INF-UFRGS at SemEval-2017 Task 5: A Supervised Identification of Sentiment Score in Tweets and Headlines

Tiago Zini and **Marcelo Dias** and **Karin Becker**
Instituto de Informática (INF)
Universidade Federal do Rio Grande do Sul
Porto Alegre, RS, Brazil
{tiago.zini, marcelo.dias, karin.becker}@inf.ufrgs.br

Abstract

This paper describes a supervised solution for detecting the polarity scores of tweets or headline news in the financial domain, submitted to the SemEval 2017 Fine-Grained Sentiment Analysis on Financial Microblogs and News Task. The premise is that it is possible to understand market reaction over a company stock by measuring the positive/negative sentiment contained in the financial tweets and news headlines, where polarity is measured in a continuous scale ranging from -1.0 (very bearish) to 1.0 (very bullish). Our system receives as input the textual content of tweets or news headlines, together with their ids, stock cashtag or name of target company, and the polarity score gold standard for the training dataset. Our solution retrieves features from these text instances using n-gram, hashtags, sentiment score calculated by a external APIs and others features to train a regression model capable to detect continuous score of these sentiments with precision.

1 Introduction

Sentiment analysis involves the automatic identification of opinions, feelings, evaluations, attitudes expressed by people in the written language. A popular line of work in this field is opinion mining (Liu, 2012; Tsytsarau and Palpanas, 2012). Growing attention has been dedicated to sentiment analysis in the financial domain, given its links to market dynamics. The challenges are to detect how sentiment is expressed in documents in this domain, and how it can translate to a reaction over a company stock, ranging from bullish to bearish. This problem is addressed as part of SemEval-2017 (International Workshop on Semantic Evaluation 2017), Task 5[1]. The task was defined as follows: "given a text instance (microblog message in Track 1, news statement or headline in Track 2), predict the sentiment score for each of the companies/stocks mentioned. Sentiment values need to be floating point values in the range of -1 (very negative/bearish) to 1 (very positive/bullish), with 0 designating neutral sentiment." The task was divided into two subtasks, according to the type of document (i.e. tweets and financial headlines) and sentiment target, and this paper describes our solution for both problems.

We addressed these sub-tasks by building a supervised model to do regression of sentiment value in the documents based solely on their textual content. The target of the sentiment in Task 5-1 is the company stocks for which two sets of annotated tweets were supplied: a training corpus with 1700 annotated tweets and a test corpus with 800 unannotated tweets for task evaluation purpose. Two sets of news headlines were made available as part of Task 5-2, where the target of opinion is a company. The training set was composed of 1142 annotated instances, and the test corpus has 491 unannotated instances for task evaluation. Details of Task 5 can be found at (Cortis et al., 2017).

The regression of sentiment in a text can be complex, because the sentiment can be related in different levels and complexities to the document or just with an aspect or even with a comparison between entities (Feldman, 2013). Our strategy was to address the regression as an opinion mining problem. In addition, sentiment score detection faces challenges common to sentiment analysis in general, such as use of vocabulary and slang specific of the stock market, orthography errors, sarcasm, etc.

Our method extracts a set of features from financial texts and associate this data with annotated

[1] http://alt.qcri.org/semeval2017/task5/

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 837–841,
Vancouver, Canada, August 3 - 4, 2017.

sentiment score provided by each task to train a prediction model specific to sentiment found in tweets and an other for sentiment found in headlines. To explain the details of our solution the remaining of the paper describes the obtained results, the proposed solution and the experiments developed in the next sections respectively.

2 Results

Tasks 5-1 and 5-2 evaluated the proposed solutions according to the cosine similarity of bearish and bullish, considering the respective test dataset. The evaluation was based on cosine similarity as defined by Equation 3, where G_i is the gold standard of instance and P_i is value predicted by our system. The cosine similarity ranges from 0.0 to 1.0. We calculate the cosine similarity considering G like a single vector with all instances of the gold standard, and P with all instances of predictions.

$$cosine(G, P) = \frac{\sum_i^n G_i \cdot P_i}{\sqrt{\sum_i^n G_i} \cdot \sqrt{\sum_i^n P_i}} \quad (1)$$

$$weight\_cosine = \left|\frac{P}{G}\right| \quad (2)$$

$$final\_cs = weight\_cosine \cdot cosine(G, P) \quad (3)$$

In the Task 5-1, our solution was ranked 17th among 25 participants, with a cosine similarity of 0.6142038157. Similarly, in the Task 5-2, we ranked 21srt among 29 participants, with a cosine similarity of 0.6081537843.

3 The Process

This section explains the sequence of steps to preprocess the documents, extract features and train the regression model.

3.1 Text Pre-processing

Before extracting text features, we preprocessed the content of tweets and headlines messages. Full URLs, company cashtags and company names were replaced by the symbols "url", "$cashtag" and "company" respectively. Numbers, monetary values, percentages were replaced by the symbols "positive_number", "negative_number", "money", "positive_percentage", and "negative_percentage". We do the replacing of expression with numeric digits from the more complex to more simple ones, being the more simple case a numeric part of a sequence of characters being replaced by the "positive_number" word. Other substitutions were also performed with dates and other types of numbers. Special character sequences, like emoticons, were replaced by symbols designating their positive or negative value. Emoji's special characters, when identified, were also replaced by the symbol "_emoji_". We also identified expressions that determine negation in a sentence, and replaced these expressions by the symbol "_NOT_", maintaining the adjacent related words unchanged.

Additional pre-processing was implemented over the spans field provided in each tweet input instance. The Span field corresponds to the part of the tweet message related to the target of annotated sentiment. The adjustment done is concatenating its text with the prefix "SPAN_" in order to differ the features derived from spans, from the ones extracted from the complete tweet text.

All these substitutions aim to preserve the original meaning and context of the expressions within the documents, given that these properties would be lost if the textual features were extracted before the pre-processing.

3.2 Features

We extracted the following groups of features from the preprocessed text instances:

Features Common For Both Tasks: The features present in the model of both tasks are:

a) n-grams: we experimented with different variations of n-grams ($n = [1..4]$), which were extracted from both tweet contents/headlines and tweet spans. To deal with sparsity and non-discriminant features, we removed all n-grams whose frequency was below and above given thresholds. Experimentally, we defined as minimum threshold at least 2 times, and as maximum threshold, at most in 95% or 100% of the instances. We chosen a Boolean representation for these features;

b) sentiment polarity and score: we used IBM Alchemy[2] API, providing the tweet text/headline as input. This choice was motivated by our earlier experience on the use of this tool (Dias and Becker, 2016).

Features For Tweets: These are features explored just for tweets:

a) has-hashtag: indicates the presence of hashtag in the document;

b) external stock features: based on the tweet

[2]http://www.alchemyapi.com/

date, we used the Python module Yahoo Finance[3] to get data about stock quotes of cashtag mentioned in the tweet at opening and close time of market. We also calculate the variation from the stock quote price from this date and a future date, using two lags: 7 days and 1 month. We used this data to build three features with the variation in percentage, and three aditional features with information about variation delta symbolized by "increase", "decrease" or "none". Despite the good results provided by the adoption of these features, they could not be not included in the final microblog model because, differently from training dataset, the test dataset had very few instances that included tweet creation date.

3.3 Training

We used the group of features selected for each subtask as detailed in Section 3.2 to train a regression model using a algorithm named Support Vector Regression (SVR), available in the Scikit-learn[4] tools for Python language. The SVR learning was configured only with parameters of linear kernel and $C = 1.0$.

3.4 Training Results

Using annotated sentiment score provided by the SSIX project (Davis et al., 2016), we run our regression models over the test data and compared the results to build a confusion matrix for each subtasks. Tables 1 and 2 describe these matrix in terms of precision and recall, where Bullish is represented by scores greater than 0, and Bearish by negative scores. It is interesting to observe that our solution did not predict any Neutral sentiment, probably because neutral score is exactly 0. It is also possible to observe that recall and precision for Bullish detection is much higher (about 15 percentage points), compared to Bearish. This result might be explained by the prevalence of positive scores in the training instances, as detailed on Table 3.

[3]https://pypi.python.org/pypi/yahoo-finance
[4]http://scikit-learn.org

		Predicted			
		Bullish	**Bearish**	**Neutral**	**Recall**
Actual	**Bullish**	449	72	0	86.02
	Bearish	96	161	0	62.64
	Neutral	5	9	0	0
	Precision	81.04	67.64	0	
	F-score	83.46	65.05	0	

Table 1: Confusion Matrix - Microblog

		Predicted			
		Bullish	**Bearish**	**Neutral**	**Recall**
Actual	**Bullish**	208	68	0	75.36
	Bearish	148	55	0	72.90
	Neutral	6	6	0	0
	Precision	77.32	66.66	0	
	F-score	76.33	69.65	0	

Table 2: Confusion Matrix - News Headline

Subset	**Polarity**	**Quantity**
Microblog	Bullish	1092
Microblog	Bearish	581
Microblog	Neutral	27
Headline	Bullish	653
Headline	Bearish	451
Headline	Neutral	38

Table 3: Polarity Distribution in the Training Datasets

Our solution achieved a higher evaluation score in the first subtask, apparently because the tweets contained more textual information and were freely written using emoticons, Emojis, slangs, financial values and financial language. News headlines were shorter and written in a more formal and standard style. Thus, more discriminative features to train the regression model could be extracted from tweets.

Another difference was the use of cashtags, a compact form to identify one type of company stock, in the tweets. They simplified the detection of company, while the news headlines, in most cases, expressed the companies as composed names. Many news headlines were written entirely using upper case, complicating the distinction of proper names parts from words that have important meaning.

4 Experiments

We made experiments as the basis for our proposed solutions. The experiments for Tasks 5-1 and 5-2 are described in subsections 4.1 and 4.2, respectively. In the both experiments we use different baselines. For each subtask we add some features and test the improvements in cosine similarity measurements.

Based on the models built with improvement results reported in the experiment of each subtask (using 70% of instances for training the model and 30% of them to test the cosine similarity) we evaluate the test instances provided for each subtask.

4.1 Experiments for Subtask of Microblogs

The results of our experiments are reported in Table 4. To evaluate the performance of the proposed system, we adopted as baseline a simple model trained over n-grams (with $n = [1, 2, 3]$). As an improvement, we kept the same n-gram textual features that appeared least twice, and at most in 95% of tweets instances. Then we added the "has-hashtag' feature and the Alchemy score. These results are reported in Table 4 as *Final*, as it corresponds to the solution submitted to Task 5-1.

We further improved this model (labeled *Intermediate* in Table 4) using the previous features, and in addition, all external stock features mentioned in Section 3.2. The only exception was the feature *variation delta in tweet date*. Despite the better result, this model was not submitted to the task, because the features added were not trustworthy in the test data due to the reasons explained in Section 3.2.

4.2 Experiments for Subtask of Headlines

To evaluate the performance of the proposed system, we compare it to a baseline trained over n-grams with $n = [1, 2, 3, 4]$ and keeping only its features that are present at least two instances of headlines. Using the same algorithm we add the feature of sentiment polarity and score of Alchemy API. Results are reported in table 4.

Task	Baseline	Final	Intermediate
Microblog	0.487855	0.518896	0.524003
Headline	0.413345	0.468760	-

Table 4: Improvements gained after the changes in the initial baseline of models in the metric of cosine similarity

5 Conclusions and Future Work

The results obtained by the participants of SemEval Task 5-1 and Task-5-2 and specially our results reveals that polarity regression using cosine similarity as target metric is a hard problem, for which available solutions could evolve.

One of the difficulties we faced was assuming there were no significant differences in the structure of the tweets in the training and testing datasets. As the testing dataset contained very few instances with date information, we could not explore the external features that provided the best results in the training dataset. Another difficulty common to many participant of Task 5 was dealing with the ambiguity in the definition of similarity calculation of the cosine proposed in the description of the tasks. Maybe a standard regression measure like Mean Squared Error would have been a more direct evaluation choice.

The publication of the gold standard for the tasks of Task 5 will allows to us to improve the process, focusing mainly in strategies for increasing the performance with regard to the more complex sentences. Among the strategies are combine Alchemy score with score of others external APIs like Haven On Demand[5] and Vivekn[6] , and the investigation of pre-processed issues like Emojis sentiment. Another approach would be do experiments of deep learning approach.

Acknowledgments

Two of the authors would like to acknowledge Procempa - Brazil by the support provided to the development of this work. This research is partially sponsored in part by CNPq (Brazil) under Grant No. 459322/2014-1.

References

Keith Cortis, André Freitas, Tobias Dauert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 517–533. http://www.aclweb.org/anthology/S17-2089.

Brian Davis, Keith Cortis, Laurentiu Vasiliu, Adamantios Koumpis, Ross McDermott, and Siegfried Handschuh. 2016. Social sentiment indices powered by x-scores. In *2nd International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2016 Lisbon, Portugal*.

Marcelo Dias and Karin Becker. 2016. INF-UFRGS-OPINION-MINING at semeval-2016 task 6: Automatic generation of a training corpus for unsupervised identification of stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, CA, USA, June 16-17, 2016*. pages 378–383.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM* 56(4):82–89.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

[5]https://www.havenondemand.com/

[6]http://sentiment.vivekn.com/docs/api/

Mikalai Tsytsarau and Themis Palpanas. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery* 24(3):478–514.