SentiHeros at SemEval-2017 Task 5: An application of Sentiment Analysis on Financial Tweets

Narges Tabari, Armin Seyeditabari, Wlodek Zadrozny
Narges Tabari: nseyedit@uncc.edu
Armin Seyeditabari: sseyedi1@uncc.edu
Wlodek Zadrozny: wzadrozn@uncc.edu

Abstract

Sentiment analysis is the process of identifying the opinion expressed in text. Recently it has been used to study behavioral finance, and in particular the effect of opinions and emotions on economic or financial decisions. SemEval-2017 task 5 focuses on the financial market as the domain for sentiment analysis of text; specifically, task 5, subtask 1 focuses on financial tweets about stock symbols. In this paper, we describe a machine learning classifier for binary classification of financial tweets. We used natural language processing techniques and the random forest algorithm to train our model, and tuned it for the training dataset of Task 5, subtask 1. Our system achieves the 7th rank on the leaderboard of the task.

1 Introduction

The recent explosion of textual data creates an unprecedented opportunity for investigating people's emotions and opinions, and for understanding human behavior. Although there are several methods to do this, sentiment analysis is an especially effective method of text categorization that assigns emotions to text (positive, negative, neutral, etc.). Sentiment analysis methods have been used widely on blogs, news, documents and microblogging platforms such as Twitter.

Although social media and blogging are popular and widely used platforms to discuss many different topics, they are challenging to analyze. This is to large extent due to the specific of vocabulary and syntax, which are dependent on topics, with the same words possibly expressing different sentiments in different contexts. For example, a word in a casual context might have positive or neutral sentiment (e.g., crush), while the same word generally has a negative sentiment in finance. Therefore, with the absence of general natural language understanding, context-dependent and domain-specific approaches allow us to increase the accuracy of sentiment analysis at a relatively low implementation cost.

Domain-specific sentiment analysis is being used to analyze or investigate various areas in finance, such as corporate finance and financial markets, investment and banking, asset and derivative pricing. Ultimately, the goal is to understand the impact of social media and news on financial markets and to predict the future prices of assets and stocks.

The proposed task in SemEval-2017 targets a sentiment analysis task, which we should identify a range of negative to positive affect on the stock of certain companies. The objective of the task was to predict the sentiment associated with companies and stock with floating point values in the interval from -1 to 1.

Previous research on textual analysis in a financial context has primarily relied on the use of bag of words methods, to measure tone (Tetlock, 2007) (Loughran & McDonald, 2011) which is one of the prominent efforts to improve sentiment analysis in financial domain, showed that using non-financial word lists for sentiment analysis will produce misclassifications and misleading results. To illustrate this, they used the Harvard-IV-4 list on financial reports, and found that 73.8% of the negative word counts were attributable to words that were not actually negative in a financial context.

Recently, there has been an increasing interest towards the use of machine learning techniques to get better sentiment result; e.g., naïve Bayesian classifier (Saif, He and Alani 2012) with various features got the accuracy of 83.90%. Other reported results include the use of support vector machines (SVMs) with the accuracy of 59.4% (O'Hare et al., 2009), and multiple-classifier

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 857–860,
Vancouver, Canada, August 3 - 4, 2017.

voting systems with the 72% accuracy (Das & Chen, 2007).

In this paper, we describe our approach to building a supervised classifier predicting the sentiment scores of financial tweets provided by SemEval-2017. The classifier is fed preprocessed tweets as input and it predicts the binary labels of the tweets. Once tweets were preprocess and features were extracted, various classification models were applied using Weka tool (Hall et al., 2009). This environment contains a collection of machine learning-based algorithms for data mining tasks, such as, classification, regression, clustering, association rules, and visualization. We ultimately used Random Forest as our classifier as in our various tests it showed the best and accuracy in classifying the tweets. After predicting the binary labels, we then use the probability of the tweets being correctly classified to create a range of predictions from -1 to 1 as it was requested in the task.

2 Method

2.1 Preprocessing the data

SemEval task 5, subtask 1 provided a training dataset with 1800 tweets. Every tweet had a sentiment score between -1 to 1 and it showed its sentiment toward the stock symbol that was assigned to that tweet. Table 1 describes variables in the training dataset we used for analyzing the tweets:

Label	Description
ID	Each tweet was assigned a unique ID
Span	Part of tweet that was considered to carry the sentiment toward the company or stock.
Sentiment	Score provided to us with numbers between -1 to 1.
Cashtag	Stock symbol that was the target of each tweet, e.g. $GE.

Table 1. Attributes used to create the sentiment classification model.

To prepare the dataset for classification, we first converted the sentiment scores to -1, 0 and 1. Tweets with sentiments between -0.01 and 0.01 were labeled as zero, positive sentiments labeled as 1 and negative tweets were labeled as -1. We then disregarded the tweets with neutral sentiment, which left us 1560 tweets to train our model. Some tweets had multiple Spans, describing the sentiment toward the Cashtag. To keep things simple, we concatenated the spans of each tweet with each other. Then using the Python NLTK[1] library we deleted the punctuations, tokenized the spans, and deleted the stop words.

Since certain stop words in financial context can have impact on the sentiment of the tweets, we excluded them from the stop word list. Words like "up", and "down" were not removed from tweets. We also removed the negations from the stop word lists, as we later handle the negations on our own when creating the features.

2.2 Feature Selection Process

To add features to our training dataset, we used the McDonald's wordlist (Loughran & McDonald, 2011). This is a list of positive and negative words for financial 10-K reports containing the summary of the company's performance.

We calculated number of positive or negative words in each Span, using the McDonald's wordlist in the added features. There were some words, such as "short" which was not in any wordlist as a negative word, yet shorting a stock expresses a negative sentiment toward that stock. For this reason, we manually added positive or negative words to each list that to our best knowledge carry those sentiments. Table 2 shows some of the words were added to McDonald's wordlist:

Word	Sentiment
Profit	Positive
Long	Positive
Short	Negative
Decay	Negative

Table 2. Example of the words added to McDonald's wordlist. (See full list in Appendix A)

Adding these words to the wordlist improved our results. Then we realized in context of finance, co-occurrence of some words with each other in one tweet changes the sentiment of the tweet completely. For example, "short" and "sell" are both negative words in context of finance, but selling a short contains a positive sentiment in stock market context. Another example would be the co-occurrence of "go" and "down", or "pull" and "back" in our tweets. In a similar fashion we

[1] http://www.nltk.org/

also we handled the negations. Once we found these patterns, we normalized our data, i.e. we replaced the combinations of words in the tweet with a single positive or negative label, which we treated just as another positive or negative word. We then re-counted the number of positive or negative words in the tweet and updated our feature vectors. Table 3 shows examples of patterns we found in the tweet to have changed the sentiment of the word. The normalization had a benefit of increasing the counts of rarely occurring ex

Word 1	Word 2	Replaced with
Go	Up	OKAY
Go	Down	NOTOKAY
Sell	Short	OKAY
Pull	Back	NOTOKAY

Table 3. Example of the word couples and their replacements used to normalize the data (tweets). (See full list in Appendix B.)

2.3 Sentiment Prediction

Classifier	Accuracy	F-score	Precision	Recall
Random Forest	91.26%	86.5%	91.3%	82.2%
SVM	90.43%	85.4%	88.9%	82.2%
Logistic Regression	84.69%	79%	74.3%	84.3%
Naïve Bayes	83.73%	73.3%	83.3%	65.4%

Table 4. Results of different Weka classifiers using 10-fold cross validation and default settings.

After pre-processing our data and creating all our features (Tweet, Positive-Count, Negative-Count), we used WEKA to classify our tweets. Our feature vectors were the combination of document vectors generated by Weka's StringToWordVector filter, followed by the features extracted from the data as explained above. Among all the classification methods that we used, Random Forest did give us the best result with accuracy of 91.2%. Table 4 shows results from various classifiers using our training data. The random forest model in WEKA provided both a class prediction and class probability for each tweet in the training and test set.

Since the final float score needed to be between -1 and 1, for tweets classified as negative we made the sentiment score the negative of the class probability; for positive classifications, the sentiment score was simply the class probability.

2.4 Other Experiments

We have done several other experiments first to find a promising approach, and to gauge alternative methods of classification and data preprocessing.

In our initial experiment, after pre-processing the tweets, we first ran the tweets on WEKA to classify using only the feature vector, WEKA's StringToWordVector which is a term document matrix. Random forest and Logistic regression had the highest accuracy of 83.3% and 85.3% respectively. This experiment shows the impact of our additional features to be around 6%.

Before deciding on the final features of the model, we tried other types of features. Although many of them did not improve the model, we still thought they were worth mentioning, with description of them following:

Bigrams: In the first experiment, bigrams were used. (Kouloumpis, Wilson, & Moore, 2011) showed that using unigrams and bigrams are effective in improving sentiment analysis. (Dave et al., 2003) reported that bigrams and trigrams worked better than unigrams for polarity classification of product reviews. Unfortunately, bigrams reduced accuracy of Random Forest and Logistic regression to 76.7% and 73.9% respectively. We imagine that with a larger data set, bigrams might be valuable.

Feature selection using logistic regression: In another experiment, we used logistic regression to produce a list of words with the higher odds ratio. We then removed other words from tweets, in an attempt to amplify the stronger signals. However, applying filtered tweets, with various ranges of odds ratio did not help with improving the results. The best result was when words only with odds ratio of [-5, 5] stayed in our training set; this gave us the accuracy of 83.5%.

Using word embedding (GloVe vectors): GloVe vectors (Pennington, Socher, & Manning, 2014) are vector representations of the words. In two separate experiments, we used vectors based on the Common Crawl (840B tokens, 2.2M vocab, cased, 300 dimensions), and the pre-trained word vectors for Twitter (2B tweets, 27B tokens, 1.2M vocab, 200 dimensions). We represented every word in each tweet by a corresponding vector. We then calculated the tweet vector, using the mean of word vectors of the tweet. In this experiment, McDonald's (Loughran & McDonald, 2011) positive and negative wordlist again were used. That is, we created a positive and negative vector using words in those lists. Comparing the cosine similarity of tweet vectors with positive and negative vector, we classified the tweets. The accuracy of this method was 72% and 73.8% for tweet and common crawl respectively.

3 Conclusion

The purpose of this paper was to create a classification method for SemEval-2017 task 5, subtask 1. In our approach after pre-processing the data, negation handling, and feature selection approaches, we used Weka to classify our data using Random Forest algorithm. Our classifier was ranked 7th and achieved accuracy of 91.26%.

In the next step, we think it is important to capture more complex linguistic structure, irony, idioms, and poorly structured sentences in financial domain. To this regard, we would like to apply dependency parser trees for tweets to see if that would improve our results; it might also be necessary to capture some of the idiomatic constructions in this domain.

Also, SemEval-2017 training dataset was a relatively small dataset, which would prevent us from implementing any neural network models for prediction. Therefore, we think a step to create a better model is to increase the size of training dataset.

References

Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. *Management Science*, *53*(9), 1375–1388. http://doi.org/10.1287/mnsc.1070.0704

Dave, K., Lawrence, S. & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of the 12th International Conference on World Wide Web*, 519–528. http://doi.org/10.1145/775152.775226

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg! *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 11)*, 538–541. Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2857/3251?iframe=true&width=90%25&height=90%25

Loughran, T. I. M., & McDonald, B. (2011). When is a Liability not a Liability? Textual Analysis , Dictionaries , and 10-Ks. Journal of Finance, 66(1).

O'Hare, N., Davy, M., Bermingham, A., Ferguson, P., Sheridan, P. P., Gurrin, C., … OHare, N. (2009). Topic-Dependent Sentiment Analysis of Financial Blogs. *International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, 9–16. http://doi.org/10.1145/1651461.1651464

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543. http://doi.org/10.3115/v1/D14-1162

Saif, H., He, Y., & Alani, H. (2012). Semantic sentiment analysis of twitter. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7649 LNCS*(PART 1), 508–524. http://doi.org/10.1007/978-3-642-35176-1-32

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, *62*(3), 1139–1168. http://doi.org/10.1111/j.1540-6261.2007.01232.x

Appendix A. Words Added to McDonald's Wordlist.

Negative words: cult, brutal, fucked, suck, decay, bubble, bounce, bounced, low, lower, selloff, disgust, meltdown, downtrend, bullshit, shit, breakup, dropping, cry, dumped, torture, short, shorts, shorting, fall, falling, sell, selling, sells, bearish, slipping, slip, sink, sinked, sinking, pain, shortput, nervous, damn, downtrends, censored, toppy, scam, censor, garbage, risk, steal, retreat, retreats, sad, dirt, flush, dump, plunge, crush, crushed, crying, unhappy, drop, broke, overbought.

Positive words: epic, highs, recover, profit, long, upside, love, interesting, loved, dip, dipping, secure, longs, longput, rise, able, buy, buying.

Appendix B. Full List of Word Couples to Detect the Semantic of a Tweet.

Positive word couples: (go, up), (short, trap), (exit, short), (sell, exhaust), (didnt, stop), (short, cover), (close, short), (short, break), (cant, risk), (not, sell), (dont, fall), (sold, call), (dont, short), (exit, bankruptcy), (not, bad), (short, nervous), (dont, underestimate), (not, slowdown), (aint, bad).

Negative word couples: (high, down), (lipstick, pig), (doesnt, well), (bounce, buy), (isnt, cheap), (fear, sell), (cant, down), (not, good), (wont, buy), (dont, trade), (buy, back), (didnt, like), (profit, exit), (go, down), (not, guaranteed), (not, profitable), (doesn't, upward), (not, dip), (pull, back), (not, optimistic).