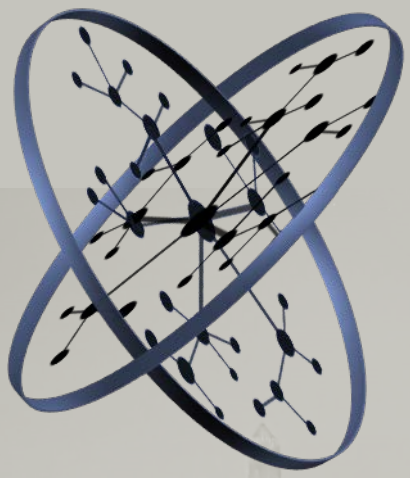
Algo Depth

Financial News to Predict Stock Market

By Zixuan Zhang
Machine Learning
&
Hai'yi Mao
Machine Learning

July 2016



Financial News to Predict the Stock Market

Financial news provides information to the general public. Consumers rely on information they read or hear before buying a product. The internet makes content easily accessible and more relevant than ever.

Remember the game where you guess how many jellybeans are in the jar? Your guess, and all your friend's guesses probably spanned across a wide range. Wisdom of crowds states that the average of a mass populations' guesses will be closer to reality than an individual expert guess. This phenomenon holds true for predicting jelly beans, or sentiment around a stock.

Consumer sentiment plays an important role in financial markets. We apply natural language processing to analyze text and understand what consumers are reading or talking about. We prove that using financial articles to predict stock movements outperforms a random guess investment strategy.

We test three methods in our analysis. First, we use deep learning to test article and sentence level prediction. Next we create features from characters, words, and sentences in each data source. Finally, we create sentiment scores from each article to then predict stock returns.

Our research is broken up as follow:

1.) Data explained.
2.) Method 1: Computer vision applied to sentiment analysis
3.) Method 2: Decomposing each article into features for stock return prediction.
    - 2.1: Feature construction
    - 2.2: Hyperparameter Optimization
    - 2.3: Model training
    - 2.4: Experiment Results
4.) Method 3: Using our features to predict sentiment of each article.
5.) Future work.

Data

We collect articles from ten unique financial data sources, as listed in Table 1. In this research we focus exclusively on Seeking Alpha articles.

Table 1. Data sources we collected

Financial News
SeekingAlpha
Insider Monkey
Reuters
NasDaq
Motley Fool
Business Insider
Market Watch
Guru Focus
Zacks
The Street

We filter Seeking Alpha to find news related to companies in the S&P 500 from 2010-present. Our experiment contains over a million relevant articles for analysis.

Format Data

From each article we extract all text from the URL, the data of the publication, and the primary stock ticker the article relates to.

Returns

We use today and prior days' information to predict future stock returns. We calculate returns as the change in stock price from period (P_n) to current period (P).

$$Return = \frac{(P_n) - (P)}{P}$$

P = Stock Price
n = Time Period

Returns are then classified as either positive or negative, rather than their percent return. In this experiment we predict the direction of stock movement rather than the percent return.

Article Collection

News articles are released at all times during the day, but financial markets in the US are only open from 9:30am Eastern Time to 4:00pm EST. We begin the data collection process at 3:00pm EST to capture the past 24 hours of news releases. This gives enough time to organize our data, run our models, output predictions, and tailor a portfolio strategy before markets close.

Non-Trading Days

On the day after a non-trading day, we fall into the unique scenario that we have more than 24 hours of unused data. In order to incorporate all meaningful news releases into our predictions, we organize all news releases from the last trading day at 3:00pm and use them to predict next trading day stock returns.

Model Testing

Method 1: Deep Learning Algorithm

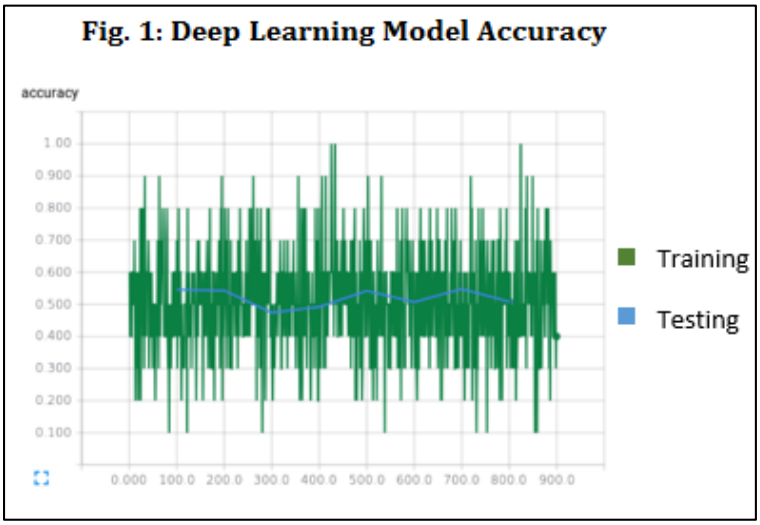
In our first experiment we use a neural network to test for predictive power between article level information and next day stock returns.

We select a convolutional neural network (CNN), a traditional image recognition algorithm to identify article level predictive power. A CNN takes data, i.e., text from each news feed, and converts the data into pixels. Each character, word, and sentence is input into the CNN; patterns are identified by position of one data point (pixel) relative to the others, and then to the stock return. CNN are complex deep learning models that do not tell us why a pattern exists, only what the pattern is.

We use TensorFlow to deploy our computation (an open source machine intelligence software initially created by Google Brain Team). 2010-2014 is selected as our training set, and 2015 is testing set.

In deploying our model, we aim to predict tomorrow's stock price return. During the training process, we experience training accuracy of 52.0% and testing accuracy of 50.1%, as seen in Figure 1. Our model is unstable, and does not converge during the training process. This is caused by limited graphical processing computing power and hyperparameter tuning.

Fig. 1: Deep Learning Model Accuracy



Applying our deep learning model to SeekingAlpha on an article-level does not have predictive power significantly better than a random coin flip. In the future after we increase our GPU computing power we will tune the hyperparameters and test this model further.

Method 2: Features to Stock Return

In this experiment, we analyze the sentiment in news by recognizing the emotional state expressed in an article. We construct different types of features.

1. Features

We start our analysis by constructing natural language processing features:

Table 2. Features used in our experiment

Name of indicators	Description
Ngram Word	Sequence of n words.
Ngram Char	Sequence of n characters.
Part of Speech	Smallest element for words to have distinctive meaning.
NER (ngram)	Classifies entities into pre-defined categories.
NER (stack)	Classifies entities into pre-defined categories.
Punctuation	Marks used in writing to separate sentences and clarify their meaning.
Dictionaries	Sentiment word lists.
NER (count)	Classifies entities into pre-defined categories.
Document Level	Features applied to entire article.
A to B	Relationship between two numerical values.

Ngram

A continuous sequence of words or characters in a sentence. Metrics on syllables, letters, or words are collected and analyzed to make predictions. In our analysis we focus on Ngram words and characters. These features are used to assess the probability of word sequences on article sentiment.

Example: Google **revenue increased** during their quarterly earnings announcements.

An article with "revenue increased" and "quarterly earnings" as its most frequent word may exhibit a certain pattern in sentiment.

Part of Speech

The distinctive meaning of a word. There are 8 major parts of speech in English grammar: noun, pronoun, verb, adverb, adjective, conjunction, preposition, and interjection.

> Noun: A name, person, place, or event.
> Example: **Bill Achman** is a hedge fund manager.

Preposition: The locator of place or time.
Example: Wall Street is **in** New York.

Dictionaries
Different word corpuses and their associations with eight basic emotions and sentiments.

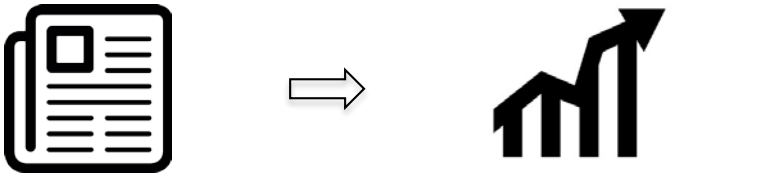
Dictionaries
Sentiment 140 Lexicon- Yelp Reviews
Sentiment 140 AfflExNegLex
NRC Hashtag Sentiment Lexicon
NRC Hashtag Emotion Lexicon
NRC Emotion Lexicon
NRC Colour Lexicon
Hashtag Sentiment AffLexNegLex
Amazon Laptop Electronic Reviews

Named Entity Recognition
Locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, monetary values, or percentages.

A to B
The action between two quantitative variables separated by the word "to".

Example: Revenue increased from **20% to 30%** last quarter.

Experiment Labels



Article Features Stock Return Classification

Each article's features are input into our machine learning models, labeled against next period stock returns. The returns are classified as positive or negative. This method assumes that our features account for all future stock returns, a process we expand upon in Method 3.

2. Hyper-parameter Optimization

We apply grid search during our validation set to identify paramters that maximize our returns. The focus in on optimizing parameters of Supper Vector Machine model, and applying these across all predictions.

There are four parameters we need to tune in SVM: kernel ("poly", "rbf", "linear"), C (regularization parameter, [1, 10, 100, 1000]), γ (gamma in kernel function, [1, 2, 3, 4, 5]), d (degree of kernel function, [1, 2, 3]).

To choose a good parameter set, we conducted a 5-fold cross validation on the training set using all the parameter combinations, and obtained the best parameters. Then, we tested the holdout set by using the model with best parameters.

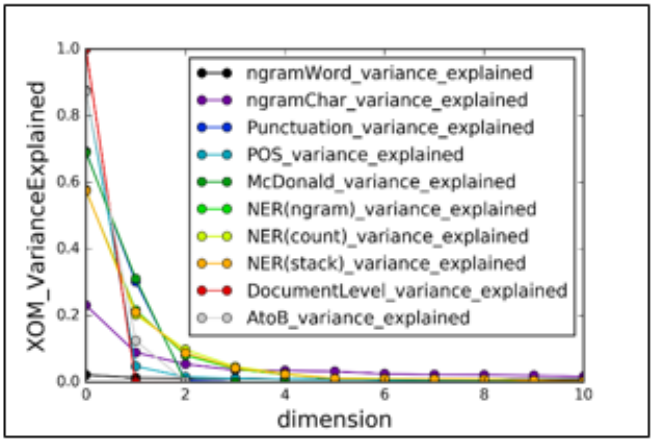
SVM
Penalty: rbf
Set C = [2**i for i in xrange(-8, 21, 4)]
Which is [0.00390625, 0.0625, 1, 16, 256, 4096, 65536, 1048576]

Random Forest
RandomForestClassifier(n_estimators=55,oob_score=True,min_samples_split=500,max_features='sqrt', n_jobs = 128, max_depth = 16)

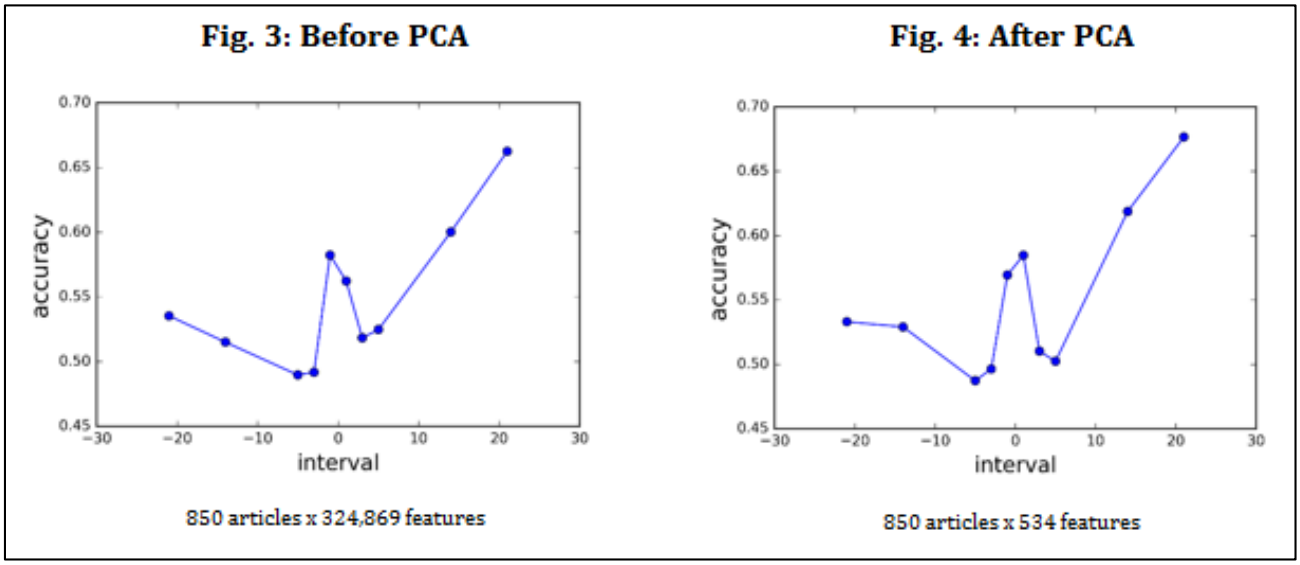
Feature Dimensions
Each ticker tested on Seeking Alpha has 1,000 historical news articles on average. Each article has thousands of features, causing a risk of over-fitting in our model.

We apply Principal Component Analysis (PCA) to reduce the dimensions of XOM in Figure 2.

Fig. 2: Principal Component Analysis Dimension Reduction



Using 10 dimensions results in 99.9% of all variance being explained. This means we can reduce our dimensions significantly without losing much useful information. We apply this dimension reduction on Apple stock price to test prediction accuracy before and after PCA in Figures 3 and 4. It is important to note that Figures 3 and 4 show results of validation period rather than test period. In these charts we are not concerned with percent accuracy, but instead the change from Figure 3 to Figure 4 after dimension reduction.



Next day accuracy increases from 56% to 58% after reducing our features from 324,869 to 534.

3. Model training and prediction

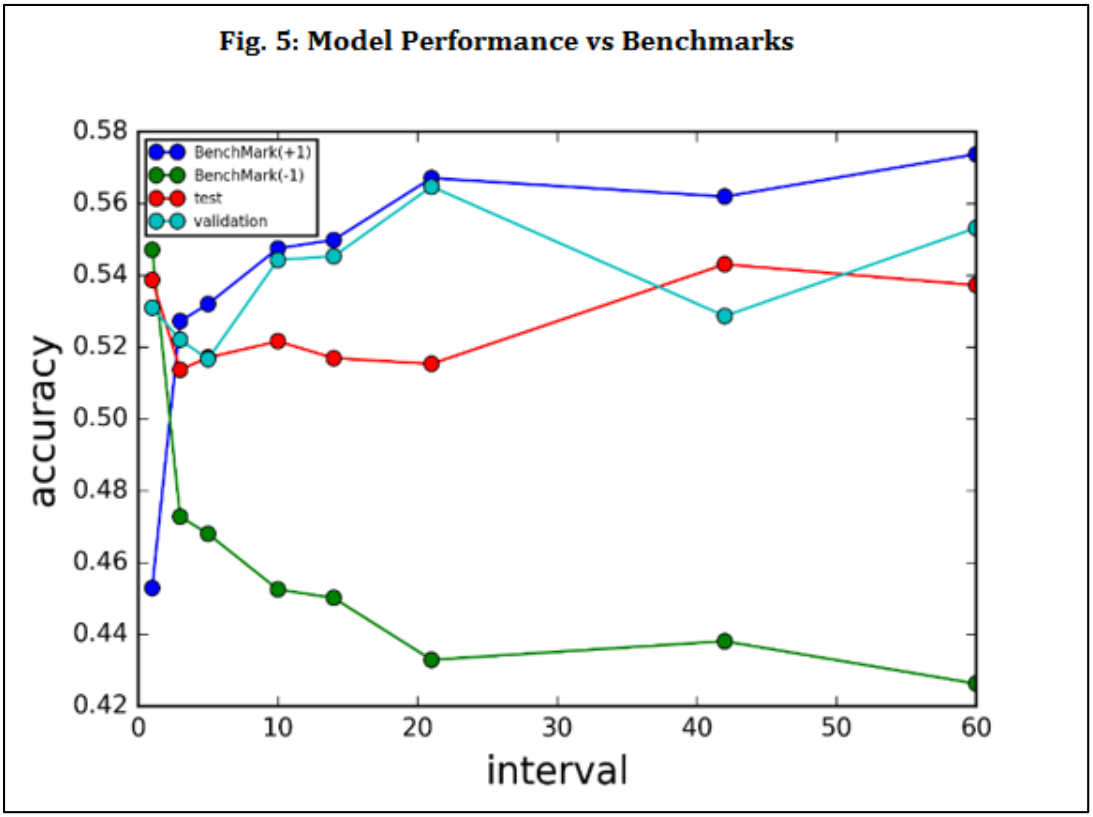
After obtaining the best parameters, we re-trained the model using new training sets. Specifically, we used data from 2012-2014 (60%) for training and 2015-2016 (40%) for testing.

4. Experiment Results

We tested linearSVC, Random Forest, and Gradient Boost models to predict future stock returns on 20 tickers in the S&P 500. We compared the results from our testing period to that of our training (validation) period, a buy-and-hold strategy, a short-only strategy, and a random guess strategy.

Validation: Look-ahead period used to optimize model hyper-parameters.
Buy-and-hold (benchmark +1): A strategy of buying a stock and holding it permanently.
Short-only (benchmark -1): A strategy of selling a stock and holding it permanently.
Random Guess strategy: Assumes a 50% accuracy rate.

Figure 5. shows the results of our linearSVC model. Next day prediction accuracy is 53.9%.



Next Steps

In an attempt to improve predictive power, there are a few tests we will conduct:

1.) End to intermediate step, labeling each article with sentiment score rather than stock return. This process is described in method 3.
2.) Relevance score to weight how relevant an article is to the company being mentioned. Articles that are more relevant to a company will have more power in prediction.
3.) Event driven data targeting key financial words and their impact on stock returns. By identifying corporate financial events, ie: buy-back announcements, dividend increases, or acquisitions, we can identify major events that will drive stock returns.

Method 3: Sentiment Label to Stock Return

In this method, we add an intermediate step to label each article with a sentiment score.

To train our model with existing features, we need a database of text with sentiment scores. There are a few options we can use:

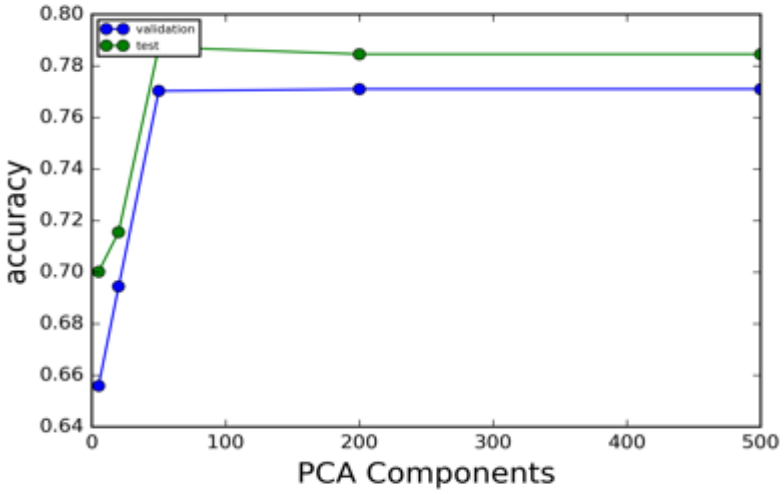
1) Movie reviews from Rotten Tomato website, with sentiment scores.
2) Collect subjective human sentiment scores with Amazon Mechanical Turks.
3) CrowdFlower Economic News Article Tone and Relevance
4) IBM Watson Alchemy sentiment score.

Article Features → Pos. or Neg. Sentiment → Stock Return

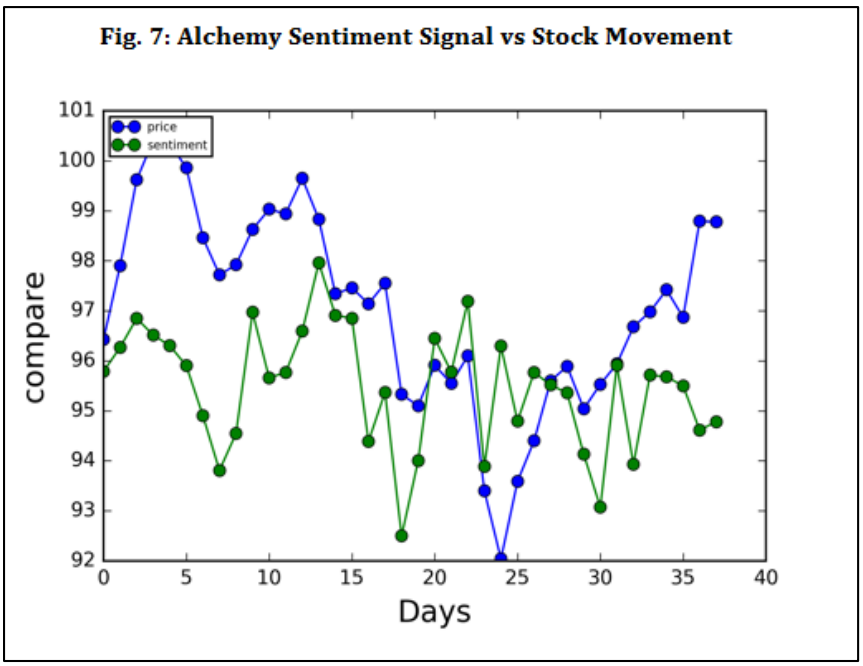
In this experiment, we tested IBM Watson's Alchemy API to create sentiment scores. Alchemy holds a strong reputation in the natural language processing community regarding its sentiment scoring system, accessing hundreds of different databases to provide various metrics.

We use Alchemy's sentiment scoring system to train our set of features to then predict the sentiment of each article. We have strong testing accuracy in our binary prediction problem, as seen in Figure 6.

Fig. 6: Predicting Sentiment Score



Each article published by Seeking Alpha is provided a positive or negative sentiment. Our testing accuracy of ~79% is strong, showing our features are successful in identifying the sentiment of each article. We now much test for the signal strength of Alchemy sentiment to stock movement.



Apply Sentiment Score to Stock Returns

Finally we arrive at reliable sentiment scores using Alchemy. We plot the daily sentiment scores derived from Alchemy vs Apple company stock returns over a 53 day sample period in Figure 6.

Fig. 7: Alchemy Sentiment Signal vs Stock Movement

Our next steps for research

1.) We will add additional tickers and run statistical processes to test Alchemy signal strength. Should no significant pattern exist, we will test Amazon Mechanic Turks and Crowdflower for signal strength.
2.) We will apply our framework to each other datasource to find comparable results.
3.) Develop a relevancy model to create weights for articles used in prediction.
4.) Apply natural language processing features to identify sentiment around corporate events.

Important Disclaimer and Disclosure Information

Algo Depth makes no representations of any kind regarding this report. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the absence of errors, whether or not known or discoverable. In no event shall the author(s), Algo Depth or any of its officers, employees, or representatives, be liable to you on any legal theory (including, without limitation, negligence) or otherwise for any claims, losses, costs or damages of any kind, including direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages, arising out of the use of the report, including the information contained herein.

This report is prepared for informational and educational purposes only, and is not an offer to sell or the solicitation of an offer to buy any securities. The recipient is reminded that an investment in any security is subject to many risks, including the complete loss of capital, and other risks that this report does not contain. As always, past performance is no indication of future results. This report does not constitute any form of invitation or inducement by Algo Depth to engage in investment activity.

Algo Depth has not independently verified the information provided by the author(s) and provides no assurance to its accuracy, reliability, suitability, or completeness. Algo Depth may have opinions that materially differ from those discussed, and may have significant financial interest in the positions mentioned in the report.

This report may contain certain projections and statements regarding anticipated future performance of securities. These statements are subject to significant uncertainties that are not in our control and are subject to change.

Algo Depth makes no representations, express or implied, regarding the accuracy or completeness of this information, and the recipient accepts all risks in relying on this report for any purpose whatsoever. This report shall remain the property of Algo Depth and Algo Depth reserves the right to require the return of this