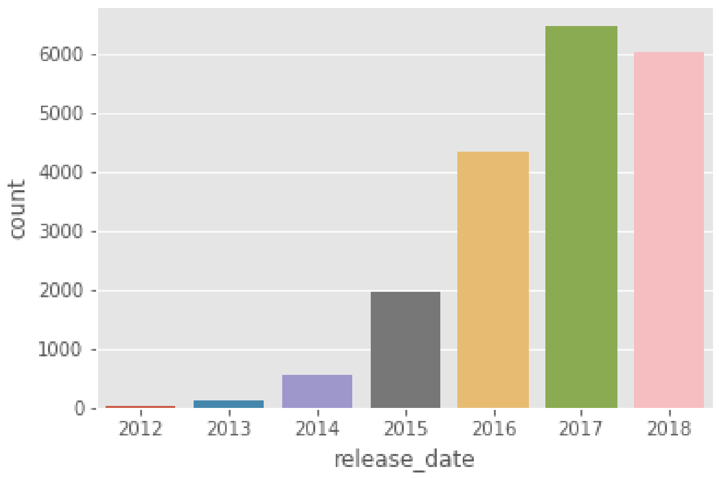

Fig. 5. Breakdown of report counts year of release

mean of 0 and standard deviation of 1. The dataset is 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 is 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. We constructed a model using Keras with a Tensorflow backend, consisting of two input layers (one for text documents, one for features), an embedding layer with the pre-trained GloVe vectors, and a multilayer perceptron(MLP) fully connected network. The network was trained for 10 epochs, with a batch size of 16 due to memory limitation. The model.fit method returns a History callback, which has a history attribute containing the lists of successive losses and other metrics. Further extending the study to use other architectures like RNN, CNN, RNN-CNN we have achieved better accuracy, lesser loss. Figure 6 below shows a high level overview of the architecture from the solution.

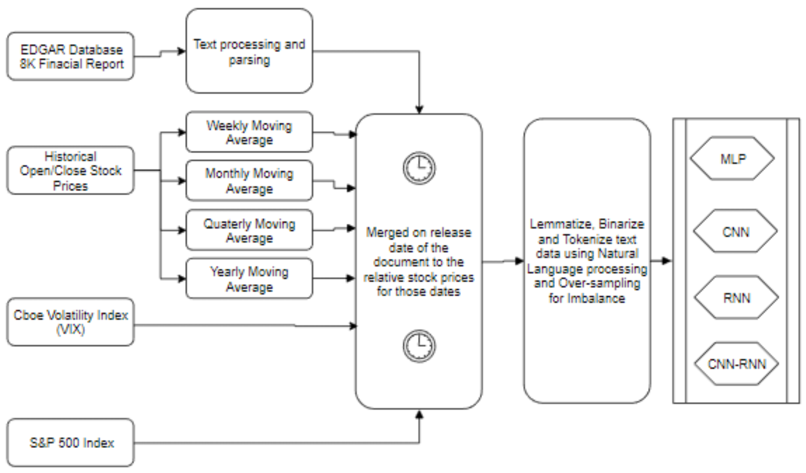
EDGAR Database 8K Finacial Report
Text processing and parsing
Historical Open/Close Stock Prices
Weekly Moving Average
Monthly Moving Average
Quaterly Moving Average
Yearly Moving Average
Cboe Volatility Index (VIX)
S&P 500 Index
Merged on release date of the document to the relative stock prices for those dates
Lemmatize, Binarize and Tokenize text data using Natural Language processing and Over-sampling for Imbalance
MLP
CNN
RNN
CNN-RNN

Fig. 6. Architecture of the Solution

6 Results

The lower the loss, the better a model (unless the model has over-fitted to the training data). The loss is calculated on training and validation and its interpretation is how well the model is doing for these two sets. Unlike accuracy, loss is not a percentage. It is a summation of the errors made for each example in training or validation sets.

In the case of neural networks, the loss is usually negative log-likelihood and the residual sum of squares for classification and regression respectively. Then, naturally, the main objective in a learning model is to reduce (minimize) the loss function's value with respect to the model's parameters by changing the weight vector values through different optimization methods, such as back-propagation in neural networks.

The loss value implies how well or poorly a certain model behaves after each iteration of optimization. Ideally, one would expect the reduction of loss after each, or several, iteration(s). Loss used is the Categorical Cross-Entropy Loss.

The accuracy of a model is usually determined after the model parameters are learned and fixed and no learning is taking place. Then the test samples are fed to the model and the number of mistakes (zero-one loss) the model makes are recorded, after comparison to the true targets. Then the percentage of misclassification is calculated. AUC-Roc visualizes performance classification. The higher, the better.

We have performed MLP, RNN, CNN and RNN-CNN to the dataset and the result can be seen in the table and graphs below. Table 2. Below shows the comparison of result across the different models. Graph in Fig. 6., Fig. 7. And Fig. 8. Below show the comparisons of losses, accuracy and AOC-ROC's across different MLP, RNN, CNN and RNN-CNN respectively.

Model	Loss	Accuracy	AOC-ROC
MLP	1.1611	58%	0.7129
CNN	0.8077	65%	0.8046
RNN	0.8017	66%	0.8166
CNN-RNN	0.7990	68%	0.8162

Table 2. Comparison of Results across Models

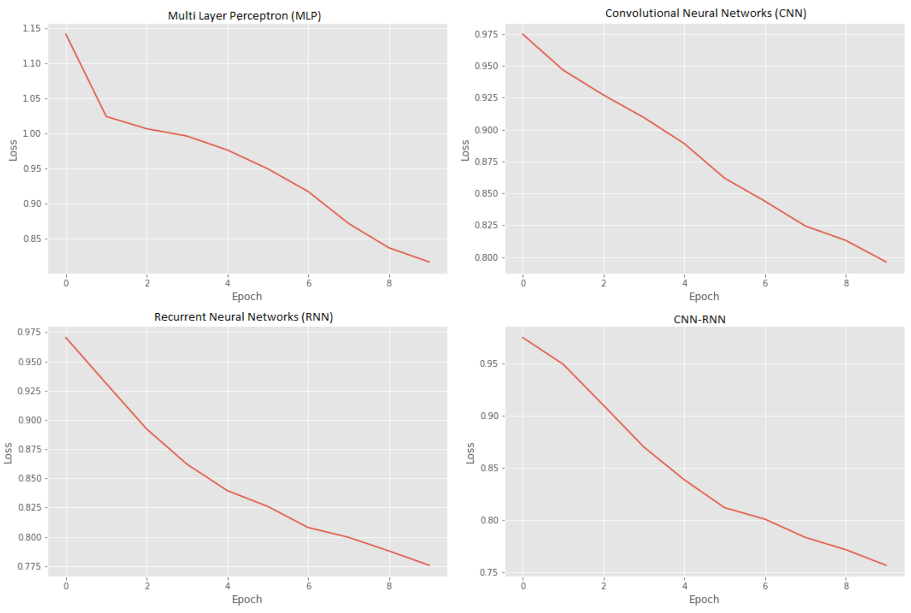

Fig. 7. Loss from calculated model against test data

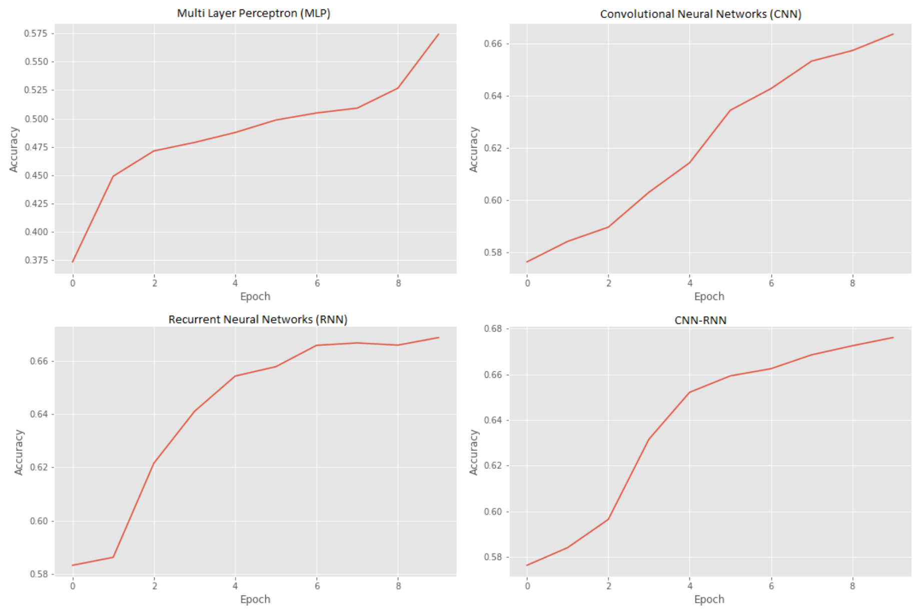

Fig. 8. Accuracy from calculated model against test data

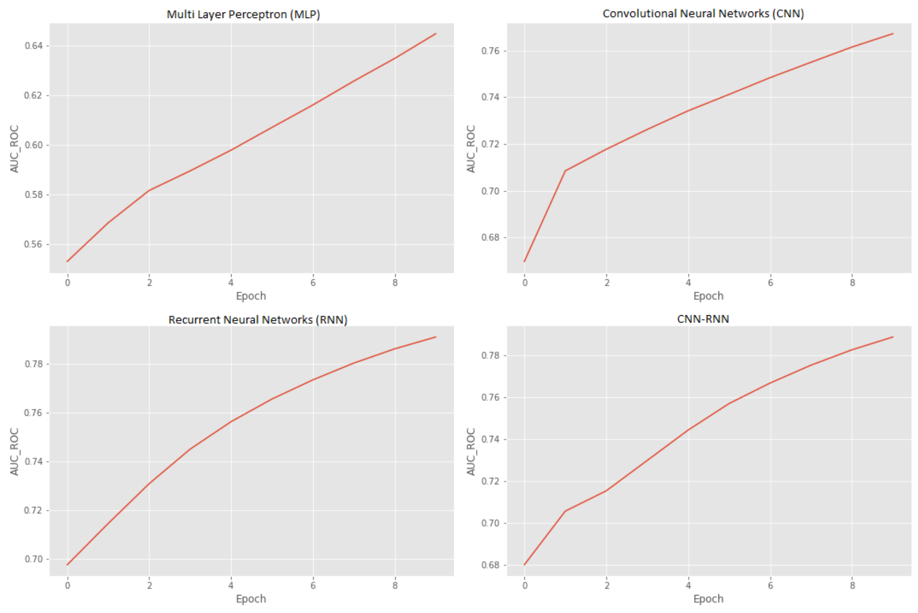

Fig. 9. AUC-ROC curve from calculated model against test data

7 Ethics

Natural Language Processing (NLP) systems analyze and/or generate human language, typically on the user's behalf. One natural and necessary question that needs to be addressed in this context, both in research projects and in production settings, is the question how ethical the work is, both regarding the process and its outcome.

Ethics, the part of practical philosophy concerned with all things normative (moral philosophy, answering the fundamental question of how to live one's life) permeates all aspects of human action. Applying it to Natural language Processing (NLP), we can ask the following core questions:

1. What ethical concerns exist in the realm of NLP?
2. How should these ethical issues be addressed?

At the time of writing, automation using machine learning is making great practical progress, and this includes NLP tasks, but is by no means limited to it. As more areas in life are affected by these new technologies, the practical need for clarification of ethical implications increases; in other words, we have reached the level where the topic is no longer purely academic: we need to have solutions for what a driverless car should morally do in situations that can be described as ethical dilemmas, and in language and speech-enabled system ethical questions also arise. Governments and NGOs are also trying to come to grips with what machine learning, which NLP also relies on, means for policy making (Armstrong, 2015).

Prior work on the topics of ethics in NLP can be grouped into three categories. First, there is the general body of literature covering applied ethics and moral philosophy. Second, within computer science, there are discussions around big data, data mining and machine learning and their ethical implications, often focused on privacy and bias/discrimination. Few if any of these works have mentioned issues specific to language processing, but a lot of the unspecific issues also do apply to NLP.[18] Third, there is a body of works on professional ethics, often talked about in the context of curriculum design for computer science teaching (didactics of computing), governance and professional conduct and legal/ethical aspects of computing (computing as a profession, continued professional development).

The question we will pose in here is does automation using advances in deep learning and computing power, reduce the ethical awareness and responsibility of financial professionals. Who will be responsible for Artificial Intelligence (AI) decisions and actions made. Since AI models are essentially a black box, and the more mature the model gets, the more you're not able to explain it. So if you as a company are not able to explain your own model, would you stand behind the decisions taken by your AI algorithm. We inherently think that the ethical awareness of financial professionals is already very low, so to bring a third party AI based platform that is making decisions for them, will more or less make these professionals feel LESS RESPONSIBLE for these decisions.

8 Conclusion

Textual data can sometimes be deceiving when delivering the right frame of speech. 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 (textual data) and asses swings in a company's stock price. Through this project, it helped us understand the basics of Natural Language Processing. It was interesting to know about how to go from text data to vectors of numbers and applying Deep Learning techniques that can help to influence the stock market of a company.

We also learned about scraping our own data which made this project a little more interesting and challenging. To get more insights into use for different deep learning architectures, we learnt by reading through research papers and Keras help portal. As any project that does not take a straight path towards completion, we hit several roadblocks while implementing this project like, handling humongous amount of data, figuring out the hardware requirements for data processing and modeling and learning new techniques to improve the project.

Even though you can't bet your money on the stock from this project, this work can be treated as a solid understanding of the basics of Natural Language Processing. Looking at the results after 10 epochs, the RNN and RNN-CNN models generalize the best. We wanted to focus on the CNN-RNN model as it has had a comparable accuracy of the RNN model and was trained in half the time. The best accuracy, we were able to achieve is 68%. Literature scan for prediction using legacy techniques could only achieve accuracy less than 60%. This is a huge improvement from the legacy results.

9 Future Work

The results achieved using the new techniques can be further improved by building models specific to separate industries and hence isolate the effects due to other factors occurring in an industry. Also future study can include extending training for more number of epochs to see the effects of improvements when applying it to the training data.

LSTMs are very powerful in sequence prediction problems because they're able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price, hence also using other deep learning techniques like RNN-LSTM needs to be explored. Further adding other price indicators like daily volume, price volatility, and other indexes could greatly improve the result. Lastly, another improvement could be adding twitter sentiments and company news as features to see their effects on improving the model.

References

1. Sudhindra Bhat : Published June 2007, *Security Analysis and Portfolio Management.*

2. Cram101, Zvi Bodie : Just the Facts101 9th Edition *Essentials of Investments*
3. B. T. O'Connor. *Statistical Text Analysis for Social Science.* Carnegie Mellon University Pittsburgh, PA, USA, 2014.
`http://brenocon.com/phdthesis/thesis.pdf`
4. W Kenton: Investopedia, *Form 8-K*
`https://www.investopedia.com/terms/1/8-k.asp`
5. U.S. Securities and Exchange Commission, *Form 10-K*
`https://www.sec.gov/fast-answers/answers-form10khtm.html`
6. H. Lee, M. Surdeanu, B. MacCartney, D. Jurafsky. *On the Importance of Text Analysis for Stock Price Prediction.*
`https://nlp.stanford.edu/pubs/lrec2014-stock.pdf`
7. S. Kogan, D. Levin, B. Routledge, J. Sagi. *Predicting Risk from Financial Reports with Regression.*
`http://www.aclweb.org/anthology/N09-1031`
8. *EDGAR.*
`https://searcherp.techtarget.com/definition/EDGAR`
9. *Accessing EDGAR Data.*
`https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm`
10. *Modern Deep Learning Techniques Applied to Natural Language Processing.*
`https://nlpoverview.com`
11. S. Mahapatra, *Why Deep Learning over Traditional Machine Learning?.*
`https://towardsdatascience.com/`
12. Y. Bengio, R Ducharme, P Vincent, C Jauvin. *A Neural Probabilistic Language Model*, Journal of Machine Learning Research 3 (2003) 1137–1155
`http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf`
13. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean *Distributed Representations of Words and Phrases and their Compositionality.*
`https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`
14. T. Young, D. Hazarika, S. Poria, E. Cambria, *Recent Trends in Deep Learning Based Natural Language Processing.*
`https://arxiv.org/pdf/1708.02709.pdf`
15. R. Collobert, J. Weston. *A Unified Architecture for Natural Language Processing Deep Neural Networks with Multitask Learning*
16. E. Saravia *Deep Learning for NLP: An Overview of Recent Trends https://www.kdnuggets.com/2018/09/deep-learning-nlp-overview-recent-trends.html/2*
17. J. Chen *VIX - CBOE Volatility Index.*
`https://www.investopedia.com/terms/v/vix.aspixzz5WauaJ1mo`
18. Y. Wilks *Close Engagements with Artificial Companions https://www.mitpressjournals.org/doi/abs/10.1162/COLI-r-00053*
19. D. Manuel, *Knowing the Difference Between a 10-K, 10-Q and an 8-K.*
`https://www.davemanuel.com/2008/07/31/knowing-the-difference-between-a-10-k-10-q-and-an-8-k/`
20. J. Pennington, R. Socher, C. D. Manning, *GloVe: Global Vectors for Word Representation.* Computer Science Department, Stanford University
`https://nlp.stanford.edu/pubs/glove.pdf`
21. K. Chen, *Use Python to download TXT-format SEC filings on EDGAR.*
`http://www.kaikaichen.com/?p=59`
22. Wikipedia *Multilayer perceptron.*

23. Y. Goldberg, 2015 *A Primer on Neural Network Models for Natural Language Processing.*
`http://u.cs.biu.ac.il/ yogo/nnlp.pdf`