News' Effect on Stock Prices
CSC-560 Fall 2009

Jason Anderson jander06@calpoly.edu, **Dominic Camargo** dcamargo@calpoly.edu, **Ben Davini** bdavini@calpoly.edu, and **Brian Oppenheim** boppenhe@calpoly.edu

Computer Science Department
California Polytechnic State University

Abstract. The growth of the media in terms of its expansion across new technological mediums has greatly increased the availability of information. Specifically in the business world, this has translated into greater access to news and detailed reports relating to the world's largest corporations. Such information generally conveys some sort of indication of predicted success or failure of the companies. This means that, more than ever, those trading stocks are reading and likely being influenced by several news sources every day. We believe that this influence can push traders to make stock trades that they might not otherwise have made. Unfortunately correlation and causation are very difficult to prove. Thus, this work is intended to show that there exists an association between the information released by major news sources and the change in stock price immediately following the release of such information. Additionally, we seek to rank news organizations based on the amount of association our model predicts that they have with stock prices.

1 Introduction

This goal of this project is to find an association between news sites and the effect of the performance of securities and/or change in opinions on the value of the securities being analyzed. The project varies from many existing projects that seek to predict the outcome of the stock market based on the vocabulary related to a particular security or current work in neuroeconomics [6, 4] or behavioral finance [9]. Instead, the work addressed in this project seeks to determine whether the reporting of major news outlets relates to stock prices, and if so, how reliable or strong the association is. The metrics of this association are discussed in the following sections.

The technologies utilized in this project are many. We will be using the Java programming language to develop each part of our project. We will be using multiple techniques to acquire data (stock prices and news articles), and also to analyze the data (article rating - i.e., how strongly it speaks for or against the company being analyzed). We will be doing text analysis to obtain the article rating, and exploring two types of text analysis - phrase-based and bayesian - to procure the best results.

The project looks primarily at the day-after influence of a particular news source's article on a stock (e.g., how much does an article written on Wednesday affect the price of a stock on Thursday), but we also explore the delayed effect of chronologically contiguous (e.g., a week's worth) articles.

The below sections offer more detail on the proposed implementation of our project. Our solution (Section 4) details the different parts and algorithms required to accomplish our task; the Data/Datasets show the specific members of the two different pieces of data - articles and stock prices - upon which our analyses rely.

2 Problem Statement

Our goal is to determine if there is an association between news article reports on companies and the performance of the securities (stocks) of those companies in the following days or weeks. [1, 10] If associations are identified, news sources will be ranked according to which news source has the highest average score as determined by our article rating scorer.

While identifying a *correlation* (linear relationship) between news articles and company stock performance would yield the most useful (both in this project and in financial applications) results, it is not feasible to do so within the scope of this project. Instead, we are looking for an *association* - any relationship between two measured quantities that renders them statistically dependent. [8] As an example, we are not looking to show that all articles from the Wall Street Journal about IBM *always* correctly predict the performance of IBM's stocks the next day; instead, we are looking to make a quantitative generalization about the relationship between the *Wall Street Journal* articles and IBM stocks - e.g., 80% of the time, articles from the *Wall Street Journal* about IBM appear to influence the performance of IBM stock the following day. We are not finding a correlation which would be "80% of the time, articles from the *Wall Street Journal* about IBM directly influence the performance of IBM stock the following day."

3 Data/Datasets

Since our project deals with both news articles and the stock market, we are creating and extracting datasets from information freely on the web. The following sections describe these datasets in more detail.

3.1 News Article Dataset[3]

Google news provides an aggregation of multiple news sources in one location. Unfortunately, Google does not offer any API's to programmatically access their information. Web scraping techniques are instead used to extract the news information into one organized database. The dataset is generated based on the following parameters:

– List of companies for which the articles should relate
– Beginning/end date for article publication
– Max number of articles to find

The dataset contains the following fields for a given article:

– article_id : int(11)
– news_source : varchar(1000)
– headline : varchar(1000)
– text : text
– url : varchar(1000)
– publication_date : date
– company : varchar(1000)

3.2 Stock Values[2]

Google Finance provides an easily parsed .csv file to access historical security (stock) data. However, this only accesses one company at a time, and only with a connection to the Internet. We want to be able to access our data without a connection, and also to access only the data desired for the task at hand. For this reason, the stock values are persisted to a local database. The dataset is generated based on the following parameters:

– Stock prices for companies we want to analyze
– Beginning/end dates for stock quotes

The dataset contains the following fields for a given company stock:

– company : varchar(20)
– symbol : varchar(20)
– market : varchar(20)
– date : date
– opening : float
– high : float
– low : float
– close : float

Where *company* is the name of the company whose stock we are analyzing, *symbol* is the stock ticker symbol for the company, *market* is the stock market where the security can be found (NYSE, NASDAQ, etc.), *opening* is the opening price of the stock for the *date*, *high* is the high price of the stock for that day, *low* is the low price of the stock for that day, and *close* is the price at which the stock closed on that date.

4 Our Solution

In order to determine if an association is present, we built a software application that examines the past year's worth of online news articles about a given company and compares that information to actual stock market data. After our comparisons are complete, we then determine which news source is the most accurate stock price predictor. To accomplish these tasks, we have separated our solution into multiple components. Each component works as shown in the following subsections. A complete system diagram is shown in Figure 1.

4.1 Article Extraction

The first step is to collect news articles from various news sources via the Internet. To minimize the different sites required to gather data, Google News Archive[3] is used as an aggregator of articles from different news sources. Given the name of a company to search, Google News is queried and a list of articles is then extracted and inserted into our MySQL database. This greatly reduces the time required in the following steps since all articles are cached locally. Once all the articles have been extracted, they are in the correct form to be rated.

4.2 Article Rating

The next step that articles take after being extracted from a news source is passing through an Article Rater. This process assigns each article an Article Company Rating (abbreviated **ACR**). This score, represented on a scale of -1 to 1, indicates how negatively or positively the article portrays the company that is the subject of the article. Since we believe that the results of the Article Rater will be vitally important to getting good overall results, we designed our application to make this component pluggable. In this way, we were able to write several different algorithms that compute an **ACR** and easily choose between them at runtime. We use this pluggability to compare the performance of our system across different algorithms.

As part of our work, we looked at several methods for assigning an **ACR** to each article. These methods all fall under the subsection of data mining called sentiment mining or sometimes called opinion mining. Techniques considered involved part of speech tagging, correlation between search query results, bayesian filtering, and other classification techniques. Most of the work in these areas have been using the concrete example of product reviews. Some of our work involved finding ways that the algorithms need to be tuned and tweaked to get good results on news data. [7, 5]

Bayesian Classification Our first method of rating an article is accomplished by using a Naive Bayes text classifier, the point of which is to compute the probability of a document D being in a class C. In our case, an article will either be in a positive or negative class, but with a varying degree of polarity.

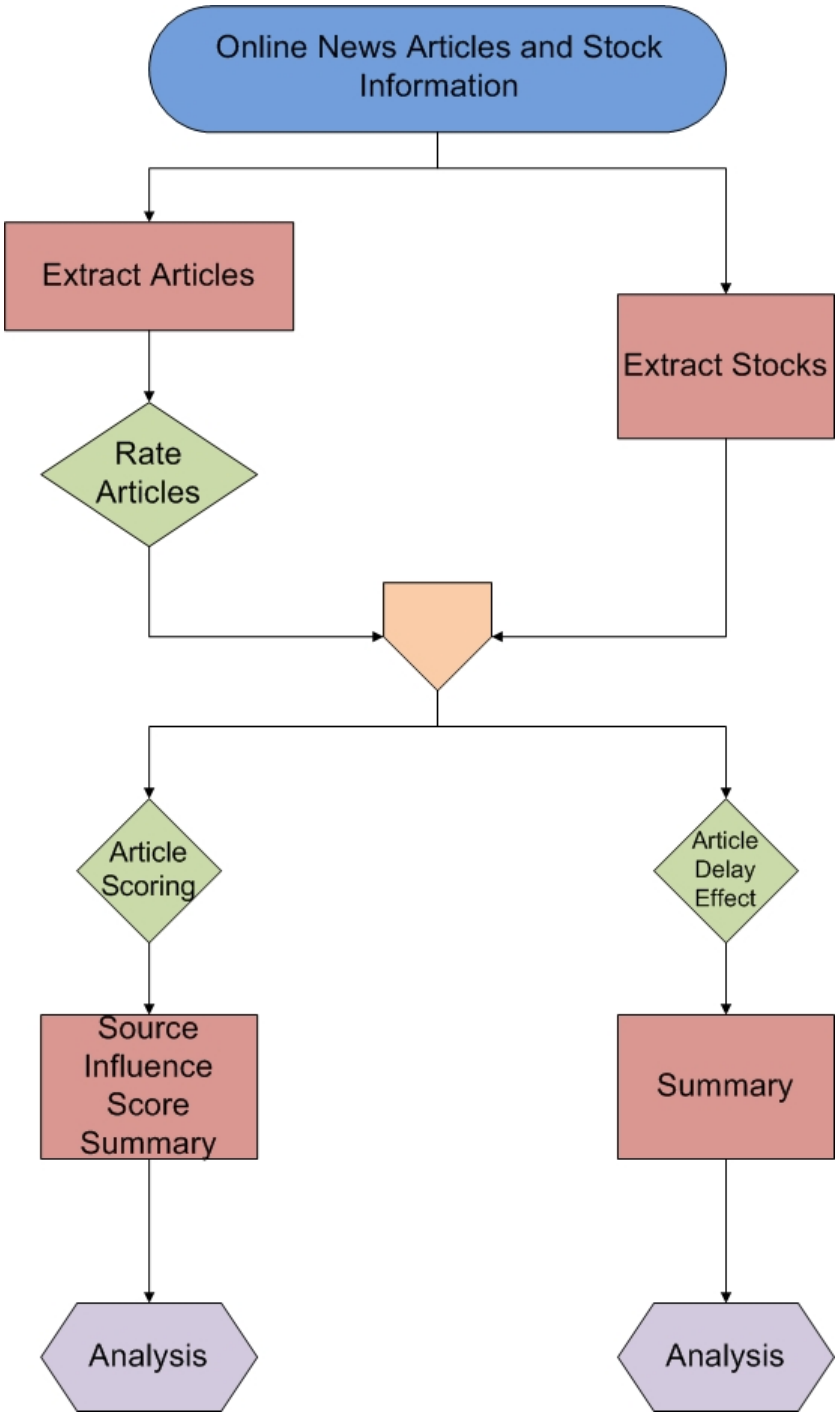

Fig. 1. Design of Our System

Our algorithm is constructed as outlined in the Naive Bayes text classification document published by scholars at Stanford[5], with a few exceptions.

We begin by using a training set of articles that are strongly polarized in their respective class. These articles were found manually using the Google News Archive. In addition, the training set consists of an equal number of positively and negatively polarized articles. We then train our algorithm by examining the text from all of the articles in our training set, on a class by class basis. As each word is extracted from the text, the algorithm calculates the conditional probability that the given word will appear in a document of its respective class. As more of the same words are discovered, each word's conditional probability will increase. After all of the training articles have been examined, the algorithm is ready to rate the articles received from the article extractor. The training algorithm is detailed in **Algorithm 1**.

Algorithm 1 TrainNB(C,D)

$V \leftarrow ExtractVocabulary(\mathrm{D})$
$N \leftarrow CountDocuments(\mathrm{D})$
for each $c \in \mathrm{C}$ **do**
  $N_{\mathrm{c}} \leftarrow CountDocumentsInClass(\mathrm{D}, c)$
  $prior[c] \leftarrow N_{\mathrm{c}}/N$
  $text_{\mathrm{c}} \leftarrow ConcatenateTextOfAllDocumentsInClass(\mathrm{D}, c)$
  for each $t \in V$ **do**
    $T_{\mathrm{ct}} \leftarrow CountTokensOfTerm(text_{\mathrm{c}}, t)$
  end for
  for each $t \in V$ **do**
    $condprob[t][c] \leftarrow \frac{T_{\mathrm{ct}}+1}{\Sigma_{t\prime}(T_{ct\prime}+1)}$
  end for
end for
return *V, prior, condprob*

As the class of each article is unknown, the algorithm computes two scores for each article representing the probability that the given article is either positive or negative. Each score is initialized with the ratio of the number of documents in that class to the number of documents total. Then the logarithm of that value is calculated and stored as the score. From there, each score continues to add the logarithm of the respective conditional probability for each word found in the article. Once all of the text has been examined in the article, our algorithm then performs several computations to determine its article rating. The algorithm for calculating the positive and negative scores is detailed in **Algorithm 2**.

The final score calculation helps us in several ways. Most noticably, it outputs the article rating in the correct range from -1 to 1; unlike the raw output. More importantly, the final score takes into account that an article can be both positive and negative as oppsed to just being classified strictly as positive or negative. The first calculation taken for the final score is the subtraction of the negative score from the positive score. That value is then divided by the minimum value

Algorithm 2 ApplyNB(C,V, *prior, condprob, d*)

```
W ← ExtractTokensFromDocument(V, d)
for each c ∈ C do
  score[c] ← log prior[c]
  for each t ∈ W do
    score[c]+ = log condprob[t][c]
  end for
end for
return FinalScore calculation
```

between each score's absolute value. From here, the final score is multiplied by 10 to ensure that it is in the correct range from -1.0 to 1.0. A final calculation can be written as follows:

$$\text{ACR} = \frac{\textit{positive score} - \textit{negative score}}{\text{MIN}(|\textit{positive score}|, |\textit{negative score}|)} * 10$$

It is important to note that the choice of multiplying by 10 was not derived in any exact way. Instead, it was found by inspection of typical results. Since this can sometimes lead to values outside of the range [-1, 1], our algorithm changes any score of greater than 1 to be 1. Likewise, scores of less than -1 become -1.

Phrase-Based Approach Our second article rating method builds off of work done in [7] that uses two-word phrases meeting criteria based on part-of-speech. This method also uses web search data to determine word associations. This method begins by tokenizing the input text into sentences, each an ordered list of single words. Then, every word is assigned a part-of-speech using a part-of-speech tagger. We will refer to the words now as tagged words to convey the fact that they have been annotated with their part-of-speech. At this point, our algorithm slightly diverges from what was described in [7]. In the original work, they extracted 2-word phrases meeting certain criteria, sometimes also "looking ahead" to a third word. The criteria for selecting tagged words as phrases involves specific part-of-speech words occurring in certain orders. These are patterns like "adjective noun" that are likely to convey positivity or negativity. The mentioned difference between the original work and our implementation of the algorithm is that we do not place any constraints on the third word as we do not find it to be useful in our dataset.

Each of these phrases are evaluated to determine their semantic orientation. This is a numerical representation of how the phrase conveys a positive or negative sentiment. To do this, specific web searches are performed that strive to establish the co-occurrence of each phrase with a pre-established positive word and with a negative word. In the original paper the words "excellent" and "poor" were used. The following formula was used to establish the semantic orientation of each phrase where the $h(phrase)$ function represents the number of hits re-

turned by the web search for the given query:

$$\text{SO}(phrase) = \log_2 \frac{h(\text{`poor'}) \cdot h(\text{`excellent } \text{``}phrase\text{''`})}{h(\text{`excellent'}) \cdot h(\text{`poor } \text{``}phrase\text{''`})}$$

While the original paper used AltaVista as the search engine used for getting hit counts, we decided to use Google. This decision was based on Google's extensive index of pages on the internet and its general acceptance as the dominant search engine. Additionally, we felt that using a single positive and single negative word did not give a strong enough of a signal as to the positivity or negativity of each phrase. Instead, we chose to use a set of positive baseline words and a set of negative baseline words. Since the original formula was written with one of each of the two types of baseline words, we had to make a slight modification. Instead of the *hits* function being the count of one of the baseline terms or one of the terms with the phrase, we replace it with a combined value over all positive or all negative baseline terms. In symbols, with combining function s we have positive word set P and negative word set N giving our semantic orientation function as:

$$\text{SO}(phrase) = \log_2 \frac{s(h(\text{`}n_1\text{'}), h(\text{`}n_2\text{'}), ...) \cdot s(h(\text{`}p_1 \text{ ``}phrase\text{''`}), h(\text{`}p_2 \text{ ``}phrase\text{''`}), ...)}{s(h(\text{`}p_1\text{'}), h(\text{`}p_2\text{'}), ...) \cdot s(h(\text{`}n_1 \text{ ``}phrase\text{''`}), h(\text{`}n_2 \text{ ``}phrase\text{''`}), ...)}$$

After trying several combinations of combining functions, positive word sets, and negative word sets, we chose the following:

$$s = \text{median}$$

$$P = \{\text{excellent}, \text{good}, \text{positive}, \text{better}, \text{outstanding}, \text{gain}, \text{up}, \text{happy}, \text{encouraged}\}$$

$$N = \{\text{poor}, \text{bad}, \text{negative}, \text{worse}, \text{terrible}, \text{loss}, \text{down}, \text{sad}, \text{discouraged}\}$$

.

Another problem that we noticed with the algorithm was that it seemed to produce ratings close to zero. Such results were not particularly desirable to us given that we were using articles that we classified as very positive or very negative. Upon further inspection of the phrases and their semantic orientation scores, we were able to see that some phrases being included did not convey positive or negative sentiment. To combat their influence of bringing results towards zero, we decided to drop their inclusion in the final score. Specifically, phrases with semantic orientations having an absolute value of less than 0.1 were ignored.

Combining Rating Methods After we developed both the phrase-based and Bayesian approaches, we realized that neither stood out as a well-performing system. In order to take advantage of both methods strengths, we realized that we needed to somehow combine the results from each approach. With limited time left in the quarter, we decided to use a simple average of the two rating quantities to determine the final result. This generally improved the results since the two algorithms seemed to usually perform bad on different sets of articles. There were some articles where both algorithms gave unsatisfactory results and in those cases the average caused some results to worsen.

4.3 Stock Price Extraction

We procure historical stock data (high, low, open, close, volume, etc.) off of the Internet and store them in our own database for quick access. This process simply accesses a Google Finance URL that returns a specified range of stock data for a certain company and persists them to a local database for quick, Internet-less access in the future. The process can be repeated for any number of companies and any number of dates, so as long as they exist in Google's records. Google returns a comma-delimited text file (.csv) for each company and set of dates that is easily parsed and committed to a database via a few Java ODBC methods. The specific dataset obtained in Stock Price Extraction is detailed in Section 3.2

4.4 Article Prediction Scoring and Summary

Article Scoring This component of the system gives a unit-less score to an article based on the performance of the stock of the company that the article is about and the rating of that article. It is important to note the range of values that our scoring algorithm "allows" an article to have (for ease of analysis): we define a "perfect" stock performance (+1) to be a one-day growth of 10% (or more); similarly, a -1 is given to stocks that drop 10% or more in a day. This is used to maintain consistency in our analysis and to ensure that normalization of stock data does not favor or discriminate against stocks that perform beyond these bounds.

In our algorithm, if an article rating shares the same sign as the stock price change (e.g., a positive rating and an increase in stock price), the article will receive a positive score, and vice versa. The closer the stock performance and rating value, the higher the score given to an article. The scorer uses the same [-1,1] scale that is used to rate an article and stock performance for consistency. An exponential scoring system returns a fractional, unit-less score equal to:

$$\text{score} = \left(\frac{\min(\text{abs}(articleRating), \text{abs}(stockPerformance))}{\max(\text{abs}(articleRating), \text{abs}(stockPerformance))} \right)^2$$

for articleRating and stockPerformance values that share the same sign. This value is near 1 for ratings that are nearly equal to the stock performance, and close to 0 (but still positive) for ratings that don't reflect stock performance but still share the same sign. For ratings and stock performances that are of opposite signs, the score is:

$$\text{score} = -\left(\frac{articleRating - stockPerformance}{2} \right)^2$$

This value will be negative to reflect the fact that the article incorrectly predicted the direction of the stock price movement the following day.

Source Influence Score Summary Once the source influence scoring has occurred, the score must be grouped in order to determine the news source's influence on a company's stock price. One way to achieve this goal is to simply average all of the source influencing scores for a particular news source.

Another benefit of this method of calculating the influence score for all company-news source pairs is that other statistics can also be generated. They are described in the following list.

1. The influence each news source has in general over all companies (the main calculation)
2. How companies are influenced over all news sources
3. How much a particular company is influenced from a particular news source

4.5 Article Delay Effect and Summary

Article Delay Effect In order to address delays between an article being published and its effect on the stock market, we have implemented a program to generate errors between articles predictive rating and the actual change in the stock market. We do this with a variation of the sliding window algorithm. The slide portion of the algorithm is the delay in days which an article is said to have an effect. In our experiments we look at slides 1 (one day delay) to 30 (a month delay) and find the minimum error to estimate how long an article takes to effect the stock market.

The window aspect of the algorithm is used in calculating the error between the predicted stock change using the article ratings and what actually occurred with the stock market. A window size of 1 represents one day's worth of articles prediction of one day's worth of stock. A window size of 7 represents a week's worth of articles prediction of a week's work of stock. Window sizes of 1 to 30 were analyzed. An example report can be seen in Figure 2. The color legend for the result screen can be seen in Table 1.

Color	**Error Value** (% Stock Change)
Green	$< .1$
Yellow	$< .5$
Red	$> .5$

Table 1. The Color Legend for the Article Delay Effect Report

Article Delay Effect Summary While creating a massive report with all values for window and slide is useful, a more summarized version allows a data analyst another view of the data. The summary view is a two-dimensional table with one axis being slide amount and the other being window size. An example version of this table can be found in Figure 3.

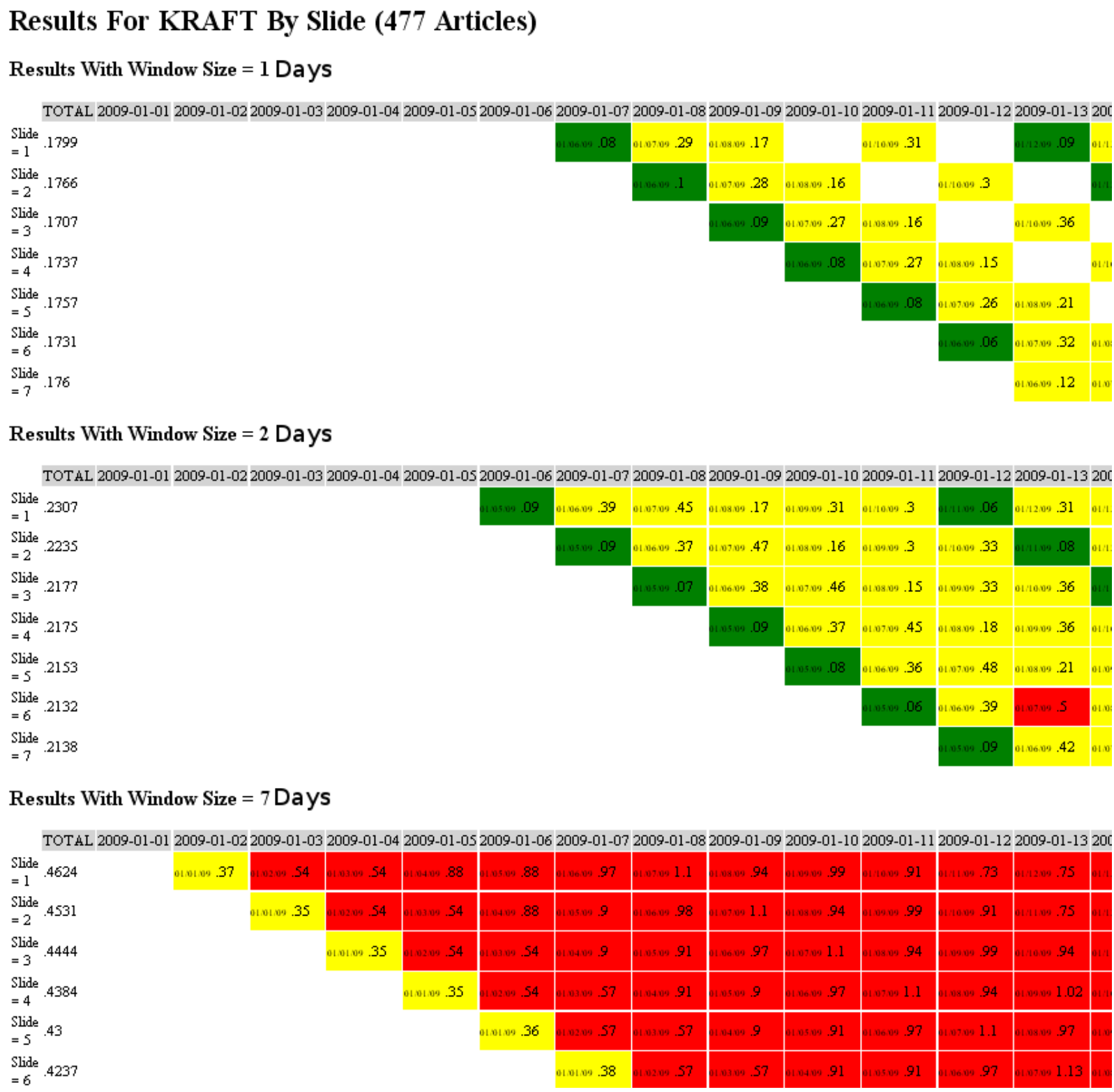

Fig. 2. An Example of the Sliding Window Results

GOOGLE Average Results

Slide Amount (Days) \ Window Size (Days)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	.1557	.213	.2668	.3229	.3824	.446	.5089	.5741	.6391	.7033	.7695	.8374	.9064	.9752	1.0433
2	.1563	.2135	.2668	.3232	.3827	.4433	.5068	.5725	.6375	.7022	.7685	.8362	.9053	.9738	1.0417
3	.158	.2142	.267	.3234	.3803	.4416	.5045	.5703	.6356	.7005	.7672	.8344	.9035	.9718	1.04
4	.1574	.2148	.2667	.3202	.3764	.4383	.5022	.568	.6334	.6984	.7652	.8326	.9011	.9699	1.0382
5	.1584	.2141	.2637	.3164	.3728	.4357	.4999	.5656	.6309	.6959	.7629	.8303	.8995	.9678	1.0352
6	.1574	.2105	.2593	.3131	.3704	.433	.4968	.5629	.6277	.6926	.7596	.8273	.8966	.9645	1.0323
7	.1533	.207	.2575	.3121	.3702	.432	.4952	.5609	.6258	.6905	.7577	.8256	.8942	.9623	1.0304
8	.1505	.2065	.2583	.3128	.372	.4327	.4955	.5608	.6262	.6913	.7583	.8255	.8944	.9627	1.0304
9	.1516	.2074	.2592	.3151	.3728	.433	.4954	.5609	.6266	.6918	.7582	.8255	.8943	.9624	1.0303
10	.1524	.208	.2606	.3152	.3732	.4336	.496	.5617	.6275	.6915	.7587	.8259	.894	.9622	1.0304
11	.1519	.2079	.2587	.3145	.3733	.4341	.4974	.5628	.6276	.6917	.7586	.8255	.8938	.962	1.0304
12	.1539	.2069	.2586	.3144	.3737	.4355	.4993	.5637	.6282	.6922	.7578	.8246	.8934	.9617	1.0299
13	.1508	.2063	.2584	.3142	.374	.437	.4996	.5641	.6289	.6916	.7571	.824	.8926	.9612	1.0293
14	.1526	.2075	.2586	.3151	.3746	.4369	.5005	.5651	.6292	.6923	.7574	.8245	.8929	.9609	1.0291
15	.1551	.2099	.2618	.318	.3756	.437	.5009	.5656	.6298	.6931	.7584	.8251	.8931	.961	1.0294
16	.1565	.2121	.2643	.32	.3774	.4389	.5024	.567	.6309	.6941	.7588	.8251	.893	.9614	1.0301
17	.1579	.2145	.2655	.321	.3791	.4398	.5031	.5678	.632	.6948	.7583	.8251	.8936	.9621	1.0311
18	.1585	.2137	.2657	.3209	.3783	.4396	.5033	.5682	.6319	.6944	.758	.8248	.8939	.9628	1.032
19	.1564	.2115	.2645	.3199	.3765	.4388	.5032	.5676	.6307	.6936	.7579	.8247	.8941	.9636	1.0331
20	.1558	.2122	.2633	.3183	.3763	.4384	.5028	.5673	.6309	.6941	.7586	.8264	.896	.9657	1.0326
21	.1563	.2109	.262	.3168	.3766	.4389	.5019	.5665	.6307	.6943	.7594	.8278	.8982	.9652	1.0316
22	.1554	.2105	.2617	.3171	.3769	.4378	.5005	.5655	.6303	.6948	.7609	.8294	.8974	.9642	1.0305

Fig. 3. An Example of the Sliding Window Summary

5 Implementation

Our entire project was coded in the Java programming language. For the storage of our data, we used a MySQL database, contacted by our program using the JDBC (Java database connector). Our code used several Google APIs as well as their collections library. In order to extract text from raw HTML, we used the Jericho HTML parsing library. For the phrase based article rater, we used the Stanford Posttagger Library and their given models of the English language for determining the part of speech of words. Finally, we used the free library provided by `http://www.json.org` for getting data from JSON strings.

In terms of system design, we tried to keep decision and calculation steps in our algorithms easily modifiable by extending an interface. For instance our system lets a user easily plug in different methods of article rating. We also attempted to optimize the speed of our code by taking advantage of multithreading by use of Java's built in concurrency library.

6 Experiments, Results, and Analysis

This section details the results of our experiments. Our experiment analyzed articles about the companies in the Table 2., and the prices of the stocks of those companies. The date range of the articles and stocks in our data set is from January 1, 2009 - December 3, 2009. Stock prices are available for every week day in that range except for federal holidays; articles are normally available daily (depending on the output of news sources).

Company Name	**Company Stock Symbol**	**Stock Market**
GOOGLE	GOOG	NASDAQ
YAHOO	YHOO	NASDAQ
FORD	F	NYSE
DISNEY	DIS	NYSE

Table 2. The Companies Used In Our Experiments

6.1 Article Company Rating

The article company rating part of our code posed a challenge in terms of evaluating our results since we believed its meaning to be very subjective. We determined that in the limited time of a quarter project, that we would have to use manual evaluation of a random sample of rated articles. Specifically, two members of the research team each manually reviewed 60 randomly selected articles. Of these, 20 were common to both reviewers in order to verify that both raters were giving similar scores. The other 40 articles were unique to each reviewer.

Before discussing the results concerning the ratings themselves, it is important to discuss a few issues that arose during the test. These problems stem from our simplistic and automatic technique for finding articles. First, there were several occurrences (about 2%) in which a 404 not found return code was returned by a web server that was supposed to have an article with a given URL. It turns out that while Google's index contained the article and was able to match its text/headline to our search query, the site from which it was extracted removed the content since the crawl. Another problem that we found in the pages was that some articles were not about the company that was in the query. This was more often the case when the company's name was a normal English word or phrase for instance "Southwest". Other companies, such as Google, with more unique names did not have this problem. Mistaken articles account for about 50% of articles rated. The last issue that came up was that sometimes a company would be referenced in an article that is really about another company. For instance, in an article about Microsoft's Bing search engine the article compares it to the Google search engine. Overall, this made only 50% of our articles useful to determining the accuracy of the rater.

Since this section should be purely about the rating component, the analysis presented here only ignores articles with 404 errors. Ideally the extraction component would also do some verification to ensure that the extracted articles contain the correct company. First, of the 20 common articles, 11 were found to be suitable for rating. For each of the 11 articles, the absolute value of the difference between the ratings given by the 2 members of the research team was computed. The average of these values came out to 0.365. This confirms our belief that the ratings are highly subjective and thus difficult to measure precisely.

In terms of comparing the manual ratings to the ratings given by the algorithm, the results were very intriguing. Of the possible 120 ratings possible, only 72 were considered due to the aforementioned issues with the extraction process. First, the average difference between the manual rating and the computer rating came out to 0.001 which at first glance would be considered an excellent result. However, on closer inspection of the individual results it seemed that the quality of the rating on individual articles was not as good as we would have liked. To measure this, we took the average of the absolute value of the difference of the ratings in order to figure out how off each result was on average. This computation came out to 0.359. Together, we believe that this shows that our rating system did not perform well when considering an individual articles. However, given that the average difference was only 0.001, it would seem that our system performs well when presented with a large set of articles. Another important comparison that should be made is the similarity of the average absolute value of difference between the two manual raters and the algorithm ratings. That is, there was an average absolute value of difference of 0.365 between the two human raters and a 0.359 average absolute value of difference between the human raters. This goes to show that our system has the subjectivity of just another human.

Static Rater	Average Absolute Value of Difference
Random Rater	0.528
-1 Rater	1.003
0 Rater	0.997
1 Rater	0.508

Fig. 4. Results of tests with static raters.

We also confirmed that our method performs a reasonably intelligent computation by comparing our results to several rating methods that did not look at the text. These raters were: a rater that assigned a different random rating to each article, one that always assigned a -1, one that always assigned a 0, and one that always assigned a 1. The results of these tests are given by the table shown in figure 4.

The main reason that we believe that the results are not more positive is the extra textual content that is on web pages. This is text such as headers and ads that add noise to our rating systems. This would be solved by a better extraction process that either was able to directly get the article text or by a system for detecting and eliminating the text content of article pages that is not the article itself. We also had significant negative impact from articles not about the company that our system considered them to be about.

6.2 Day-to-Day Association

The results of running our program show that there is, on average, an association between news articles and the stock performance of the four companies that we chose to analyze. Following below are small subsets and examples of the whole data, which can be found in Appendix A.

For 288 articles on Ford, the average score is .049; for 202 articles on Google, .095; for 340 articles on Yahoo, .101, and .108 for the 326 articles on Disney. It is important to note that because the score for each article is an unweighted, unit-less score, these Company Influence values do not necessarily mean that the majority of articles about these companies have a positive association (i.e., predict the sign of the stock movement correctly). Figure 5 below shows the average article score for all articles over all companies.

Num Articles	Company	Average article score
288	FORD	0.049
202	GOOGLE	0.095
340	YAHOO	0.101
326	DISNEY	0.108

Fig. 5. Company Influence Score

A more useful and granular measure is that of the News Source Influence. This value shows the average of the scores of the articles from a particular news source (e.g., Wall Street Journal) about the four companies. As an example, with 27 articles, the Wall Street Journal (WSJ) averages a score of -.072 - that is, the average of articles from the WSJ predict opposite (even if just barely) the performance of the stock on which it writes the following day. Like the Company Influence value, it does not mean that the majority of articles are incorrect; instead, 26 articles may see a slight positive association (low positive score), and one article could have performed extremely poorly. Figure 6 below shows an example of a few well-known news sources and how well, on average, their rating predicted the price of stock the following day.

Num Articles	News Source	Average News Source Score
27	online.wsj.com	-0.072
18	www.msnbc.msn.com	-0.012
32	www.telegraph.co.uk	0.027
83	www.reuters.com	0.066

Fig. 6. News Source Influence Score

Another metric weaned from our analysis is the Company-News Source Influence - how much a particular company's stock is influenced by a particular news source. As an example, 36 articles about Ford from www.reuters.com have a .063 average score. This value is subject to the same clause as the Company Influence and News Source Influence values. Figure 7 below shows an example of how well or poorly a few news sources rated particular companies.

Num Articles	Company	News Source	Average Score
22	DISNEY	www.latimes.com	0.111
13	FORD	www.businessweek.com	0.058
37	YAHOO	www.washingtonpost.com	0.084
23	GOOGLE	www.informationweek.com	0.102

Fig. 7. Company News Source Influence Score

The last and probably best metric of our analysis, in Figure 8 below, shows an example of how many news articles from a given news source had a positive score (i.e., how many article ratings shared the same sign as the movement of the stock the following day). The table shows that no news source had a better than 50% showing, which means, more often than not, the news source predicted the performance of the stock opposite to its actual performance.

Though these numbers in specific are discouraging, as Figure 5 shows, the average of all of the article scores is positive across the board. This may suggest that though there are a smaller number of correctly predicted articles (532 out

News Source	Company	Num Articles	Num Same Sign
www.reuters.com	YAHOO	17	7/17 (41%)
www.informationweek.com	GOOGLE	23	10/13 (43%)
www.nytimes.com	DISNEY	20	10/20 (50%)
www.marketwatch.com	FORD	14	6/14 (43%)

Fig. 8. News Source Influence Per Company - Percent Correct

of 1156 [46%]), those articles that did share the same sign value were closer in magnitude to the stock performance than those that were not.

6.3 Article Delay Effect

Since our program allows the generation of error reports for any time period using the sliding window concepts, we have a large amount of report data to analyze. One of the main questions which can be answered using the generated reports is what is the most frequent time delay required for an article's rating to associate with the change in the stock market. By looking at the slide rows, we can find the row with the least about of error. Looking at the table in Figure 9, the least amount of error appears to be the day after the article is written. This is consistent for all the other companies analyzed in our experiments.

Another interesting relation found in our results is that the stock prediction error increases as the window size increases. This phenomenon can be seen in Figure 3. One possible explanation for this is in the way our algorithm calculates error for a given time window. Currently, the algorithm sums all of the article ratings for the days in the window to determine the total rating. This total rating is then translated to an estimated change in the stock market. Unfortunately, the average article in our dataset has a rating of 0.13 which corresponds to a small increase in stock value. This is problematic since as the system sums the article ratings for each day in the window, the percent increase in stock over the period is linearly unbounded and creates optimistic stock performance.

One way which our algorithm can be modified to alleviate this issue would be to use the log function in conjunction with the summed_article_rating for the window period. This would have the effect of negating large estimates in stock change which tend to not occur in the stock market, thus decreasing our average error.

7 Future Work

Although our program has identified associations between news articles and the stock market, future work can be accomplished to further support current and new associations as well as to expand the scope of our project.

One improvement that could be made, would be a company-specific stock price change threshold. As each company has a different price for one share of stock, each company will have a different percentage in increase or decrease of

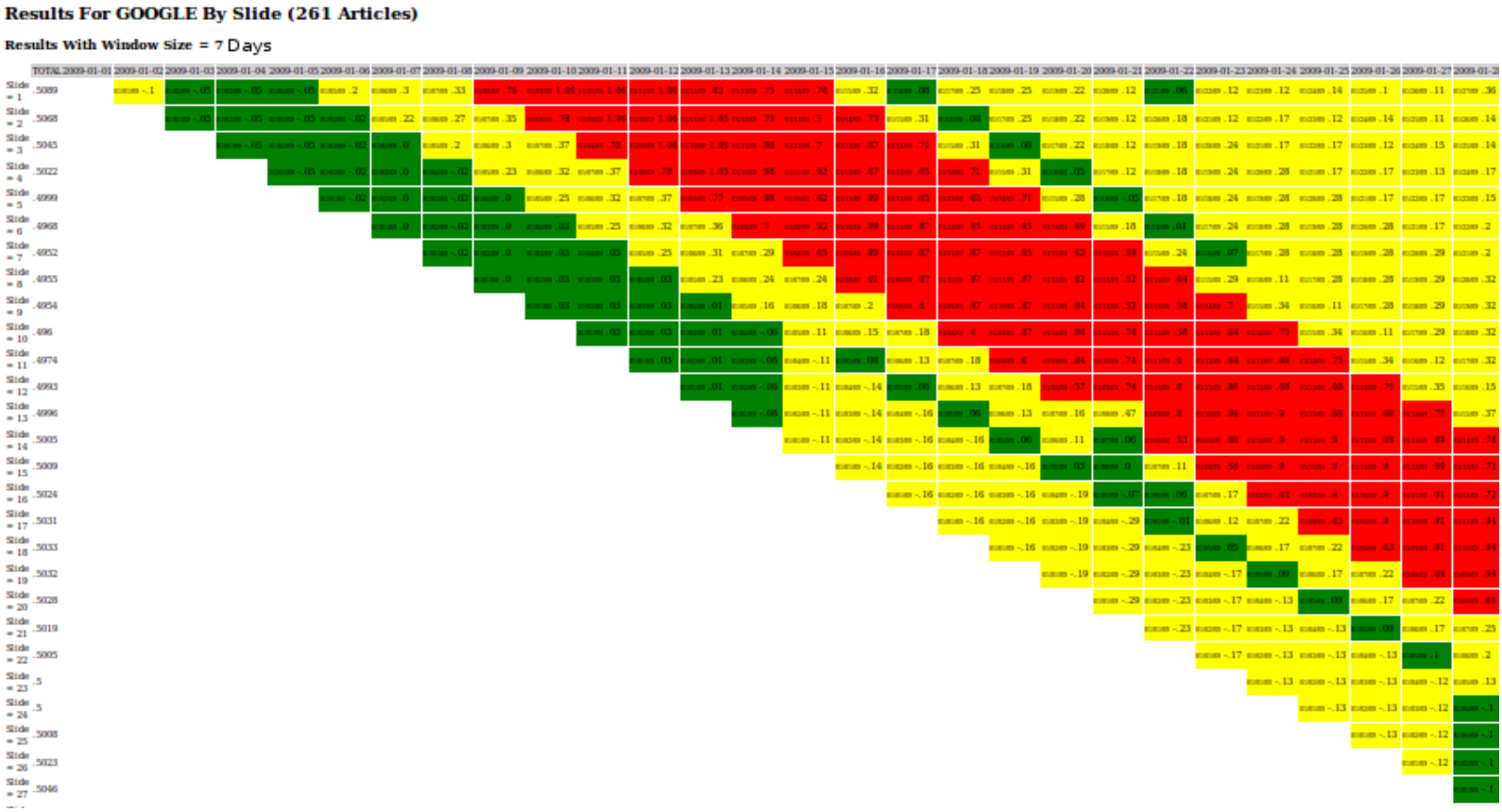

Fig. 9. The Table which Provides the Answer to the Article Effect Delay

value. For instance, a company who has a share of stock that is worth $100, will probably only change in small percentages. While a company whose stock is worth only a few dollars will have changes in larger percentages. Our system currently reviews articles and expects a general percentage increase or decrease, independent of a company's stock price.

Several improvements could be made to increase the effectiveness of our article rating and scoring systems:

We could add another depth to identifying an association between a news source and the stock market. Determining the author of a news article could be a potential reason why the stock of a company would change. The assumption would be that a more renowned journalist would have a greater affect to a company's stock than would a less prominent journalist.

Grasping the size of the readership base for a given news source would also help in rating an article. Obviously, the larger and more popular news sources would have a larger affect on a company's stock than would a smaller news source. In addition, determining if an article found online had been published in paper-printed format would be essential in helping to identify the readership base for a particular article. Identifying the location of where an article was published would further help understand the readership base of a news source and how much affect that region has towards the stock market. Lastly, the time at which an article is published would also determine the readership size of a particular article. For example, an article published in the middle of the night would most likely have a smaller readership than those produced during the day.

Another general problem with articles, is their possibility of multiple publications. Part of this stems from the fact that some news sources have the same parent company, and as such, articles could be published through each child news source. An excellent example would be the Retuers company as they have a separate news outlet in several countries across the world. Also, several news sources publish their articles based on the findings of Associated Press reports. As a result, several news sources will publish the same article. Our system currently treats these as separate articles without any relation to each other, causing our expectations of a company's stock to rise or fall unnecessarily. Articles that stem from the same parent company, or are taken from an AP report should be analyzed as a single article to correctly determine it's effect on the stock market.

Lastly, a general improvement that could be made is the process of article extraction. This problem has been mentioned in sections before hand, and is probably the source of several issues in our current system. Being able to extract text without extra noise would be excellent, as the noise (such as ad text, header content from the web page, etc.) caused our scoring system to not behave correctly. Having a system that can identify if a given company, is indeed, the subject of that article, as opposed to just being mentioned in the article, would be a needed addition. Also, it is imperative that the exractor can determine if a linked article actually exists, i.e., not a 404, not a member-only article, etc.

8 Conclusions

The goal of this project was to attempt to identify associations between news article reports on companies and the performance of the securities (stocks) of those companies in the following days or weeks. We were successful in implementing a number of separate but dependent solutions into one cohesive solution that allowed us to extract data and aid our analyses.

With specific concern to the rating part of the project, it was shown that overall, the subjective human-rated articles were nearly identical to the same articles graded by the raters. However, the average of the absolute value of the difference in ratings was a bit higher, as noted in section 6.1. Though larger than the ideal value (0.0), it is positive to note that it performed better than any of the raters seen in Figure 4.

Generalized associations were found as evidenced by positive average article scores - that is, on average, the news articles ratings as determined by our Bayesian Classifier and Phrase-based approach had a positive association with the performance of the stock the following day. This data is partially discounted by the fact that less than half of the articles (46%) shared the same sign as the stocks, but helped by the fact that those articles that did share the same sign had a stronger association (i.e., the articles more closely predicted the stock performance) than those that did not.

With concern to the generalized time association, it appears that the strongest association of a stock or group of stocks comes in the days immediately following the publication of articles - a phenomenon that lends greatly to the day-to-day association that was the main purpose of this project.

While this project still leaves many questions unanswered and addresses only a fraction of those desired in a full analysis of the stock market, given the scope of the project, the resources available, and the inherent unpredictability of the stock market, the project was nonetheless successful in offering a brief though thorough glance at the association between news articles and the performance of the stocks of the companies addressed in those articles in the following days.

References

1. Candlestick Trading Forum, *Stock market terms - saying it the right way*, `http://www.candlestickforum.com/PPF/Parameters/11_1419_/candlestick.asp`.
2. Google, *Google finance*, `http://www.google.com/finance/historical?q=<market>:<symbol>`.
3. ———, *Google news archive search*, `http://news.google.com/archivesearch`.
4. MarketPsych LLC, *Marketpsych llc*, `http://www.marketpsych.com/`.
5. C.D. Manning, P. Raghavan, and H. Schutze, *Naive bayes text classification*, An introduction to information retrieval, 2009, pp. 258–262.
6. Richard Peterson, *Neuroeconomics*, `http://www.richard.peterson.net/Neuroeconomics.htm`.
7. Peter Turney, *Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 417–424.

8. Wikipedia, *Association (statistics)*, http://en.wikipedia.org/wiki/Association_(statistics).
9. ———, *Behavioral economics*, http://en.wikipedia.org/wiki/Behavioral_finance.
10. ———, *Stock market*, http://en.wikipedia.org/wiki/Stock_market.

A Source Influence Results

The following pages include more detailed results regarding the source influence summary.

Company Influence

Num Articles	Company	Average article score
288	FORD	0.049
202	GOOGLE	0.095
340	YAHOO	0.101
326	DISNEY	0.108

News Source Influence

Num Articles	News Source	Average Score
1	www.asiaone.com	-0.096
1	www.miamiherald.com	-0.087
1	www.indianexpress.com	-0.083
1	www.stuff.co.nz	-0.082
1	news.smh.com.au	-0.074
1	travel.nytimes.com	-0.072
27	online.wsj.com	-0.072
4	www.npr.org	-0.064
3	www.brisbanetimes.com.au	-0.064
3	www.dailymail.co.uk	-0.057
2	www.theatlantic.com	-0.048
4	money.cnn.com	-0.048
1	www.abcnews.go.com	-0.04
2	www.infoworld.com	-0.036
1	www.tehrantimes.com	-0.032
1	www.straitstimes.com	-0.03
3	www.theinquirer.net	-0.026
4	www.networkworld.com	-0.025
1	www.calendarlive.com	-0.025
1	www.eastandard.net	-0.024
3	english.chosun.com	-0.021
2	news.asiaone.com	-0.02
8	sports.espn.go.com	-0.02
1	www.zdnetasia.com	-0.02
1	www.csmonitor.com	-0.015
2	www.hollywoodreporter.com	-0.015
1	www.buffalonews.com	-0.014
5	www.timesonline.co.uk	-0.014
2	www.mlive.com	-0.013
1	www.rockymountainnews.c	-0.013
1	seattletimes.nwsource.com	-0.012
18	www.msnbc.msn.com	-0.012
1	showbizandstyle.inquirer.ne	-0.01
1	www.freep.com	-0.01
1	www.cbc.ca	-0.009
1	fr.reuters.com	-0.009
2	www.canada.com	-0.008
1	www.newsobserver.com	-0.006
1	www.theage.com.au	-0.004
11	www.itpro.co.uk	-0.003
1	www.cbssports.com	-0.002
1	www.news.com.au	-0.001
1	feeds.wired.com	0
1	www.dawn.com	0
2	www.internetnews.com	0.001
2	entertainment.timesonline.c	0.001

2	www.chicagotribune.com	0.001
4	www.theregister.co.uk	0.001
1	www.taipeitimes.com	0.002
1	business.theage.com.au	0.003
1	www.dailynews.com	0.004
3	www.chinadaily.com.cn	0.014
3	www.nydailynews.com	0.017
1	detnews.com	0.025
1	www.projo.com	0.025
1	www.chitramala.com	0.027
32	www.telegraph.co.uk	0.027
1	www.thetimes.co.za	0.028
12	news.bbc.co.uk	0.031
9	www.boston.com	0.031
3	www.cbsnews.com	0.034
2	sports.sportsillustrated.cnn.	0.034
3	dir.salon.com	0.036
7	business.timesonline.co.uk	0.037
9	www.time.com	0.037
1	www.mirror.co.uk	0.041
13	www.salon.com	0.042
3	www.adweek.com	0.049
5	www.nypost.com	0.053
3	www.orlandosentinel.com	0.054
1	news.nationalgeographic.c	0.055
5	movies.nytimes.com	0.06
33	in.reuters.com	0.062
1	apps.money.cnn.com	0.064
4	rss.cnn.com	0.065
18	www.sfgate.com	0.065
83	www.reuters.com	0.066
28	www.usatoday.com	0.067
33	www.businessweek.com	0.07
1	www.gmanews.tv	0.071
53	www.marketwatch.com	0.077
28	www.pcworld.com	0.078
26	www.guardian.co.uk	0.079
32	www.thestreet.com	0.081
3	www.foxbusiness.com	0.081
24	www.wired.com	0.092
4	www.sportingnews.com	0.092
32	abcnews.go.com	0.095
53	www.forbes.com	0.096
2	www.eweek.com	0.099
3	www.theglobeandmail.com	0.099
54	www.washingtonpost.com	0.101
61	news.yahoo.com	0.106
8	www.independent.co.uk	0.107
2	www.upi.com	0.107
34	www.informationweek.com	0.111
26	www.foxnews.com	0.111
31	www.latimes.com	0.111
3	cgi.money.cnn.com	0.122
49	www.nytimes.com	0.137
24	sify.com	0.139
1	business.watoday.com.au	0.142
5	news.xinhuanet.com	0.142

35	uk.reuters.com	0.145
1	www.hurriyet.com.tr	0.153
1	www.itworld.com	0.166
1	www.nebraska.tv	0.169
1	www.nationalpost.com	0.171
3	www.watoday.com.au	0.186
15	www.bloomberg.com	0.188
2	www.financialpost.com	0.194
1	www.twincities.com	0.195
6	www.cctv.com	0.198
11	ca.reuters.com	0.201
2	www.techtree.com	0.203
3	www.philly.com	0.211
3	www.etaiwannews.com	0.219
1	www.channelnewsasia.com	0.22
3	www.chron.com	0.227
3	news.cnet.com	0.237
4	business.theatlantic.com	0.238
3	www.rollingstone.com	0.239
1	www.timesdispatch.com	0.249
3	www.newsweek.com	0.25
5	www.bizjournals.com	0.257
1	www.startribune.com	0.26
3	www.business-standard.co	0.288
1	living.oneindia.in	0.305
2	correspondents.theatlantic.	0.305
1	www.mercurynews.com	0.329
2	www.detnews.com	0.347
1	www.courier-journal.com	0.349
2	sportsillustrated.cnn.com	0.35
1	www.mysinchew.com	0.449
2	www.nbr.co.nz	0.513
2	www.v3.co.uk	0.513
2	www.ctv.ca	0.552
1	www.dailyrecord.co.uk	0.647
2	www.ajc.com	0.682
1	googlewatch.eweek.com	0.685
1	www.thestar.com	0.719
2	www.mtv.com	0.751

Company - News Source Influence

Num Articles	Company	News Source	Average Score
1	DISNEY	www.indianexpress.com	-0.083
1	DISNEY	travel.nytimes.com	-0.072
10	DISNEY	online.wsj.com	-0.055
1	DISNEY	www.infoworld.com	-0.045
1	DISNEY	www.theinquirer.net	-0.036
1	DISNEY	www.foxbusiness.com	-0.034
1	DISNEY	www.mlive.com	-0.028
1	DISNEY	www.calendarlive.com	-0.025
1	DISNEY	www.etaiwannews.com	-0.024
1	DISNEY	www.chinadaily.com.cn	-0.024
2	DISNEY	www.informationweek.com	-0.022
2	DISNEY	news.asiaone.com	-0.02
2	DISNEY	sports.espn.go.com	-0.02

1	DISNEY	www.csmonitor.com	-0.015
3	DISNEY	www.sfgate.com	-0.015
2	DISNEY	www.hollywoodreporter.com	-0.015
2	DISNEY	www.nydailynews.com	-0.014
1	DISNEY	www.rockymountainnews.com	-0.013
1	DISNEY	seattletimes.nwsource.com	-0.012
1	DISNEY	money.cnn.com	-0.011
1	DISNEY	showbizandstyle.inquirer.net	-0.01
1	DISNEY	business.timesonline.co.uk	-0.01
1	DISNEY	www.cbc.ca	-0.009
1	DISNEY	fr.reuters.com	-0.009
1	DISNEY	www.theatlantic.com	-0.007
3	DISNEY	www.businessweek.com	-0.007
1	DISNEY	www.newsobserver.com	-0.006
4	DISNEY	www.nypost.com	-0.006
1	DISNEY	www.chicagotribune.com	-0.005
1	DISNEY	www.theage.com.au	-0.004
1	DISNEY	www.canada.com	-0.004
7	DISNEY	www.guardian.co.uk	-0.002
1	DISNEY	www.sportingnews.com	-0.002
1	DISNEY	www.news.com.au	-0.001
1	DISNEY	www.timesonline.co.uk	0
7	DISNEY	www.msnbc.msn.com	0
2	DISNEY	entertainment.timesonline.co.uk	0.001
1	DISNEY	www.taipeitimes.com	0.002
3	DISNEY	www.npr.org	0.002
2	DISNEY	www.dailymail.co.uk	0.004
1	DISNEY	www.dailynews.com	0.004
4	DISNEY	news.bbc.co.uk	0.019
1	DISNEY	www.projo.com	0.025
1	DISNEY	www.chitramala.com	0.027
1	DISNEY	www.thetimes.co.za	0.028
1	DISNEY	www.mirror.co.uk	0.041
3	DISNEY	www.adweek.com	0.049
6	DISNEY	www.washingtonpost.com	0.05
3	DISNEY	www.orlandosentinel.com	0.054
1	DISNEY	news.nationalgeographic.com	0.055
22	DISNEY	www.reuters.com	0.074
1	DISNEY	www.pcworld.com	0.076
4	DISNEY	movies.nytimes.com	0.08
4	DISNEY	www.salon.com	0.083
7	DISNEY	www.bloomberg.com	0.084
3	DISNEY	www.boston.com	0.088
20	DISNEY	www.marketwatch.com	0.094
6	DISNEY	www.foxnews.com	0.097
11	DISNEY	www.telegraph.co.uk	0.099
11	DISNEY	uk.reuters.com	0.106
2	DISNEY	www.upi.com	0.107
22	DISNEY	www.latimes.com	0.111
18	DISNEY	www.forbes.com	0.114
9	DISNEY	www.usatoday.com	0.118
11	DISNEY	in.reuters.com	0.12
2	DISNEY	www.cctv.com	0.153
7	DISNEY	www.thestreet.com	0.153
1	DISNEY	www.cbsnews.com	0.16
1	DISNEY	www.nationalpost.com	0.171
8	DISNEY	abcnews.go.com	0.178

1	DISNEY	www.twincities.com	0.195
20	DISNEY	www.nytimes.com	0.21
1	DISNEY	www.channelnewsasia.com	0.22
1	DISNEY	www.timesdispatch.com	0.249
1	DISNEY	business.theatlantic.com	0.255
5	DISNEY	ca.reuters.com	0.27
2	DISNEY	www.watoday.com.au	0.27
5	DISNEY	www.wired.com	0.281
3	DISNEY	sify.com	0.287
1	DISNEY	living.oneindia.in	0.305
2	DISNEY	news.xinhuanet.com	0.328
3	DISNEY	www.bizjournals.com	0.333
2	DISNEY	www.philly.com	0.341
1	DISNEY	cgi.money.cnn.com	0.38
2	DISNEY	www.chron.com	0.382
2	DISNEY	www.time.com	0.404
2	DISNEY	www.ajc.com	0.682
1	DISNEY	www.rollingstone.com	0.703
2	DISNEY	www.mtv.com	0.751
1	FORD	www.npr.org	-0.263
1	FORD	www.dailymail.co.uk	-0.179
4	FORD	www.time.com	-0.12
8	FORD	online.wsj.com	-0.118
2	FORD	money.cnn.com	-0.099
1	FORD	www.asiaone.com	-0.096
1	FORD	correspondents.theatlantic.com	-0.09
1	FORD	www.theatlantic.com	-0.088
10	FORD	www.usatoday.com	-0.087
1	FORD	www.miamiherald.com	-0.087
1	FORD	www.chron.com	-0.082
2	FORD	www.sportingnews.com	-0.047
1	FORD	www.financialpost.com	-0.044
1	FORD	business.theatlantic.com	-0.044
7	FORD	www.telegraph.co.uk	-0.042
1	FORD	www.abcnews.go.com	-0.04
2	FORD	ca.reuters.com	-0.036
1	FORD	www.straitstimes.com	-0.03
2	FORD	www.cbsnews.com	-0.028
5	FORD	sports.espn.go.com	-0.024
3	FORD	www.timesonline.co.uk	-0.023
1	FORD	movies.nytimes.com	-0.019
1	FORD	www.independent.co.uk	-0.016
1	FORD	sports.sportsillustrated.cnn.com	-0.016
5	FORD	news.bbc.co.uk	-0.013
4	FORD	www.sfgate.com	-0.012
1	FORD	www.freep.com	-0.01
1	FORD	www.etaiwannews.com	-0.006
1	FORD	www.cbssports.com	-0.002
10	FORD	www.nytimes.com	-0.002
1	FORD	feeds.wired.com	0
1	FORD	www.mlive.com	0.002
1	FORD	business.theage.com.au	0.003
1	FORD	www.bizjournals.com	0.005
1	FORD	www.chicagotribune.com	0.007
6	FORD	www.msnbc.msn.com	0.009
2	FORD	www.salon.com	0.01
2	FORD	www.theglobeandmail.com	0.011

3	FORD	news.xinhuanet.com	0.019
1	FORD	detnews.com	0.025
11	FORD	sify.com	0.026
2	FORD	www.chinadaily.com.cn	0.033
11	FORD	abcnews.go.com	0.034
3	FORD	www.cctv.com	0.041
2	FORD	www.boston.com	0.042
14	FORD	in.reuters.com	0.046
14	FORD	www.marketwatch.com	0.053
10	FORD	www.foxnews.com	0.054
13	FORD	www.businessweek.com	0.058
5	FORD	www.washingtonpost.com	0.063
36	FORD	www.reuters.com	0.063
4	FORD	business.timesonline.co.uk	0.065
10	FORD	www.thestreet.com	0.127
14	FORD	www.forbes.com	0.127
2	FORD	www.latimes.com	0.129
2	FORD	www.foxbusiness.com	0.139
5	FORD	www.wired.com	0.155
18	FORD	uk.reuters.com	0.16
1	FORD	www.guardian.co.uk	0.261
1	FORD	rss.cnn.com	0.283
1	FORD	www.nypost.com	0.286
3	FORD	www.bloomberg.com	0.309
2	FORD	www.detnews.com	0.347
1	FORD	www.courier-journal.com	0.349
2	FORD	www.ctv.ca	0.552
1	FORD	www.thestar.com	0.719
1	GOOGLE	www.salon.com	-0.028
5	GOOGLE	online.wsj.com	-0.024
3	GOOGLE	www.msnbc.msn.com	-0.02
1	GOOGLE	rss.cnn.com	-0.018
3	GOOGLE	www.boston.com	-0.014
1	GOOGLE	cgi.money.cnn.com	-0.012
2	GOOGLE	www.networkworld.com	-0.012
2	GOOGLE	www.theregister.co.uk	-0.009
2	GOOGLE	in.reuters.com	-0.002
3	GOOGLE	www.time.com	0.003
6	GOOGLE	www.wired.com	0.004
1	GOOGLE	business.timesonline.co.uk	0.007
2	GOOGLE	www.rollingstone.com	0.007
6	GOOGLE	www.itpro.co.uk	0.008
11	GOOGLE	www.telegraph.co.uk	0.01
7	GOOGLE	abcnews.go.com	0.012
1	GOOGLE	english.chosun.com	0.026
2	GOOGLE	www.newsweek.com	0.028
1	GOOGLE	www.brisbanetimes.com.au	0.031
3	GOOGLE	www.latimes.com	0.037
5	GOOGLE	www.thestreet.com	0.04
7	GOOGLE	www.guardian.co.uk	0.044
11	GOOGLE	www.pcworld.com	0.049
8	GOOGLE	www.forbes.com	0.05
1	GOOGLE	www.gmanews.tv	0.071
6	GOOGLE	www.usatoday.com	0.071
8	GOOGLE	www.reuters.com	0.072
9	GOOGLE	www.marketwatch.com	0.075
2	GOOGLE	sify.com	0.092

6	GOOGLE	www.businessweek.com	0.097
23	GOOGLE	www.informationweek.com	0.102
3	GOOGLE	news.bbc.co.uk	0.119
1	GOOGLE	dir.salon.com	0.125
13	GOOGLE	www.nytimes.com	0.132
1	GOOGLE	business.watoday.com.au	0.142
5	GOOGLE	www.foxnews.com	0.146
5	GOOGLE	www.independent.co.uk	0.161
1	GOOGLE	www.nebraska.tv	0.169
1	GOOGLE	www.eweek.com	0.17
4	GOOGLE	www.sfgate.com	0.189
6	GOOGLE	www.washingtonpost.com	0.292
3	GOOGLE	uk.reuters.com	0.31
3	GOOGLE	ca.reuters.com	0.317
1	GOOGLE	www.techtree.com	0.366
2	GOOGLE	business.theatlantic.com	0.37
1	GOOGLE	www.dailyrecord.co.uk	0.647
1	GOOGLE	googlewatch.eweek.com	0.685
1	GOOGLE	correspondents.theatlantic.com	0.7
1	GOOGLE	www.bloomberg.com	0.888
2	YAHOO	www.brisbanetimes.com.au	-0.111
2	YAHOO	www.msnbc.msn.com	-0.105
4	YAHOO	online.wsj.com	-0.084
1	YAHOO	www.stuff.co.nz	-0.082
1	YAHOO	news.smh.com.au	-0.074
1	YAHOO	www.philly.com	-0.049
2	YAHOO	english.chosun.com	-0.044
2	YAHOO	www.networkworld.com	-0.038
1	YAHOO	www.tehrantimes.com	-0.032
1	YAHOO	www.boston.com	-0.026
1	YAHOO	www.infoworld.com	-0.026
1	YAHOO	www.eastandard.net	-0.024
1	YAHOO	ca.reuters.com	-0.023
2	YAHOO	www.theinquirer.net	-0.021
1	YAHOO	www.zdnetasia.com	-0.02
5	YAHOO	www.itpro.co.uk	-0.017
1	YAHOO	www.buffalonews.com	-0.014
1	YAHOO	www.canada.com	-0.012
3	YAHOO	www.telegraph.co.uk	-0.01
2	YAHOO	dir.salon.com	-0.008
2	YAHOO	rss.cnn.com	-0.003
1	YAHOO	cgi.money.cnn.com	-0.003
1	YAHOO	www.timesonline.co.uk	-0.002
1	YAHOO	sports.espn.go.com	-0.002
8	YAHOO	www.wired.com	-0.001
1	YAHOO	www.dawn.com	0
2	YAHOO	www.internetnews.com	0.001
1	YAHOO	business.timesonline.co.uk	0.002
10	YAHOO	www.thestreet.com	0.004
2	YAHOO	www.theregister.co.uk	0.012
6	YAHOO	in.reuters.com	0.014
1	YAHOO	money.cnn.com	0.018
1	YAHOO	www.watoday.com.au	0.018
1	YAHOO	www.eweek.com	0.028
3	YAHOO	uk.reuters.com	0.03
2	YAHOO	www.independent.co.uk	0.031
6	YAHOO	www.salon.com	0.037

1	YAHOO	www.techtree.com	0.039
17	YAHOO	www.reuters.com	0.059
1	YAHOO	apps.money.cnn.com	0.064
13	YAHOO	www.forbes.com	0.065
7	YAHOO	www.sfgate.com	0.073
10	YAHOO	www.marketwatch.com	0.079
1	YAHOO	www.nydailynews.com	0.08
37	YAHOO	www.washingtonpost.com	0.084
1	YAHOO	sports.sportsillustrated.cnn.com	0.084
11	YAHOO	www.businessweek.com	0.091
16	YAHOO	www.pcworld.com	0.097
4	YAHOO	www.bloomberg.com	0.103
61	YAHOO	news.yahoo.com	0.106
11	YAHOO	www.guardian.co.uk	0.137
6	YAHOO	www.nytimes.com	0.139
1	YAHOO	www.hurriyet.com.tr	0.153
4	YAHOO	www.latimes.com	0.158
9	YAHOO	www.informationweek.com	0.162
1	YAHOO	www.itworld.com	0.166
6	YAHOO	abcnews.go.com	0.194
5	YAHOO	www.foxnews.com	0.208
3	YAHOO	news.cnet.com	0.237
8	YAHOO	sify.com	0.249
1	YAHOO	www.startribune.com	0.26
1	YAHOO	www.theglobeandmail.com	0.277
1	YAHOO	www.bizjournals.com	0.281
3	YAHOO	www.business-standard.com	0.288
1	YAHOO	www.mercurynews.com	0.329
2	YAHOO	sportsillustrated.cnn.com	0.35
3	YAHOO	www.usatoday.com	0.418
1	YAHOO	www.financialpost.com	0.432
1	YAHOO	www.mysinchew.com	0.449
1	YAHOO	www.sportingnews.com	0.463
2	YAHOO	www.nbr.co.nz	0.513
2	YAHOO	www.v3.co.uk	0.513
1	YAHOO	www.etaiwannews.com	0.688
1	YAHOO	www.newsweek.com	0.695
1	YAHOO	www.cctv.com	0.759