CS230

# Investing with NLP: Market Predictions

**Alex Sareyan**
June 10, 2018

## Abstract

Although financial research has long sought quantitative metrics to explain equity pricing, such relationships are often quickly exploited as soon as they are discovered. As a supplemental means of predicting market movements, text data offers some promising features: it contains sentiments that may not be captured by metrics alone, it is relatively abundant, and additionally it might be used to identify investors who are consistently prescient (or, just as interestingly, misguided). In this project, I used a recurrent neural network (RNN) consisting of long short-term memory (LSTM) units to predict daily market movement based on news headlines alone. Without incorporating additional data, the model was able to predict directional movements in the Dow Jones Industrial Average approximately 57 percent of the time.

## 1 Introduction

In the world of equity research and investment, the idea that it is possible to identify above average stocks via an intelligent strategy is sine qua non. What makes a good strategy, however, is more tenuous because of the market's dynamics of self-correction. That is, intelligent strategy consistently outperforms unintelligent strategy over time, but it is difficult to identify intelligent strategy. What if, however, the group of individuals using underperforming investment strategy remained the same over time?

Spinning off this idea that the savvy of a group of investors might be revealed through their writing, this project establishes that natural language processing can be used at the highest level to glean insights about market movements through news headlines. Specifically, the algorithm takes a set of daily news headlines as input to output a predicted market movement in the Dow Jones Industrial index.

Although this application was more of a proof of concept exercise for the NLP methods used, as an exercise in translating written expression to financial activity, this macro level modeling shares much of the structure of the aforementioned problem, and if successful, would lend itself to a wide range of problems in which qualitative data might be useful in quickly and broadly enhancing financial decision making.

## 2 Related work

Much of the current work applying NLP to stock predictions uses financial reports (there is a patent filed as early as 2001 for related methods [1]) which contain information that is more patently connected to existing commonly used financial metrics. As an example, Microsoft [2] recently used Convolutional Neural Networks (CNNs) on public financial statements like the 10-K to predict future performance for specific companies. Another paper [3] finds that using text data from the 8-K significantly improves predictions over financial metrics alone. As the dataset I analyze here

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

was taken from Kaggle (description to follow) others have used Deep Learning to analyze the same variables [4], to similar results, but with different methods (e.g. n-gram and bag of words modeling). On a more financially targeted dataset or looking at specific companies [5], [6], some ML approaches do remarkably well, but the less targeted nature of my approach (i.e. looking at a diverse set of world news headlines) might suggest NLP may have value in accumulating insight in some of the least articulable contexts.

## 3 Dataset and Features

The data, "Daily News for Stock Market Prediction" was sourced from Kaggle [7] and includes several years of daily movements in the DJIA, either up or down, each accompanied by 25 of the top worldwide news headlines (single sentences) taken from the Reddit WorldNews Channel. The dates themselves were not used, but rather the model was trained exclusively on the headline sentences, with the movement direction of the DJIA as the outcome variable. Of the 2000 days of available headlines (= approximately 50,000 headlines), 70 percent were used for training, with the remainder split evenly between dev and test sets, while all the data was shuffled first.

Preprocessing the data involved first stripping leading white space and decoding some of the sentences that were in bytes literal format. As the headlines come from many sources, there was little consistency in their grammar and punctuation, let alone the face-value relevance to market behavior (e.g. "Russia: U.S. Poland Missile Deal Won't Go 'Unpunished'" vs. 'Swedish wrestler Ara Abrahamian throws away medal in Olympic hissy fit '). I used NLKT's [8] PUNKT Word Tokenizer to split the headlines into words. Examining the data manually, the tokenizer generally dealt well with punctuation, with one exception being leading single quotes, for which the dataset was adapted.

## 4 Methods

Using Pytorch [9], the architecture chosen was a many-to-one LSTM model, with a hidden layer size of 50, initially with one sentence corresponding to one prediction, and later, for improved accuracy, the full set of 25 daily headlines as input to one market prediction. Each headline was prepended with a unique word denoting the headline ranking, in case such information might be useful to the neural network, and appended with an end-of-sentence word. The loss function was a simple cross-entropy loss performed on the log softmax of the output prediction, and as there were only two predictions (up or down), this just meant the log softmax likelihood of the movement that actually occured.

## 5 Experiments/Results/Discussion

Running on an AWS EC2 instance, the dataset was not so large that mini-batch size, number of epochs or the learning rate were important factors to choose carefully. In the end, a mini-batch size of 5 days was used, with a learning rate of 1e-4 and typically around 20 epochs. Overfitting, on the other hand, was a major issue from the start, and much experimentation went into choosing the appropriate regularization before a weight decay of 0.001 was finally reached (in the process, the learning rate was also reduced). Accuracy was simply calculated as the percentage of days where the prediction was correct; given the somewhat limited size of the dataset there was a fair bit of fluctuation in accuracy, although the cross entropy loss followed changes in accuracy fairly closely. In the final model for which both training and validation loss plateaued, the prediction accuracy was 57 percent, as opposed to 50 percent via random guessing, 53 percent via always choosing "up" over this time period, and 55 percent utilizing only a single headline.

## 6 Conclusion/Future Work

While the success of the model over a random guessing or always-choose-up method might appear to be marginal, the facts that the stakes are so high and that the inputs were not actually selected from financial documents bolster the achievement of any improvement over baseline. The LSTM architecture saw definite improvement when given a longer list of headlines to work with, showing potential for analyzing a broader set of text. One of the key lessons was the importance of regularization in this context.

Given more time, this approach could be expanded to a broader and more selectively chosen data set, iterating until the most valuable data sources are found. Additionally, combining other currently used financial metrics could harness the power of natural language processing as a supplement, rather than replacement of traditional quantitative data.

## References


[1] Herz, F. S., Ungar, L. H., Eisner, J. M., & Labys, W. P. (2012). U.S. Patent No. 8,285,619. Washington, DC: U.S. Patent and Trademark Office.

[2] Ryan, Patty (2017). Stock Market Predictions with Natural Language Deep Learning. https://www.microsoft.com/developerblog/2017/12/04/predicting-stock-performance-deep-learning/

[3] Lee, H., Surdeanu, M., MacCartney, B., & Jurafsky, D. (2014). On the Importance of Text Analysis for Stock Price Prediction. https://nlp.stanford.edu/pubs/lrec2014-stock.pdf

[4] https://www.kaggle.com/jackcai/omg-nlp-with-the-djia-and-reddit

[5] Joshi, K., Bharathi, H.N., & Rao, J. (2016). Stock Trend Prediction Using News Sentiment Analysis. https://arxiv.org/pdf/1607.01958.pdf

[6] Lee, K., & Timmons, R. (2007). Predicting the Stock Market with News Articles. https://nlp.stanford.edu/courses/cs224n/2007/fp/timmonsr-kylee84.pdf

[7] https://www.kaggle.com/aaron7sun/stocknews

[8] Bird, Steven, Edward Loper and Ewan Klein (2009). Natural Language Processing with Python. O'Reilly Media Inc.

[9] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in PyTorch.