Brill-based Stock Message Classifier:

A Framework for classifying News Stories
and Internet Message Board Posts
to predict Stock Price movements

MIT 6.863J/9.611J Natural Language Processing
Massachusetts Institute of Technology
Prof. Robert C. Berwick
Armando Valdes

ABSTRACT

Qualitative verbal information contained in news stories and earning announcements that are available to investors through newswires should, in theory, be incorporated into stock prices. However, the higher costs involved in processing this information introduce frictions that delay its immediate incorporation into stock prices. In this paper, I propose a method for training the Brill Part-of-speech (POS) tagger to assist in the classification of news stories and/or Internet messaging board posts into three possible types: positive, neutral, and negative. I will also test the hypothesis of whether adding information to the adjective tag to identify the adjective as positive or negative can help in predicting the direction of stock price movements.
# Table of Contents

Table of Contents..........................................................................................................2

Acknowledgement........................................................................................................4

1 Introduction and Motivation .........................................................................................5

1.1 The Brill Part of Speech Tagger ................................................................................ 6

2 Previous Work: Analyzing Qualitative Information.........................................................7

3 Selection of Testing and Training Data ........................................................................10

3.1 Stock Selection ........................................................................................................... 11

3.2 Raging Bull ................................................................................................................ 12

3.2.1 Cleaning up the data .............................................................................................. 13

4 Correlating Message Board Posts to Stock Price Movements ..................................14

5 Training the Brill Tagger ..............................................................................................15

5.1 Creating ‘gold standard’ training and testing files ...................................................... 15

5.1.1 Qualifying Adjectives ............................................................................................ 16

5.2 Creating the Brill Tagger lexicon file ......................................................................... 17

5.3 Learning Contextual Rules ......................................................................................... 18

6 Classifying Posts with the Brill Tagger .......................................................................19

6.1 Results ....................................................................................................................... 21

7 Conclusion and Future Work ......................................................................................23

7.1 Adding Support to other Parts of Speech ................................................................. 23
7.2 Working with Probabilities ................................................................. 24
7.3 Independence Assumptions .............................................................. 24
7.4 Volatility .............................................................................................. 25
7.5 Post to Price Correlation ................................................................. 25

8 Works Cited .......................................................................................... 26
Acknowledgement

I would like to express my deepest gratitude to Prof. Robert C. Berwick for his help and support during this project. I would also like to thank Gabriel Zaccak and Yuan Shen for their input and feedback during the course of the class, which helped me understand some of the concepts applied in this project.
1 Introduction and Motivation

On September 9, 2008, the shares of United Airlines’ parent company, UAL Corp., quickly fell to $3 from their previous level of $12.50. In just 13 minutes, investors reading a report that United had filed for bankruptcy, wiped out $1 billion of United’s market value. The investors were largely unaware, however, that the report that they were reading was a six-year-old report that was erroneously flagged by Google’s news aggregation service as breaking-news [1]. Once the stock sell-off began, the problem was exacerbated by the growing adoption in Wall Street of algorithmic trading programs that buy or sell stocks based in part on news headlines and earnings data [2].

Although, in the end, United’s bankruptcy story turned out to be factually incorrect, and the stock price movement unwarranted, the case highlights the importance of qualitative data in asset pricing. Yet, this case is the exception rather than the norm. Soft information such as news stories or the language used in earnings announcements is often not immediately incorporated in security prices [3]. As a result, future returns could be predicted based on current publicly available information, such as news stories or posts in Internet message boards like Yahoo Finance or Raging Bull.
Finding a way of efficiently classifying qualitative information is compelling because it could potentially give investors the ability to predict movements in stock prices before they occur. From a Natural Language Processing point of view, analyzing financial messages is a good way of testing different NLP techniques given that the content of these messages can be expected to be similar to that of the Wall Street Journal. Because the WSJ is used for training many of these taggers and classifiers, a lot of bootstrapping work is already partially done for us.

Although some authors have started to concentrate on the importance of qualitative, or soft, information, for the most part they have focused on simple document classification based on ‘positive’ or ‘negative’ words found in the document [3]. In this paper, I propose a method for training the Brill tagger so that adjectives can be tagged as either positive or negative, based on information learned by the tagger during the training phase. This information can then be used by a classifier to determine if a message predicts a positive or a negative movement in the stock price.

1.1 The Brill Part of Speech Tagger

The Brill Part of Speech Tagger is an error-driven transformation-based tagger that begins by assigning words their most likely tag according to the training corpus, or a naïve tag (e.g. ‘noun’) if the word is not found in the corpus. Once the input text is initially tagged, an ordered list of transformations is applied to each word, based on the
word itself and the words immediately surrounding that word. These transformations account for a large increase in accuracy in the tagger even when presented with unknown words [4].

The Brill Tagger needs to go through an initial learning phase to build rules to predict the most likely tag for unknown words (e.g. ‘ing’-terminated words could be gerund verbs). Most importantly, training is needed for the tagger to build the list of ordered transformation rules that will be applied to correct errors in an initially tagged corpus. In this training step, the contextual learner compares an initially tagged version of the corpus against a ‘golden’ corpus (which could have been manually tagged). Whenever the learner finds a discrepancy between the corpora, it tries to determine which context-based transformation could be applied to the incorrectly tagged corpus to eliminate the error. For example, one such transformation could be “if the previous word is to, then the change the current tag from noun to verb.”

2 Previous Work: Analyzing Qualitative Information

In [3], Joseph Engelberg analyzes the impact that qualitative language in post earnings announcements has on stock price fluctuations. Although earnings announcements usually contain both quantitative or hard information and qualitative or soft information, he controls for the presence of quantitative information by taking forecasts of the firm’s earnings into account.
In order to determine the impact of qualitative information, Engelberg uses typed dependency parsing, a Natural Language Processing technique where negative words are paired with other words that belong to several different categories. The goal is to identify specific kinds of qualitative information that is most difficult to process. In order to arrive at this classification, Engelberg uses a program called the General Inquirer to classify words in the announcements. He then focuses on the negative category to identify news stories that convey negative information about a firm’s fundamentals, and assigns a value to each story that corresponds to the fraction of negative words with respect to the total number of words in the article.

By exploring a large cross section of firms from data obtained from the Dow Jones News Service (DJNS) available through Factiva, Engelberg is able to show that the content of financial media can predict asset returns in the medium term. His findings indicate that qualitative information about future performance has a greater ability to predict future returns.

Engelberg’s findings build upon the research by Antweiler and Frank. In [5], Antweiler and Frank use machine learning algorithms to analyze qualitative information in messages retrieved from Internet message boards such as Yahoo! Finance and Raging Bull. Messages retrieved from these boards is parsed into a standard format, keeping the user name, the date, the post title, and the contents of the message, as well as the
board from which the information was retrieved (YF for Yahoo! Finance and RB for Raging Bull), as shown in Figure 1 below.

![Sample Yahoo! Finance message board post from Antweiler and Frank’s study.](image)

In their study, Antweiler and Frank use a naïve Bayes algorithm to classify Internet message board posts in different categories. Assuming that the occurrences of words in a document (or message board post) is independent of each other, the Bayesian algorithm builds a model of conditional probabilities based on streams of words appearing on the document, and assigns probabilities to ‘buy’, ‘sell’, or ‘hold’ signals for each message. Based on the highest probability of each of the buckets, their algorithm determines the ‘bullishness’ of the news story, and further aggregate multiple stories to determine the direction of possible fluctuations in the stock’s price.

Antweiler and Frank use the ‘bullishness’ index when sequencing Internet message board posts of a particular stock to financial market changes. In particular, they try to find instances in which ‘buy’, ‘sell’ or ‘hold’ information in Internet message board
posts translate into positive, negative, or flat price fluctuations in the financial markets. Using this model, they fail to find statistically significant evidence that the posts are good predictors of stock market returns. However, they do observe that there is some effect of more messages predicting negative next-day stock returns.

Engelberg notes that one of the main differences between Natural Language Processing (NLP) methods like the one he used in his study and the Naïve Bayes algorithm used by Antweiler and Frank is that the latter does not attempt to infer the meaning of the text. In contrast, NLP methods aim to identify parts of speech, sentence structure, and grammar, which could lead to more accurate classification. Thus, the selection of the method should not be taken lightly, since it could play an important role in the results from the study.

### 3 Selection of Testing and Training Data

In order to assess the impact of stock-related messages on stock prices, I needed to get a considerable amount of messages over an extended period of time, as well as a list of prices corresponding to the same period. I decided to use one of the boards Antweiler and Frank used in their study (Raging Bull), given its free access and lack of restrictions in scraping data from their website. Historical stock prices were sourced from the Center for Research in Security Prices database maintained by Wharton Business School.
3.1 Stock Selection

I decided to select a group of 12 technology stocks which are currently trading towards the top of the NASDAQ in order to obtain the largest number of posts possible. These companies are: Apple, Amazon, Google, Intel, Research In Motion (RIM), Microsoft, Cisco, Qualcomm, Oracle, Dell, Sun Microsystems, and Nvidia. A total of 14778 posts were retrieved between the period of January 20, 2003 and April 24, 2009 (the time when the data was compiled). On average, each board received 1,231 posts during the period, with younger companies such as Research In Motion receiving the least number of posts (650) and older companies receiving the largest amount of posts (1680). In general, the data was well distributed among companies, as shown in Figure 2 below.

![Table]

<table>
<thead>
<tr>
<th>board</th>
<th>count(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAPL</td>
<td>1506</td>
</tr>
<tr>
<td>AMZN</td>
<td>1337</td>
</tr>
<tr>
<td>CSCO</td>
<td>1680</td>
</tr>
<tr>
<td>DELL</td>
<td>783</td>
</tr>
<tr>
<td>GOOG</td>
<td>1380</td>
</tr>
<tr>
<td>INTC</td>
<td>1057</td>
</tr>
<tr>
<td>JAVA</td>
<td>1502</td>
</tr>
<tr>
<td>MSFT</td>
<td>1457</td>
</tr>
<tr>
<td>NVDA</td>
<td>1523</td>
</tr>
<tr>
<td>ORCL</td>
<td>1216</td>
</tr>
<tr>
<td>QCOM</td>
<td>687</td>
</tr>
<tr>
<td>RIMM</td>
<td>650</td>
</tr>
</tbody>
</table>

Figure 2 Breakdown of number of posts by board

Daily historical prices from all these stocks were readily available in the CRSP database. I only performed some simple data processing on top of that data to
determine the net change in price between the opening and closing prices, as well as the direction of the price movement. Then, I proceeded to import all the data into a MySQL table for easier correlation between the stock prices and the corresponding message board posts.

3.2 Raging Bull

Raging Bull is a full-service financial message board and market screener that offers free access to its users. Message boards at Raging Bull are organized by company and presented in chronological order. Hence, it was easy to scrape information off the screen and store it in a MySQL database for analysis.

Messages were scraped of the Raging Bull site with a Java program using the Hibernate object-relational mapping library to persist objects in a MySQL table. As part of the process, I captured 5 attributes from each post: the username of the person who posted the message, the contents of the message, the board’s company ticker (e.g. AAPL for Apple), the date of the post (in yyyyMMdd format), and the post’s rating. Although I
ended up not using the post rating as part of the process, I thought it could be useful in filtering out SPAM or in assigning a larger weight to attributes of high rating posts.

3.2.1 Cleaning up the data

While I selected this particular board over others due to this ease of accessibility, its selection also posed some problems in compiling the data. Lacking an adequate amount of moderators, the board suffers from persistent SPAM, political rants, and irrelevant musings from the boards’ readers.

In an effort to eliminate as much noise as possible from these irrelevant posts, I started by looking at the posts from users who had a significant amount of posts in a single day, but that all of the posts were identical. Immediately, a pattern emerged in which users seemed to be copying/pasting irrelevant text or political propaganda from one board to the next. Particularly common were posts questioning President Obama’s citizenship and general comments about the election in the Sept. - Oct. timeframe.

A second major source of SPAM was users advertising other boards or other companies in their posts. Most of these posts were followed by a link to the advertised web site. Given that eliminating all posts that had a link in them still left me with a large amount of posts and data to work with, I decided to eliminate all those entries
4 Correlating Message Board Posts to Stock Price Movements

While there are multiple sophisticated ways of correlating news events or message board posts to stock price movements, as pointed out by Engelberg in [3], I decided to use the simple approach of creating a view over the tables containing the board posts and their corresponding stock prices:

```sql
CREATE VIEW correlation
AS select post.board, post.post_date, post.contents,
    price.prc, price.day_chng, price.pct_chng,
    price.direction
FROM posts post, prices price
WHERE post.board = price.ticker
AND post.post_date = price.date;
```

*Figure 4 MySQL statement to create a view over the board posts and their corresponding stock prices.*

Creating the view results in an automatic correlation between the board posts and their corresponding stock prices by matching: 1) the stock ticker (e.g. AAPL for Apple) and 2) the date the message was posted to the board with the date in which the price was captured. Finally, I added some more fields to the `correlation` view from the `price` table, to enable access to the daily net change in price (computed as the difference between opening and close prices), and the direction of the price movement (positive, neutral/flat, or negative).

Given the stock's price movement through the day, I assume that the contents of the post will be negative if the price movement is negative; and will be positive, if the price movement is positive. This assumption is important because it may be that the contents of the post depend on the price movement of a previous date or that the
information that was not yet incorporated in the price by the end of the day. A further refinement of this methodology will need to take this variable into account, as shown by Engelberg.

5 Training the Brill Tagger

Before classifying posts with the Brill Tagger, the tagger has to go through a series of learning steps so that it can more accurately assign tags to unknown words. Most importantly, it needs to create the list of ordered transformation rules that will be applied to the initial tags. In this case, training the Brill Tagger involved the following steps: creating ‘gold standard’ training and testing files, creating a lexicon from part of the training file for contextual learning, and learning contextual rules from a different part of the training file.

5.1 Creating ‘gold standard’ training and testing files

Creating the training and testing files was one of the most critical parts of training the Brill tagger, because I wanted the output of the tagger to give additional information about the kind of adjectives contained in the text. The hypothesis is that message board posts leading to negative movements in the stock price have different adjectives, verbs, and adverbs than message board posts leading to positive movements in the price. I start with adjectives as a way of illustrating the potential benefits of the method, but it should be easy to apply the same paradigms to other parts of speech.
5.1.1 Qualifying Adjectives

The first step in adding a measure of ‘positivity’ or ‘negativity’ to an adjective is identifying the word as an adjective. I approached the problem by first dividing the message board posts in two categories: positive and negative, depending on the direction of the corresponding stock’s price movement on the date of the post:

```
$ echo "select contents from correlation where post_date < 20081001 and direction = 'POSITIVE'" | /usr/bin/mysql -u avaldes -p mit6863 > positive.training
```

Please note that I am only including posts prior to Oct. 1, 2008, to reserve three months worth of data for testing. Unfortunately, the CRSP database does not include prices after Jan. 1, 2009, so I could not use the most recent 5 months of posts for testing the classifier.

Once I created the positive and negative training files, I ran an unmodified version of the Brill tagger, using the WSJ corpus lexicon, lexical, and contextual rule file:

```
tagger LEXICON.WSJ.Z positive.training BIGRAMS LEXICALRULEFILE.WSJ CONTEXTUALRULEFILE.WSJ > positive.training.tagged
```

The assumption is that the words used in the Raging Bull forum are similar enough to the Wall Street Journal, that the Brill Tagger will do a pretty good job in creating a version that I could use the WSJ files to bootstrap the Brill tagger training for the Raging Bull data.

The tagged versions of the positive and negative training files are then modified to include information about the ‘positivity’ or ‘negativity’ of the adjectives contained in the file. To do so, I wrote a simple Java program that counts all different instances of an
adjective occurring in the positive training file, as well as all instances of the same adjective occurring in the negative training file, and then computes the corresponding positivity and negativity values:

\[
\text{positivity} = \frac{\text{posCount}}{\text{posCount} + \text{negCount}}; \\
\text{negativity} = \frac{\text{negCount}}{\text{posCount} + \text{negCount}};
\]

---

**Figure 5** Pseudo-code for enhancing adjective tags with positivity and negativity information

```plaintext
For each adjective in positive and negative training files do:
    posCount := count occurrences of adjective in positive file
    negCount := count occurrences of adjective in negative file

    totalCount := posCount + negCount
    positivity := posCount / totalCount
    negativity := negCount / totalCount

    replace all occurrences of “adjective/JJ” with:
        “adjective/JJ{positivity,negativity}”
Loop
```

The positivity and negativity information is then appended to the adjective tag so that it is incorporated in the Brill tagger lexicon, lexical rules, or contextual rules, as appropriate. For example, the adjective and corresponding tag `positive/JJ` is replaced by `positive/JJ(0.5511811023622047, 0.44881889763779526)` in both the positive and negative training files.

### 5.2 Creating the Brill Tagger lexicon file

After creating the retagged versions of the positive and negative training files, with positivity and negativity information appended to the adjective tag, the next steps come right out of the `README.TRAINING` file included in the Brill Tagger documentation.
The first step is to create two lexicon files, as recommended by the tagger’s documentation: a temporary one for training, and a final one for testing.

To create a temporary lexicon training file, I start by concatenating both positive and negative training texts into one: cat positive.training.retagged \ negative.training.retagged > common.training.retagged, and then using the divide-in-two-rand.prl utility program provided with the Brill Tagger release to divide the corpora in two parts: common.training.retagged-PART1 and common.training.retagged-PART2. I then use the first part of the corpus to create the lexicon training file as follows: cat common.training.retagged-PART1 \ | make-restricted-lexicon.prl > TRAINING.LEXICON. The resulting lexicon file will have entries like similar JJ{0.5384,0.4616} and miserable JJ{0.375,0.625}, but also non-enriched adjectives like assisted JJ, as well as tag information for many other parts of speech.

5.3 Learning Contextual Rules

To train the Brill Tagger with context-based transformation rules, I use the second part of the tagged corpus: common.training.retagged-PART2. As a first step, I strip the tags from the corpus as follows: cat common.training.retagged-PART2 \ | tagged-to-untagged.prl > common.training.untagged-PART2. Then, I run the Brill Tagger against the untagged version, using the training lexicon created in the
previous step:  tagger TRAINING.LEXICON common.training.untagged-PART2 \ 
BIGRAM LEXICALRULEFILE /dev/null -i common.training.init.

The initial Brill-tagged version of the second part of the corpus will most probably present some discrepancies with the tagged version I created in step 1 above. These errors are analyzed by the contextual-rule-learn program in the Brill Tagger package to determine which ordered transformations can be applied to the corpus that result in the less number of errors. Hence, I start the program by passing the ‘golden’ tagged corpus to the program as follows: contextual-rule-learn \ 
common.training.retagged-PART2 common.training.init CONTEXT-RULEFILE \ 
TRAINING.LEXICON. The ordered rule list should then be output in the CONTEXT-RULEFILE. Yet, in my case, the file was empty at the end of the run. Although there were some discrepancies in the files, it seems none of the transformation rules applied resulted in a global reduction in discrepancies between the Brill-tagged corpus and the ‘golden’ standard.

6 Classifying Posts with the Brill Tagger

Once the Brill Tagger has been trained, it can easily be used to tag new posts by using the tagger program. To test this function of the tagger, I first create a testing file from the last three months of 2008: echo "select contents from correlation \ 
where post_date > 20081001 and direction='POSITIVE'" | /usr/bin/mysql \
-u avaldes -p mit6863 > positive.testing. And a similar testing file for negative price movement posts.

The contents dumped from the database are then preprocessed so that they conform to the Penn Tree Bank punctuation format, and so that they respect the ‘one sentence per line’ requirement of the Brill Tagger. As a final preparation step, a special marker “-----------” is appended to the end of each post, to serve as a separation between the posts.

Both testing files are then run through the tagger, using the FINAL.LEXICON and CONTEXT-RULE files created in the training step above, along with the WSJ lexicon rules and bigrams. The result is two tagged files, positive.testing.tagged and negative.testing.tagged. Both of these files will be tagged with the enriched adjective tags designed in section 5.1.1 above (e.g. miserable/JJ{0.375,0.625}).

After the files are tagged, I run them through a simple Java-based classifier that extracts the tag information (e.g. {positive, negative}), for each post and classifies the post as either ‘positive’, ‘negative’, or ‘neutral’ by adding the positive amounts and dividing by the number of tags in the post. If the resulting amount is greater than 0.5, the post is classified as ‘positive’. If the amount is equal to 0.5, it is classified as neutral. Otherwise, the post is classified as ‘negative.’ The pseudo-code for this algorithm is shown in Figure 6 below:
Figure 6  Pseudo-code for classifying message board posts tagged by the modified Brill tagger

6.1 Results

As described above, testing of the classifier was performed against message board posts in the period from Oct. 1, 2008 through Dec. 31, 2008. Although I had collected more post data for the period between Jan. 1, 2009 and April 24, 2009, it was not possible to use this data because I had no access to the corresponding daily historical stock prices. Thus, a total of 79 positive price movement messages and 110 negative price movement messages were used for testing.

Unfortunately, not all posts have adjectives, and not all adjectives are necessarily in the list of 109,722 sentences that were used to train the tagger. Hence, some posts used for testing were useless in predicting stock price movement, even with the transformations applied by the Brill tagger. However, I was still able to use 65 positive and 90 negative posts with ‘positivity’-enriched adjectives, or 82 and 81% of the posts, respectively.
Of the 65 positive posts, the classifier was able to correctly classify 43 (or 66.15%) of them as positive. However, of the 90 negative posts, only 39 (or 43.33%) were classified as negative. Although the positive classification seems more promising, with a $p$-value of $1.67 \times 10^{-5}$, the results are not statistically significant.
7 Conclusion and Future Work

Finding a way of efficiently classifying qualitative information is compelling because it could potentially give investors the ability to predict movements in stock prices before they occur. In this paper, I have shown a method of training the Brill tagger to classify Internet message board posts into three categories: ‘positive’, ‘neutral’, and ‘negative’. This classification information could be used to predict the direction of movement of a stock’s prices.

To illustrate the method, I used enriched adjective tags, in which adjectives were assigned a level of positivity that reflect the tendency of that adjective to appear in a positive price movement post, vis-à-vis appearing in a negative price movement post. Training the Brill tagger in such a way allows it to tag adjectives in new corpora with the enriched tags. By counting the number of enriched tags, and determining the positivity score across all these tags, a downstream classifier can assign a type to the post: ‘positive’, ‘neutral’, or ‘negative’.

Although initial results with enriched adjectives were not statistically significant, multiple enhancements can be incorporated to the method to improve its accuracy:

7.1 Adding Support to other Parts of Speech

Verbs and adverbs seem particularly important in determining the tone of the post. The method presented above can easily be extended to add ‘positivity’ enhanced
tags for verbs and adverbs by counting the number of verbs and adverbs in both positive and negative training files and thus determining the positivity of the word, as explained in section 5.1.1 above.

7.2 Working with Probabilities

One of the problems with the current enhanced tags is that they are not weighed. Thus, a word that appears very often in positive or negative posts is treated equally as a word that appears a single time. Instead of calculating the ‘positivity’ of an adjective (or any other part of speech), it may be fruitful to calculate the probability that such word will appear in a positive or negative post.

This idea can be easily implemented in the code above by dividing the number of occurrences in the positive or negative files by the total number of adjectives in both files. One problem with the approach, however, is that the probability distribution among all adjectives in the text may be very thin, so some smoothing may be required to compensate for this fact.

7.3 Independence Assumptions

An implicit assumption in the design of the training and test data is that each board is independent of each other, so posts in one board do not affect the stock price of a different board. However, all of these companies are in the same sector, and thus should be tightly correlated. Although it may very well be interesting to see how
messages for one company affect other companies in the sector, it may be best to first pick companies in different sectors to evaluate the method.

### 7.4 Volatility

Picking the last three months of 2008 for testing may not have been the best choice, in retrospect. Stock price volatility during that period was unprecedented, and stocks swung widely from gains to losses in short periods without much regard to the company or anything surrounding it. Hence, it may have been very difficult for the classifier to accurately predict the stock price movements in such environment.

### 7.5 Post to Price Correlation

It is hard to determine the price that the post is actually predicting. In the training and tests, the assumption was that the price would react to the post in the same day. Hence, if a negative post appeared on day \( x \), the negative price movement would occur the same day \( x \). However, it may be possible that the price movement had occurred before the post appeared (i.e. on day \( x - n \)), or that it would move in the future (i.e. on day \( x + n \)). A more sophisticated method of correlating posts to price movements when assigning ‘positivity’ scores or probabilities to part of speech tags may result in better training and better prediction power.
8 Works Cited


