Tweeting the Odds: Using Semantic Analysis of Twitter to Predict the Vegas Line for NFL games

Samuel Clark
sclark2@swarthmore.edu

David Opoku
dopoku1@swarthmore.edu

Abstract

Data from social networking sites is expanding rapidly and furthermore becoming more and more applicable to research. In this project we used data from Twitter to predict the Vegas odds for each NFL game in a given week. We collected streaming data from Twitter to the NFL for four weeks, a total of approximately 475 thousand tweets. To analyze this data, we combined several sentiment analysis algorithms and sentiment corpora. Our most accurate method—using sentiment count, negation, and outcome-sepecific weighting—surpassed random guessing in predicting the spread for each game by a 70 percent margin with 100 percent being perfect. With only three weeks of data available it is hard to be sure that our research is significant, however, all indicators suggest that we can accurately predict the favorite and spread for NFL games using sentiment analysis on data from Twitter.

1 Introduction

Trying to predict the outcome of NFL games is a multimillion dollar industry. Each week high profile television networks constantly broadcast analysts' thoughts and predictions on NFL (National Football League games. Odds-makers use complex statistical formulas to pick the favorite and spread for each week's games. We asked the question: can we use data from Twitter to predict these parameters?

Research has shown that Twitter, a rapidly expanding social networking site, can provide a vast data-set of public opinion on any number of topics.(O'Connor et al., 2010) We are generally interested in using the sentiment in tweets related to sports to make predictions. Our research–predicting the Vegas odds and spread of NFL games using sentiment analysis–poses a new challenge from simply predicting public opinion.

Surely there is an association between a fan's happiness with his or her team and the team's performance. But unlike an election, which is often a binomial event purely based upon public opinion, a football game manifests into a complicated entity—we can neither control nor influence it. No one can exactly predict the outcome of a sporting event. This uncertainty is what makes sports so unique, so lovable, and to some people, so enticing to bet on. We want to know this, though: can tweets from each week collectively predict what Vegas thinks will happen in an NFL game?

In this paper,we use several sentiment analysis algorithms to see if we can predict the odds of a match up between two NFL teams in a given week. These algorithms include methods found in similar projects and also our own analysis techniques. Our research culminates in a combination of algorithms which we feel can accurately accomplish our task. Along the way we will examine why some techniques work, why some do not, and what Twitter can tell us about the chances of a team's success.

2 Data Collection

2.1 Week 10 Data Retention

Our initial method to collect data was to collect the 1500 most recent tweets for each NFL team (32 teams) every hour for the six days prior to game-day and to store them in a text file.We cleaned this data by eliminating duplicates and ended up with around 120,000 total tweets but only 63,000 unique tweets that matched our search query. We used this method for week 10 but then changed to a more effective, accurate, and comprehensive streaming method.

2.2 Streaming Data from Twitter

The streaming method constantly streamed in queries of all 32 teams and stored them in one file. An advantage of this method is that we were able to store each tweet in Java Script Object Notation (JSON.) Storing the data in this way allows us to access many telling aspects of the tweet such as the user who tweeted it, where it was tweeted from (geo code), when it was tweeted, whether or not it was retweeted, and most relevant to our research: the text of the tweet.

One challenge we came across was sorting the streaming data-file into individual files for each team. We realized, however, that we could accurately classify which team each tweet belonged to by using the teams nickname (i.e. we could search for 'Giants' to find 'New York Giants'.) We ended up with approximately 113,000 unique tweets for week 11; 121,000 unique tweets for week 12; and 90,000 tweets for a deadline-shortened week 13. We sorted these tweets into 32 individual tweet files for each team each week to analyze. There was about an average of 3.5 thousand tweets for each team on a given week.

2.3 Sentiment Corpora

For our sentiment corpus we initially and primarily used the list of around 6,200 positive and negative sentiment words compiled by Wiebe and colleagues at the University of Pittsburgh(Wiebe et al., 2005).We used only the word and the sentiment classification, ignoring their part-of-speech tags. We chose to ignore words classified as 'neutral', using only the positive and negative words.

We also manually generated a list of about 50 positive and 50 negative words (100 total) that we found relevant to team success in football, fans' reaction or feelings about their team, and important components of success in the sport itself such as ('touchdown' or 'fumble'). This corpus was generated without looking at our data for trends or impact full words.

3 Analysis Methods

3.1 Pure sentiment count

Influenced by the success of (O'Connor et al., 2010) in using sentiment count as a lone parameter to predict the outcomes of political polls and public opinion, we initially tried to determine a team's likelihood of winning by a simple sentiment count. Using the sentiment list compiled by Jan Wiebe and collogues, we iterated through each team's tweets for a given week. Within each tweet we checked a positive sentiment dictionary and negative sentiment dictionary for the current word. If the word was positive, we incremented a team's sentiment count. If the word was negative we decreased a team's sentiment count. A team's sentiment score was the sentiment count normalized by the number of tweets that team received in the given week. Thus the final sentiment score for a team came out to be the proportion of tweets the team received that were positive, or in a small number of cases, negative. Just as the research predicting the outcome of polls and elections,(O'Connor et al., 2010), within this count-only method we did not consider negation or weighting the sentiment words by frequency(a technique called term frequency-inverse document frequency).

For this method of scoring, the formula for a team's sentiment was: **(#positive sentiment words seen - #negative sentiments words seen)**

A team's sentiment score was normalized by the number of tweets the team actually had: **#pos sentiment -# neg sentiment/Total number of team tweets**

Normalizing by the number of tweets was a technique we used in all our methods. Without considering this factor, it would be impossible to score the tweets fairly as some teams (such as the Dallas Cowboys) receive as much as 4x as many tweets each week as other teams (such as the Buffalo Bills.)

3.2 Scoring with tf-idf sentiment weighting

Our initial method of scoring gave the same word value to all sentiment words in the corpus, however, it is some times niave to assume equal weight for all sentiment. For example, a word such as ‘good’ in the sentiment dictionary which is so common and thus may appear more frequently in tweets should be weighted differently than a word such as ‘ingenious’ which is less likely to appear in a tweet. In order to determine these weights, we used the tf-idf weighting method commonly used by search engines to prioritize user queries.(Ramos, 2001)

This scoring method makes use of two different terms to weight words in a document.TF stands for the term(word) frequency, which is how many times a given term or word appears in a single in a document whereas IDF stands for inverse document frequency which is how many documents the term or word appears in. TF is calculated by dividing the frequency of the term by the number of words in the document as thus gives a high score to the most frequent terms in a document. On the other hand, IDF is calculated by dividing the total number of documents by the number of documents the term appears in, and then taking the logarithm of that. This way, TF-IDF ensure that a term appearing frequently in one document will only have a high score if it also appears in other documents also.

A single document represented all the tweets for each team i.e. all the tweets for the New York Giants from a given week represented a single document. The total number of documents thus was 32

tf-idf=Term frequency$*$log(Total#documents/ Total # of document term found in)

3.3 Scoring with tf-idf sentiment weighting

This method combines the tf-idf method with negation step and selected weighting described below. We use identical negation and weighted lists for this algorithm but this time with our sentiment words have a value or weight based on the tf-idf weighting formula.

3.4 Sentiment count,negation and word-specific weighting

Evaluating a team’s success by count only was an incomplete method given our task. One critical flaw with this method is that it does not consider a common phenomenon in sentiment structure—negation. An example of this would be if a sentence read ‘The New England Patriots are not good’. In this sentence the writer is expressing a negative sentiment towards the Patriots. A count-only analysis method, however, would score the sentence as positive.

To correct for this, we created a list of words commonly seen before positive or negative sentiment words that reverse the meaning of the sentence. This included **‘not, aren’t, can’t, don’t, won’t, cannot’** amongst others. Before we considered each sentiment word that occurred within a tweet, we checked to see if the word before it was in the negation list. Rather than weighting a negation + positive as a negative and a negation + negative as a positive, we felt that using the negation grammar structure implied weakness in opinion. Thus pure positive and negative sentiment counted as 1 full point towards or against the sentiment score, while a sentiment word preceded by a negation word only earned .75 points. This provided a nice balance between discounting negated sentiment and treating it equal to the converse. We did not experiment with this scaled value—an action that would have taken away from the unsupervised nature of this experiment.

3.5 Outcome Specific Weighting

We wanted to see if we could use sentiment to predict the Vegas favorite and furthermore if there was an association between the ratio of sentiment scores and the Vegas spread. We hypothesized accordingly that words directly related to the outcome of the game, **‘win, victory, lose, loss’**, should carry additional weight. In this scoring method, these words were weighted with twice the importance of other sentiment words.

3.6 Evaluating the sentiment scores and algorithms

We had many methods to create a sentiment score for each team. The difficulty was deciding how to measure our algorithm once we had that score. Since in the NFL, two teams compete against each other each week, we decided to take the ratio of the sentiment scores between the Vegas favorite and the Vegas underdog. We ran into a roadblock, though, when one team had a negative score(a score indicating that they had more negative sentiment than positive sentiment.) The literal ratio between the two sentiment scores would not take into account for this negative possibility.

To fix this we used two as a base and raised it to the sentiment score of each team. Then we took the logarithm of this ratio to normalize the result against zero. If the result was greater than zero it means we correctly predicted the Vegas favorite to win and we earned the related spread points. If the result was negative it means that we predicted the Underdog to win and thus lost the spread points. The formula below describes our scoring method:

For each match up where one team is the favorite and the other is the underdog where the spread predicts the margin of victory for the favorite

result $= \mathbf{log}(2^{Favorite}/2^{Underdog})$

if result$>$ 0: increment points by Vegas spread for match up

if result$<$ 0: decrement points by Vegas spread for match up

This method of scoring emphasizes the importance of the relationship between the ratio of sentiment scores and the Vegas spread. If the spread predicts one team winning by a large margin and our algorithm incorrectly predicts that the underdog is going to win, we are penalized sharply. Similarly in a game in which the spread is very small (i.e. Vegas thinks either team could win) we do not lose very much for predicting that game incorrectly. The graph created by these data points provides a nice visual, with points to the right of zero being correct predictions and points to the left of zero being wrong. Additionally, the further a point is from zero on the x-axis, the more confident the twitter sentiment is of the result. See Figure 1 and Figure 2 for more detail.

4 Results

The two most important indicators of success were the percent of Vegas favorites the specific algorithm predicted correctly and more significantly the proportion of spread points earned to the total number of spread points available. As explained in the scoring method, the percentage of points we earned is how much better our algorithm is at predicting the Vegas spread than blind guessing would.

The guessing baseline for favorites guessed correctly is fifty percent. The baseline for proportion of points earned is 0.0. These categories are both important because they first show if there is a relationship between a team's sentiment on twitter and the Vegas favorite. More importantly, the equation we used to determine results allowed us to look for a correlation between the difference in sentiment scores (the ratio of the Vegas favorite to the Vegas underdog) and spread. This weighted games that Vegas predicted to be 'blowouts' with much higher value such that if we predicted them wrong, our results would suffer immensely.

The table below (or at the end of the paper if formatting pushes it there) shows the results for each algorithm for each week at two different margins of significance. The margin of significance measures how sure we have to be in our prediction of a game to have it count. For our two most effective algorithms, we also include results from running them on the manually created sentiment corpus. When looking at the data, the proportion of points earned is the percentage we do better than a series of random guesses would; points earned is how many spread points the algorithm earned; and percentage of favorites guessed right is the number of favorites our algorithm picked. This table in order contains the Week, margin of significance, analysis method, proportion of spread points earned (our measure of success), percentage of Vegas favorites guessed, and the sentiment corpus used. The table is sorted in order of rating–that is the closer to the top of the table is, the better the algorithm performed. If the table does not show up directly below it will be located on the last page of the paper.

4.1 Most Effective Algorithms

Our best algorithm on both the Wiebe sentiment corpus and our manual corpus was the scoring method which took into account sentiment count, negation, and word-specific weighting. On the streaming data-set, a far more comprehensive data-set from week 10, the algorithm performed at an average proportion of points earned of 0.61. This indicates that, not only did it gain 60% of the overall spread points

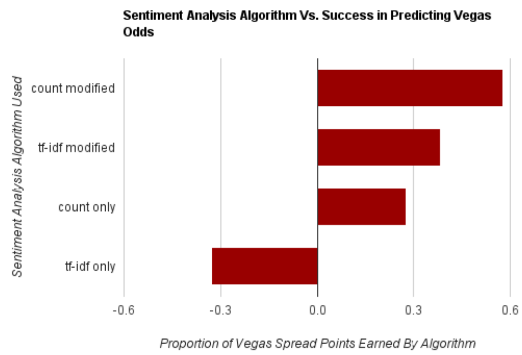

Figure 1: This graph shows the average predictive ability of each sentiment analysis algorithm we used. As illustrated, the cont + negation + weighting predicted the greatest proportion of spread points over a three week period while tf-idf predicted the least. The baseline guess on this graph would be a bar neither to the positive or negative but centered at 0

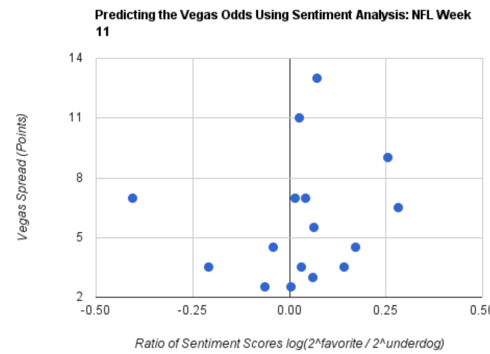

Figure 2: This graph shows the results of our count + negation + weighting algorithm on week 11 data using the Wiebe sentiment corpus. We predicted the positive data points correctly and the negative data points incorrectly

from weeks 11, 12 and 13 but it averaged performing 60% better each week than random guessing would. It had maximum proportion of points earned of 0.78 and a minimum of 0.60.

The next best algorithm was the tf-idf + negation + weighted word frequency with an average proportion of points earned of .384. The algorithm performed extremely well on the manual corpus achieving a maximum proportion of 0.80, however it dipped to a minimum of -0.20 using Wiebe's corpus. The most interesting part of this algorithm was the tremendous swing between results using the same Twitter results (week 12) and different sentiment corpora. This algorithm analyzed week 12 with an efficiency of -0.06 using Wiebe and reached 0.80 using the manually created corpus. Interestingly, Tf-idf alone (without negation and word-specific weighting) performed the worst across the board.

5 Discussion

5.1 Week 10 data

Our method of collecting data for week 10 produced a lot of duplicates and furthermore we could not be exactly sure where and when the data was coming from. This uncertainty combined with the smaller corpus size around 60,000 proved worrisome. Af-

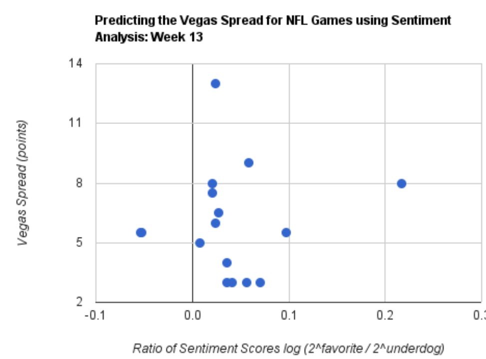

Figure 3: This graph shows the results of our count + negation + weighting algorithm on week 13 data using the manually created sentiment corpus. We predicted the positive data points correctly and the negative data points incorrectly. Comparing this graph to the graph from week 11 we can see that using the same method and a difference corpus our accuracy improved significantly

ter examining our data, we can not only say that our methods did significantly worse using week 10 (not a valid reason to toss them out), but that they demonstrated an equality to guessing.

Taking this into account, and contrasting the quality our results from week 10 with our results from weeks 11-13, we feel that week 10 was not a good data set to run our analysis on. We could not precisely pin point the content of the data and there was not enough of it for semantic analysis to be effective.

5.2 Wiebe Sentiment Corpus

We ran the majority of our tests using the sentiment corpus provided by (Wiebe et al., 2005). After looking at our results, we asked what made it so ineffective in comparison to our manually created corpus less than a 60th of the size. We then considered the data we were analyzing. We were analyzing tweets about football: 140 character lines—approximately 11 words each—about a sports team. Further more these tweets are often quick thoughts from the average American, not well drawn out ideas. The expansive vocabulary covered in the large sentiment corpus is widely irrelevant to our twitter data. Of the words that did occur, there were a plethora of words(around 500), which only occurred once. When considering general public sentiment, a singleton sentiment word should represent a red flag.

This disproportionate representation was one problem, but a bigger problem was that the Wiebe corpus was missing so many words relevant to success in football. 'Touchdown, win, won, fumble, turnover, field-goal' just to name a few. This difference may have resulted in the inaccurate calculation of total sentiment of a tweet and thus affecting the accurate determination of a team sentiment. We aimed to correct these errors with a smaller and more relevant sentiment corpus.

5.3 Count only

The count only algorithm performed decently, posting an accumulative proportion of points earned average of 0.35 (excluding week 10). While this is still a decent result, it falls short of its modified version. The most important knowledge we can derive from this is that negation is crucial in accurate prediction of sentiment in this dataset. Additionally we can conclude that weighting words which are directly related to the outcome of the game has a positive effect. Comparing this to the results of (O'Connor et al., 2010), in which they did not consider negation nor weighting, we can draw several conclusions.

The purpose of our two respective experiments was similar but the subject matter was drastically different. They were trying to determine the outcome of political polls. A political poll is a direct measure of public opinion. Our research on the other hand, considers the outcome of an NFL game. This explains why a simple sentiment count scoring mechanism worked so effectively for their work but in many ways falls short within our context.

5.4 The tf-idf methods

Our results show that the tf-idf algorithm performed terribly in predicting the Vegas odds when the Wiebe sentiment corpus was used, but contrasted that with a significant improvement and prediction with the manual sentiment corpus. With the Wiebe corpus(Wiebe et al., 2005), the best performance was seen for week 11, with the algorithm that combined tf-idf,negation and selected sentiment weights posting a proportion of 0.63 points earned as compared to 0.22 points for the tf-idf only algorithm. For the manual sentiment however, the best results was observed in week 12, where the improved tf-idf algorithm posted a proportion of 0.80 points earned. For the same week 12 data, the tf-idf only algorithm posted a proportion of points earned at -0.06 using the Wiebe sentiment corpus(our worst performance). Generally, we see that our tf-idf only algorithm performed worse than the improved tf-idf algorithm using the Wiebe corpus. Using a margin of 0 on the week 12 data, the tf-idf only algorithm posted a proportion of points earned of -0.39 compared to -0.2 for the improved tf-idf algorithm.

We can see that the poor performance of the tf-idf algorithm is not just from the structure of implementation of this algorithm but the combined effect of the sentiment corpus used. With a large sentiment corpus, in which many of the sentiment words are not related or present in the tweets, more weight is given to words rarely tweeted words which leads to low weighting for commonly used tweet words. This may eventually reduce the total sentiment of a tweet, leading to inaccurate categorization of a tweet.

5.5 Count, negation and selective-word weighting

We modified the count only to take sentiment negation and several words directly related to the win/loss outcome of an NFL match up. Negation turned out playing a less significant roll than we expected. As we can see from table 2.0, the effect of negation, while present, was minimal compared to the effect of weighting direct win loss sentiment. The two words used for classifying winning sentiment were ‘win’ and ‘victory’. For classifying losing the words were ‘lose’ and ‘lost’.

The table shows us that weighting the direct-outcome words had a significant effect on the data. Considering our count only algorithm had an average of 0.35 proportion of points earned and our count,negation and selective-word weighting had an average of 0.64 proportion of points earned and scored above 0.70 in all three weeks(11,12, and 13) at a significance margin of 0.

This method of analysis performs extremely well within this data set. We are beating a random guess by a solid margin of 70% (with the maximum being 100 if you actually had the Vegas spreads in front of you.) We could in fact do better with this algorithm if we used a machine learning algorithm such as a Naive Bayes classifier or a hidden Markov model to train the idea weight values and negation words to match NFL data specifically.

The weighting in this algorithm illustrates that if a large number of people tweet that a team is going to win, the collective tweets as a whole do a good job predicting what Vegas is going to set the line at. A less impressive measure of success, the percentage of favorites the algorithm detected, was a high 76%. This value is quite good considering a baseline of 50%.

5.6 Irrationality of Sports Fans

We cannot claim a direct relationship between the ratio in sentiment scores between team’s and the Vegas spread because the plots of our results did not have a significant r (correlation) value. If there Does this mean we got lucky? No. While ideally we would like to see a consistent association between sentiment and spread, we need to remember what our data is. We are not working with discrete statistics. We are not supervising our analysis so it is very difficult to control lurking variables. Our data consists of hundreds of thousands of Tweets from generally emotional, irrational, and bias sports fans who are likely to support their team regardless of outcome. If fans were actually rational about predicting the results of their team, the betting industry would likely fall to pieces, as would interest in professional sports.

Our algorithm actually shows the irrationality of fans, with a significant difference in positive and negative sentiment despite an equal number of positive and negative words. Generally twice as many tweets contained ‘win’ or ‘victory’ than ‘loss’ or ‘lose’. This makes sense: rarely is a fan going to Tweet about their team losing in an upcoming game. This phenomenon is also present in the minimal number of negative sentiment scores, an average of approximately 4/32 teams had a negative sentiment score each week while 16/32 teams are going to lose each and every week.

Despite this lack of rational opinion we were still able to glean predictive sentiment from the Tweets. By tweaking the weighting of the most important factor, whether a fan thinks their team is going to win or not, we were able to enhance the accuracy of our prediction in both the count-only and tf-idf only algorithms.

6 Conclusion and Confidence in our Results

With a data set of three reliable weeks of NFL tweets, it is hard to know exactly how much confidence we can place in our results as a general method of predicting the Vegas odds and spread of a match-up between two NFL teams. If we do, however, choose to trust that we did not just get significantly lucky for three weeks in a row, we claim that sentiment from Twitter can indeed predict the Vegas spread in an efficient and accurate manner. With our best method, using sentiment count, negation, and win-specific sentiment weighting, we predicted above 0.70 proportion of spread points possible for three straight weeks. Considering that a human just guessing randomly would average at the baseline of 0.0, it is safe to say that our algorithm does have value.

It is important to note, however, that we neither looked at the data for which sentiment words would be important nor which win/loss specific words would be effective nor which negation/sentiment weights would be the most effective. As mentioned previously, we believe that if we did use machine learning to train this data set with more effective parameters, this algorithm would gain strength and accuracy.

7 Future work

7.1 Testing With More Data

This project can be expanded in many directions. The first and perhaps most important work in the future is to get data from more NFL weeks so that we can get a large sample size to determine the significance of results. We will pursue this by collecting data through the end of the season and into the playoffs.

7.2 Using Machine Learning to Tweak our Algorithms

From an algorithmic standpoint, there are many things we can do to improve our work. As mentioned in the paper we can use machine learning to find the best possible parameters to use for weighting. We assumed that the logical but theorized values of 2 and .75 for outcome-related sentiment and negation respectively are not the perfect values. On the same note, our manually created sentiment corpus was generated in a 10 minute spree of thinking about what words are most important and additionally likely to be tweeted relative to NFL success. We did not check the frequency of these words which would have made this a supervised project.It would be interesting, however, to use multiple weeks as a training set to create the most effective sentiment corpus possible and then to test it on future data-sets. This would surely allow for a more accurate prediction of the Vegas odds as we would be targeting all the essential sentiment words.

7.3 Expanding Our Research to Other Sports

It would also be interesting to run our algorithms on data sets from different sports–although the data would have to predict success over time rather than game by game as football is really the only American sport with full weeks before games. I would predict that the algorithm word work well for baseball and basketball and even soccer.

8 Acknowledgements

Many thanks to Rich Wicentowski for all the help and time he put into advising us with this project. Also much thanks to John Gluck and Jeff Knerr for helping us set up the streaming query with the Twitter api.

References

Casey Fiesler. 2010. June. Twitter Sentiment Analysis: How Do We Feel About the iPhone?

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Internantional AAAI Conference on Weblogs and Social Media*.

Matt O'Hern. 2009. Why Google really wants Twitter: Real time sentiment analysis scoring.

Juan Ramos. 2001. Using tf-idf to determine word relevance in document queries.

J. Wiebe, T. Wilson, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analyusis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Table 1: Segmentation Method Results

Week	Margin of Significance	Analysis Method	Pct Spread Pts Earned	Favorites Guessed Pct	Total Points	Sentiment
12	0.01	T+N+W	0.8	85	47	manual
12	0	C+N+W	0.78	81	57	manual
13	0	C+N+W	0.77	88	73	manual
12	0.01	C+N+W	0.74	79	45	manual
12	0	T+N+W	0.73	75	53	manual
13	0.01	C+N+W	0.72	86	55	manual
11	0	C+N+W	0.7	75	65	manual
11	0.01	C+N+W	0.68	73	60	manual
11	0.01	C	0.67	80	61	wiebe
11	0.01	C+N+W	0.66	73	56	wiebe
11	0	T+N+W	0.63	75	68	wiebe
11	0	C+N+W	0.63	75	58	wiebe
11	0	C	0.63	75	58	wiebe
11	0.01	T+N+W	0.62	73	56	wiebe
11	0.01	T+N+W	0.53	67	47	manual
11	0	T+N+W	0.44	63	41	manual
12	0.01	C+N+W	0.36	67	25	wiebe
12	0	C+N+W	0.3	63	22	wiebe
11	0.01	T	0.22	58	13	wiebe
11	0	T	0.12	56	11	wiebe
12	0.01	C	0.09	54	6	wiebe
12	0	C	0.02	50	1	wiebe
10	0	C+N+W	0.01	64	1	wiebe
10	0	T+N+W	-0.03	57	-1	wiebe
10	0	C	-0.03	57	-1	wiebe
12	0.01	T+N+W	-0.06	57	-2	wiebe
10	0	C+N+W	-0.19	64	-9	manual
12	0	T+N+W	-0.2	50	-14	wiebe
12	0	T	-0.39	44	-28	wiebe
10	0	T	-0.59	28	-30	wiebe
12	0.01	T	-1	0	-19	wiebe

Table 2: Average distribution of sentiment negation and win-loss weighting

Negation	Weighting	Tweets	Weeks	Collection
417	9528	108000	3	Streaming