Attention-based Stock Price Movement Prediction Using 8-K Filings

Masoud, Mohamed
masoud@stanford.edu

Abstract

Several financial applications use stock forecasting models. These models utilize a rich mixture of quantitative financial and stock related text data to extract signals that would better capture the stock dynamics. Firm stock prices can be impacted by market trends, world events and more pertinently by companies' specific events. The SEC mandatory 8-K filings have a short and long term impact on stock prices. In this project, we investigate deep learning technique(s) to predict the short-term stock price movement (UP, DOWN and STAY) in response to financial events reported in 8-K documents. We introduce a parallel neural architecture that combines two levels of attention. First, an attention-based BiLSTM layer to encode the linguistic representation of events contained in the 8-K report. Second, an attention-based layer that computes the event relevancy to the stock price movements. Next, the model uses feed forward layer to combine the relevancy based linguistic representation with numerical and categorical features pertinent to the classification problem. We have conducted experiments using preprocessed versions of Lee et al. 2014 Stanford's 8-K stock events dataset. The experimental results of the presented models show improvement in using deep-learning attention based techniques to extract the linguistic signals compared to the baseline models.

1 Introduction

Stock prices are impacted by a variety of factors including the vast amount of new information. The efficient-market hypothesis (EMH) [3] suggests that stock prices reflect all currently available information and any price changes based on the newly revealed relevant information. Analyzing such information in real time is important for trading strategies, financial advisors and individual investors. One major body of new information that has immediate impact on stock price movements resides in the 8-K financial reports, which the publicly listed U.S. companies are required to file by the SEC. These reports describe companies' major business events such as financial disclosures, mergers and acquisitions, bankruptcies, change in management, etc. In this project, we present an attention based parallel architecture that is customized to build and evaluate four different models that extract relevant linguistic signals from these reports and study the impact of the 8-K events' information on the prediction of stock price movements (UP, DOWN and STAY).

2 Related Work

Various forms of stock related text data have been considered in conjunction with the quantitative data to extract signals that would better capture the stock dynamics. There are many attempts to use linguistic features to enhance stock prediction models. Most related to this project is the research presented by Lee et al. [1], (2014)., titled ”On the Importance of Text Analysis for Stock Price Prediction”, which introduced a corpus [5] that aligns the changes of stocks' prices with the release of 8-K reports. Additionally, Lee et al. [1], implemented a non-deep learning model to forecast companies' stock price changes (UP, DOWN, STAY) in response to financial events reported in the 8-K documents. Along the linguistic features, a set of 21 non-linguistic features (numeric and categorical) have been considered by the model. The features fall into four types: 1- Earnings surprise: The gap between consensus and reported earnings per share (EPS), 2- Recent movements: The movements of the company's stock price calculated for multiple periods before the event (1 week, 1 month, 1 quarter and 1 year), 3- VIX index: the volatility index value (ticker: VIX) at the market close before the 8-K report is released and 4: Event type of category: one or more event type number(s) reported in the document. Olmsted et al. [4], 2018, (CS230 project group), introduced a deep learning model for *binary* classification task (UP, DOWN) that uses Lee et al., dataset and similar non-linguistic features.

Similar to Lee et al., our proposed models share the same classification task with the same three classes and use preprocessed versions of the same dataset. In contrast, we are proposing an attention based deep neural architecture to capture the linguistic signals in the 8-K fillings. Moreover, we are using a smaller subset of the non-linguistic features, as shown in section 3.2, that allows to isolate the effectiveness of using attention based models in extracting the linguistic signals from the 8-K reports.

3 Data, Preprocessing and Features

The dataset contains variable length 8-K reports that describe significant business events for all S&P 500 companies between 2002 and 2012. Each report is divided into 9 categorical sections describing a *maximum* of 31 possible events (subsections - see SEC website [1]). Since the purpose of the 8-K document is to report certain event(s), each example document only contains information about these specific events, therefore not all the 31 items are present in the report. In addition to the 8-K reports, the dataset contains numeric financial data such as the consensus (estimated) and reported Earnings Per Share (EPS) and the adjusted open and close prices around the reported event.

3.1 Preprocessing and Labeling

Guided by Lee et al., 2014, in order to isolate the impact of the 8-K document release, the before and after stock prices' differences are normalized by subtracting the change in S&P 500 index prices during the same period. Therefore, the normalized differences would only reflect the effect of the report release. The normalized rates (NR) are then used to label the movements: 1- UP if NR $> 1\%$, DOWN if NR $<$ -1% and STAY if NR is within 1%. In order to label the dataset, for each report, the company's and the index ˆGSPC close prices of the previous day, the open and close prices of the report filing day and the open prices of the following day are queried from dataset or Yahoo historical prices. Each report text has been parsed to specify the event date and time, events types and events text. The labeled and legible documents with 3000 words or less compiled 138294 reports dataset considered by the presented models.

3.2 Non-Linguistic Features

To study the impact of using both the linguistic encoded features and the non-linguistic features in the classification problem, the models described in section 4 use the non-text features: 1- Event type categorical variable, 2- Earnings surprise. For the EPS surprise, the preprocessing of EPS html files in the Lee et al. [1], 2014, Stanford's dataset resulted a labeled dataset of 16185 examples.

4 Approach

Guided by the intuition about the task and the data, we have considered two main points in designing the neural model. First, for the linguistic features, the natural choice for text sequence representation is the recurrent neural network (RNN) models. However, the 8-K form can be a lengthy document with thousands of words which would challenge the effectiveness of RNN and even the LSTM or GRU sequence models. Fixed size one-dimensional convolutional neural network (a 1D CNN) is a faster alternative to RNN for sequence representation. However, for the variable length long text sequence as in the case of 8-K documents, the CNN based representation may require a multi-layer

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

architecture to capture the long-distance dependencies. In this project, we present an attention-based architecture that uses a preprocessed dataset of the parsed events in each sample 8-K document. Each sample document has one or more events. The set of events' text sequences would in parallel be fed into a bidirectional LSTM encoder to produce a representation vector for each event. An attention mechanism would then be applied to pick up the most relevant information from each individual event sequence.

Another important consideration is that different events in the 8-K report may differently impact and have relative contribution on stock price movements. Additionally, similar events in the same category may have different effects on different companies. To model the event relative impact, we use relevance attention mechanism similar to the (ATT-ERNN) method introduced in Liu et al. [2], 2017. The model computes a relevance (impact) score for each event and uses the scores to compute events' attention distribution. The distribution is then used to produce a weighted sum of the encoded events vectors.

4.1 Architecture and Formulation

The model(s) architecture as shown in figure 1, can be divided into three layers: Encoding Layer, Event Relevance Layer and Feed Forward Layer. The encoding layer contains one or more of the 8-K event encoding module, the event relevance layer contains one instance of the events relevance attention module and the feed forward layer contains one instance of the feed forward prediction module.

4.1.1 8-K Event Encoding Module

This module takes an input of *one* event (item) text and outputs an attention vector.

- **Word Embedding Layer:** Using TorchText, the pre-trained GloVE word embedding is looked up for the m sized event text, yielding $x_1, ..., x_m | x_i \in \mathbb{R}^{ex1}$, where e is the embedding size.
- **Bidirectional LSTM Encoder:** The events' text embeddings are then fed to LSTM bidirectional encoder, yielding the forward and backward hidden states and cell states (h_i^{enc} , c_i^{enc}).

$$h_i^{enc} = [\overleftarrow{h_i^{enc}}; \overrightarrow{h_i^{enc}}] \text{ where } h_i^{enc} \in \mathbb{R}^{2hx1} \text{ and } h \text{ is the hidden state size} \tag{1}$$

- **Attention:** This subcomponent extracts relevant signals from the event text by allowing encoder to attend to itself using self-attention *like* mechanism. Let $H = [\ h_1^{enc}, ... ,h_m^{enc}]$ be a matrix consisting of hidden vectors produced by the BiLSTM encoder. We compute the unnormalized alignment attention score $f_{att}(H)$ using:
  - Multiplicative Attention:

$$e = \tanh(W_a^T H) \text{ , where } W_a \in \mathbb{R}^{2hx1} \tag{2}$$

  - Simplified Additive Attention:

$$e = v_a^T \tanh(W_1 H) \text{ , where } v_a \in \mathbb{R}^{hx1} \text{ and } \in \mathbb{R}^{hx2h} \tag{3}$$

  The attention output a is then computed as the weighted sum of the encoder hidden state vectors

$$\alpha = \text{Softmax}(e) \tag{4}$$

$$a = \sum_i^m \alpha_i h_i^{enc}, \text{ where } a \in \mathbb{R}^{2hx1} \tag{5}$$

4.1.2 Events Relevance Attention Module

Guided by the ATTN-ERNN model introduced in Liu et al. [2], 2017, this module uses a self-attention *like* mechanism to compute the relevancy of different events. Let $A = [\ a_1, ... ,a_K]$ be a matrix consisting of attention outputs produced by K event encoding modules. The relevancy scores for the K events in the 8-K report are computed by:

– Multiplicative Attention:

$$e_r = \text{tanh}(W_r^T A) \text{ , where } W_r \in \mathbb{R}^{2h\text{x}1} \tag{6}$$

– Simplified Additive Attention:

$$e_r = v_r^T \text{ tanh}(W_{r_1} A) \text{ , where } v_r \in \mathbb{R}^{h\text{x}1} \text{ and } W_{r_1} \in \mathbb{R}^{h\text{x}2h} \tag{7}$$

The attention output is then computed as the weighted sum of the K attention output vectors [a_1, ... ,a_K]. Linear projection and nonlinear tanh with dropout regularization are then applied to produce the overall attention-based linguistic features encoding vector $o \in \mathbb{R}^{h\text{x}1}$

$$\beta = \text{Softmax}(e_r) \tag{8}$$

$$re_{att} = \sum_i^K \beta_i a_i \tag{9}$$

$$o = \text{Dropout}(\text{tanh}(W_{proj} re_{att}) \text{ , where } W_{proj} \in \mathbb{R}^{h\text{x}2h} \tag{10}$$

4.1.3 Feed Forward Prediction Module

The output of the events relevance module is then concatenated with the stock j numerical and categorical vector s_j. $c = [o; s_j]$. That would then feed a fully connected neural network $l = FC(c)$ to capture the nonlinear interactions between the linguistic and non-linguistic features. Finally, Softmax output layer on the logits $l \in \mathbb{R}^{3\text{x}1}$ yield the stock j movement prediction (UP, DOWN and STAY). The cross-entropy loss function is used for model optimization. The batch average loss:

$$J(\theta) = -\frac{1}{b} \sum_{i=1}^{b} \sum_{c=1}^{3} y_{i,c} log(\hat{y_{i,c}}) \tag{11}$$

– b: batch size,

– $y_{i,c}$: one hot represented ground truth label c for example i

– $\hat{y_{i,c}} = \arg\max_y$ Softmax(l) is the estimated probability of class c for example i

4.2 Models

An Incremental approach has been adopted in developing the models presented in this project. Four models have been developed and evaluated. The models are constructed

4.2.1 Single-Event-Text-Only (model-1)

The first developed model used linguistic only features and considered the whole 8-K report as one event. Thus, the model is composed of one instance of event encoding module and feed forward prediction module. The multiplicative attention was chosen as the attention mechanism. It is considered as a faster choice compared to additive attention and produced similar results on the cross validation set. This model was trained, validated and tested on the labeled dataset of 138294 examples.

4.2.2 Single-Event-Text-EPS (model-2)

This model uses the aligned numeric EPS surprise along the linguistic features. It has a similar architecture with a difference that the feed forward prediction module has a concatenation layer to combine the numerical feature and the encoded linguistic features. This model was trained and evaluated on a dataset of 16185 examples.

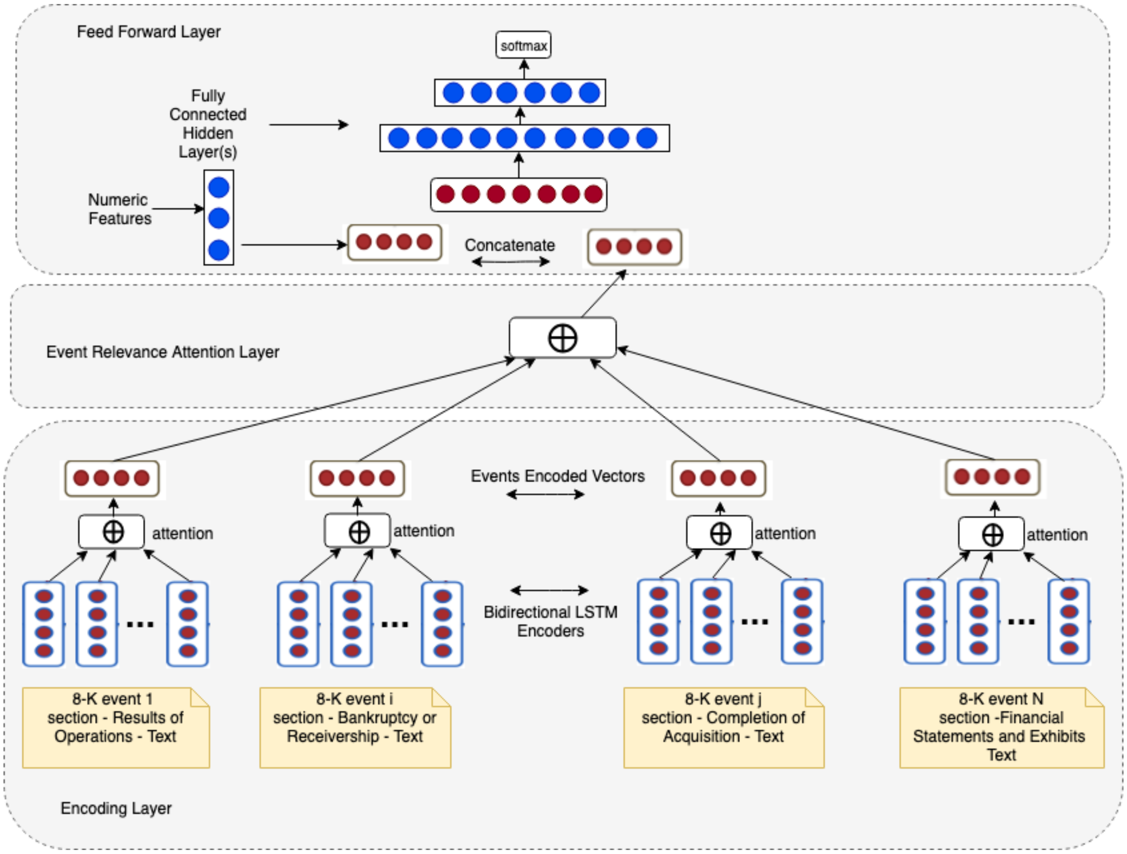
Feed Forward Layer
softmax
Fully Connected Hidden Layer(s)
Numeric Features
Concatenate
Event Relevance Attention Layer
⊕
Events Encoded Vectors
attention
attention
attention
attention
Bidirectional LSTM Encoders
8-K event 1 section - Results of Operations - Text
8-K event i section - Bankruptcy or Receivership - Text
8-K event j section - Completion of Acquisition - Text
8-K event N section -Financial Statements and Exhibits Text
Encoding Layer

Figure 1: Model Architecture

4.2.3 Single-Event-Text-EventsTypes (model-3)

This model uses events types categorical variable along the linguistic features. It has a similar architecture with a difference that the feed forward prediction module has a concatenation layer to combine one hot vector representation of the possible 31 events and the encoded linguistic features. This model was trained and evaluated on a dataset of the same size as the first model.

4.2.4 Multi-Event-Text-EventsTypes (model-4)

This model utilizes the full architecture in figure 1. The feed forward layer is similar its model-3 counterpart. The encoding layer is composed of 32 parallel event encoding modules. 31 encoders compute the representation for all the possible events (items) reported in document and one additional encoder for the exhibits section [2] that frequently appears as a separate section. The dataset created for this model is composed of 127642 examples. Similar to model-3, each example contains 32 dimensional one hot vector representing the 31 events types and one binary value for the exhibit section. For example, for documents that report “Completion of Acquisition or Disposition of Assets“ and “Regulation FD Disclosure“ events and has an exhibit section, the one hot vector would have ones in the 5^{th}, the 29^{th} and the 32^{nd} indices and zeros otherwise. In contrast, the 8-K document text is parsed and divided into a set of sub-texts corresponds the items reported in the document. Each item text feeds its corresponding encoder. For the same example, the 5^{th}, the 29^{th} and the 32^{nd} encoders would be fed with the respective item text. The other 29 encoders would take one word “<blank>“ text as input.

[2] The "Exhibits" section are referenced from all sort of events reported in the 8-K filings not only referenced by ’Financial Statements and Exhibits’. The ’Financial Statement and Exhibits’ is the most reported event in the dataset as shown in figure 2 and does not necessarily reference to a separate "Exhibits" section.

The event relevance module would then use attention mechanism described in section 4.1.2 to compute the events relevancy scores and produce the document level attention-based linguistic features vector. It is important to note that in order to remove the impact of events that are not reported in the document, we use the input one hot vector to generate masks and set high negative value (-Inf) to relevancy scores e_r calculated by equation (6) or equation (7). That would result of zero β weights calculated by equation (8) for these events.

5 Experiments and Results

The models were compared based on accuracy evaluation metric. To setup the experiments, the preprocessed datasets were divided 80/10/10 for train, validation and test sets. We chose pre-trained GloVE (42B tokens, 1.9M vocabulary and embedding size of 300). We ran cross validation on a set of hyper-parameters for first two models. The hyper-parameters are summarized in Table 1. Due to the time constraints and the slow training time specially for the multi-event model, the selected hyper-parameters were used in training the third and forth model. The smaller learning rate has shown a better convergence characteristics, as the train loss increases and accuracy decreases later at training time. The regularization with higher drop out rate and L2 weight decay of 0.001 helped reduce the average loss and increase average accuracy on both hold out cross validation and test sets.

Table 1: Hyper-parameters

Hyper-Parameter	Grid	Selected
BiLSTM hidden size h	[128, 256]	256
FC hidden layer size	[128, 256, 512]	512
Learning rate	[0.001 , 0.0001]	0.0001 - Decaying
Dropout rate	[0.1, 0.3, 0.5]	0.5
Weight-decay (L2)	[0, 0.01, 0.001]	0.001

The test results were compared to the baselines and models presented Lee et al. [1], as shown in Table 2. The Single-Event-Text-Only model-1 seems to outperform all the other models with average accuracy of 57.02%. Despite the lengthy 8-K documents with hundreds of words (less than 3000 words), the event attention-based encoder was able to extract some signals from the text pertinent to the classification.

The Single-Event-Text-EPS model-2 underperformed with 51.92% average accuracy. Although this model adds the pertinent numerical EPS surprise feature to the linguistic features, the model overfits the *small* training dataset and does not generalize well (training set average accuracy = 58.92%).

The Single-Event-Text-EventsTypes model-3 performance is marginally close to model-1. The document events types categorical variable did not seem to add predictive power to the model along with the linguistic features. However, this model might have performed slightly better, if more hyper-parameter tuning was performed.

The Multi-Event-Text-EventsTypes model-4 slightly underperformed model-1 and model-3. The underperformance is attributed to that fact that the multi-event dataset is sparse. The dataset is largely imbalanced and dominated by few events, see figure 2. Although, the event relevance attention layer would only consider the linguistic representations of the reported events in the document ($\beta > 0$), the event level encoders would have smaller sample size with actual events' text (not blank) to learn the linguistic representation of their respective events. Accordingly, for the less represented class of events, the attention based encoders' ability to extract relevant linguistic signals is largely impacted.

Overall, the results suggest that the attention-based architecture expectedly provides an improved approach to extract relevant linguistic signals from the 8-K documents. The results also suggest that the architecture requires a large enough training sample size to better capture the linguistic features and have better out of sample generalization characteristics. Moreover, the overall results of the best performing models are very promising and likely to be significantly improved if the linguistic features are combined with the pertinent non-linguistic features as the models presented by Lee et al. [1]

Table 2: Performance

Model	Accuracy
Random Guess*	33.3
NMF 200*	55.3
Ensemble*	55.5
Single-Event-Text-Only	**57.02**
Single-Event-Text-EPS	51.92
Single-Event-Text-EventsTypes	**56.71**
Mutli-Event-Text-EventsTypes	**56.07**

* Lee et al., [1] Models.

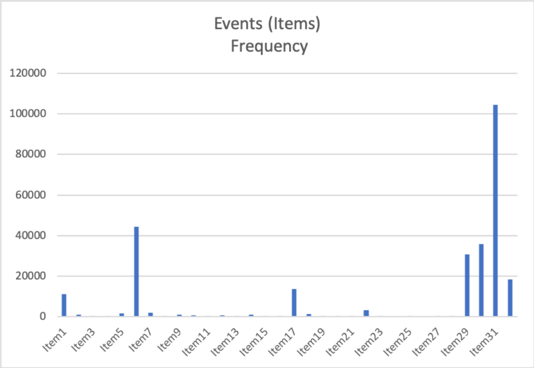

Figure 2: Events (Items) frequency in the dataset

6 Analysis

The deep learning models presented in this project aim to investigate an attention based architecture to extract the linguistic signals from the 8-K documents that help predicting the short-term dynamics of the stock prices. In conjunction with the linguistic features, model-2 uses the EPS surprise as a pertinent financial indicator of the stock momentum. An example to analyze the performance of the models is the 8-K report filed by Apple (AAPL) on April 19, 2006. In this report, there is one event type titled ”Financial Statements and Exhibits”. Here is a relevant snippet from the "Exhibits" section in the report:

... We've generated over $10 billion in revenue and almost $1 billion in earnings in the first half of fiscal 2006," said Steve Jobs, Apple's CEO ... We're very pleased to report the second highest quarterly sales in Apple's history, resulting in year-over-year revenue growth of 34 percent and earnings growth of 41 percent ...

The single event models 1 and 3 picked up on the positive linguistic signals presented in the report and correctly identified the UP label. The EPS surprise aligned with this report is 9.3%. Accordingly, model-2 correctly predicted UP label. To further investigate the impact of the EPS surprise on model-2 predictions, we manually changed the value to -9.3% which resulted into STAY label prediction. The result may be attributed to the conflicting positive signals from linguistic features and the negative EPS surprise signal.

The multi-event model incorrectly predicted STAY label for this example. However, the model nearly classified the example with the correct label (UP) as the feed forward prediction module computed the class probabilities of [P_{DOWN} = 0.32, P_{UP} = 0.338, P_{STAY} = 0.342]. Although, the event reported in this example is item 31 ('Financial Statements and Exhibits'), the bulk of the document text including the positive snippet resides in the "Exhibits" section. Figure 2 shows that item 31 is much more presented than the exhibits section (item 32). Therefore, the 32^{nd} encoding module's ability to extract the positive sentiment is slightly impacted since it has been trained on fewer examples with actual linguistic signals.

Another example to assess the performance of the models, is the Visa Inc. report on October 29, 2008. The EPS surprise aligned with this report is 3.57%. Here is a relevant snippet from the report:

....On a GAAP basis, the Company reported a net loss of $356 million ...We remain intensely focused on helping our financial institution and retail clients through this difficult period . . . providing them with products and services that build deeper cardholder relationships and boost their own bottom line ... fiscal fourth quarter transactions processed by VisaNet, were 9.6 billion, an 11% increase over the prior year

All the models incorrectly classified the stock price movement in response to this report as STAY instead of the correct label DOWN. Although, the report is highlighting a difficult period and net losses, there is still positive phrases and words such as "deeper cardholder relationships and boost their own bottom line" and "11% increase over the prior year". This misclassification is attributed to the bias in the language used by the companies in reporting the both the positive and negative events. This highlights the need for accurate quantitative non-linguistic features in conjunction with the linguistic features to produce more accurate results.

The qualitative analysis of examples such as the above shows that our proposed attention-based models are able to learn the sentiment of the text and extract the relevant linguistic signals that help with the short term stock dynamics predictions. From the above examples snippets, our attention models are likely to focus on relevant contextual sentiment words and phrases such as "pleased", "highest quarterly sales", "revenue growth", "earning growth, "losses", "helping our financial institution and retail clients", "difficult period" and "boost their bottom line" which highlights the positive and negative signals for these reports. The models are likely to attend to similar words and phrases and extract relevant positive, neutral and negative signals from the training data sample and produce linguistic encoded features that aid the classification model.

Quantitatively, the results of the models, which are mainly focused on the linguistic features, are very promising and makes a strong case and a good benchmark for using the proposed or similar attention based architectures that combine the linguistic features with a variety of non-linguistic financial quantitative features for accurate stock dynamics predictions.

7 Conclusion and Future Work

In this work, we proposed an end-to-end attention based classification model architecture that predicts the stock price movements based on the linguistic features extracted from 8-K documents. The overall results suggest that the proposed models successfully extract the relevant linguistic signals from the report that help with the stock dynamic predictions. The analysis highlights the strengths and the limitations of the models and how important to combine the linguistic features and non-linguistic feature to the performance of the model predictions. There are few aspects that we would like to address to improve the models performance:

Data Collection: As a very important first step we would start gathering public data and building a large and less sparse dataset. The dataset should have better representation of the different 8-K events. It is also imperative to collect and align non-linguistic financial features such as the temporal prices, EPS surprise, VIX, fundamental and macroeconomic factors, etc.

CNN Encoders: It seems that the relevant signals in the 8-K documents are largely localized in different areas across the document. The Apple example is a case in point. The sentiment of the document was localized over a few sentences in the middle of the 2050 words document. Therefore, CNN encoders may have an advantage over BiLSTM to speed up the training and better extract the linguistic features from the lengthy 8-K documents.

Data Sources: The proposed architecture is highly modular as described in section 4 and can easily incorporate other linguistic data sources such as 10-K and 10-Q documents, financial news and other financial text sources.

Hyper-Parameter Tuning: We would allocate more time for hyper-parameter tuning. Time and computational budget constraints largely affected the hyper-parameter tuning of model-3 and model-4. It would be important to investigate the hyper-parameters grid to find the best set of parameters for these models.

Acknowledgments

Thanks to the CS224 teaching staff for all the support and guidance, specially the project mentor, Praty Sharma. Thanks to the Lee et al. [1] and the Stanford NLP group for providing this valuable dataset that helps the researchers and practitioners in the field.

References

[1] Lee H., Surdeanu M., MacCartney B. Jurafsky D. (2014). On the importance of text analysis for stock price prediction. In Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014 (pp. 1170-1175). European Language Resources Association (ELRA).

[2] Liu J., Chen Y., Liu K., Zhao J. (2017) Attention-Based Event Relevance Model for Stock Price Movement Prediction. In: Li J., Zhou M., Qi G., Lao N., Ruan T., Du J. (eds) Knowledge Graph and Semantic Computing. Language, Knowledge, and Intelligence. CCKS 2017. Communications in Computer and Information Science, vol 784. Springer, Singapore

[3] Malkiel BG. The efficient market hypothesis and its critics. The Journal of Economic Perspectives. 2003 Mar 1;17(1):59-82.

[4] Olmsted W., Abdelghany T, Kanagal K. (2018) Predicting Stock Movement Using Form 8-K Transcripts, CS 230 Course Project.

[5] https://nlp.stanford.edu/pubs/stock-event.html.

Github

https://github.com/masoudML/CS224N/tree/master/project