NILC_USP: A Hybrid System for Sentiment Analysis in Twitter Messages

Pedro P. Balage Filho and Thiago A. S. Pardo
Interinstitutional Center for Computational Linguistics (NILC)
Institute of Mathematical and Computer Science, University of São Paulo
São Carlos - SP, Brazil
{balage, taspardo}@icmc.usp.br

Abstract

This paper describes the NILC_USP system that participated in *SemEval-2013 Task 2: Sentiment Analysis in Twitter*. Our system adopts a hybrid classification process that uses three classification approaches: rule-based, lexicon-based and machine learning approaches. We suggest a pipeline architecture that extracts the best characteristics from each classifier. Our system achieved an F-score of 56.31% in the Twitter message-level subtask.

1 Introduction

Twitter and Twitter messages (tweets) are a modern way to express sentiment and feelings about aspects of the world. In this scenario, understanding the sentiment contained in a message is of vital importance in order to understand users behavior and for market analysis (Java et al., 2007; Kwak et al., 2010). The research area that deals with the computational treatment of opinion, sentiment and subjectivity in texts is called sentiment analysis (Pang et al., 2002).

Sentiment analysis is usually associated with a text classification task. Sentiment classifiers are commonly categorized in two basic approaches: lexicon-based and machine learning (Taboada et al., 2011). A lexicon-based classifier uses a lexicon to provide the polarity, or semantic orientation, of each word or phrase in the text. A machine learning classifier learns features (usually the vocabulary) from annotated corpus or labeled examples.

In this paper, we present a hybrid system for sentiment classification in Twitter messages. Our system combines three different approaches: rule-based, lexicon-based and machine learning. The purpose of our system is to better understand the use of a hybrid system in Twitter text and to verify the performance of this approach in an open evaluation contest.

Our system participated in *SemEval-2013 Task 2: Sentiment Analysis in Twitter* (Wilson et al., 2013). The task objective was to determine the sentiment contained in Twitter messages. The task included two sub-tasks: a expression-level classification (Task A) and a message-level classification (Task B). Our system participated in Task B. In this task, for a given message, our system should classify it as positive, negative, or neutral.

Our system was coded using Python and the CLiPS Pattern library (De Smedt and Daelemans, 2012). This last library provides the part-of-speech tagger and the SVM algorithm used in this work[1].

2 Related work

Despite the significant number of works in sentiment analysis, few works have approached Twitter messages. Agarwal et al. (2011) explored new features for sentiment classification of twitter messages. Davidov et al. (2010) studied the use of hashtags and emoticons in sentiment classification. Diakopoulos and Shamma (2010) analyzed the people's sentiment on Twitter for first U.S. presidential debate in 2008.

The majority of works in sentiment analysis uses either machine learning techniques or lexicon-based

[1]Our system code is freely available at http://github.com/pedrobalage/SemevalTwitterHybridClassifier

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 568–572, Atlanta, Georgia, June 14-15, 2013.

techniques. However, some few works have presented hybrid approaches. König and Brill (2006) propose a hybrid classifier that utilizes human reasoning over automatically discovered text patterns to complement machine learning. Prabowo and Thelwall (2009) evaluates the effectiveness of different classifiers. This study showed that the use of multiple classifiers in a hybrid manner could improve the effectiveness of sentiment analysis.

3 System architecture

Our system is organized in four main components: normalization, rule-based classifier, lexicon-based classifier and machine learning classifier. These components are connected in a pipeline architecture that extracts the best characteristics from each component. The Figure 1 shows the system architecture.

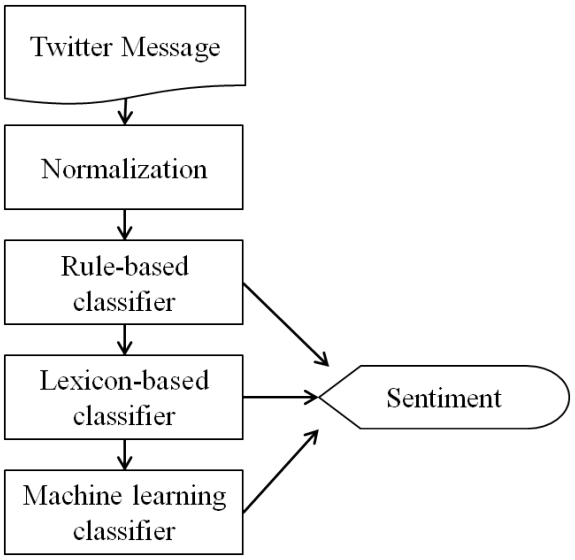

Figure 1: System architecture

In this pipeline architecture, each classifier, in a sequential order, evaluates the Twitter message. In each step, the classifier may determine the polarity class of the message if a certain degree of confidence is achieved. If the classifier may not achieve this confidence threshold, the classifier in the next step is called. The machine learning classifier is the last step in the process. It is responsible to determine the polarity if the previous classifiers failed to achieve the confidence level required to classification. The normalization component is responsible to correct and normalize the text before the classifiers use it.

This architecture improves the classification process because it takes advantage of the multiple approaches. For example, the rule-based classifier is the most reliable classifier. It achieves good results when the text is matched by a high-confidence rule. However, due the freedom of language, rules may not match 100% of the unseen examples, consequently it has a low recall rate.

Lexicon-based classifiers, for example, are very confident in the process to determine if a text is polar or neutral. Using sentiment lexicons, we can determine that sentences containing sentiment words are polar and sentences that do not contain such words are neutral. Moreover, the presence of a high number of positive or negative words in the text may be a strong indicative of the polarity.

Finally, machine learning is known to be highly domain adaptive and to be able to find deep correlations (Taboada et al., 2011). This last classifier might provide the final decision when the previous methods failed. In the following sub-sections, we describe in more details the components in which our system is based on. In the next section, we explain how the confidence level was determined.

3.1 Normalization and rule-based classifier

The normalization module is in charge of correcting and normalizing the texts. This module performs the following operations:

- Elements such as hashtags, urls and mentions are transformed into a consistent set of codes;
- Emoticons are grouped into representative categories (such as happy, sad, laugh) and converted to particular codes;
- Signals of exaltation (such as repetitive exclamation marks) are recognized;
- A simple misspelling correction is performed;
- Part-of-speech tagging is performed.

The rule-based classifier is very simple. The only rules applied here are concerned to the emoticons found in the text. Empirically, we evidenced that positive emoticons are an important indicative of positiveness in texts. Likewise, negative emoticons

indicate negativeness tendency. This module returns the number of positive and negative emoticons matched in the text.

3.2 Lexicon-based classifier

The lexicon-based classifier is based on the idea that the polarity of a text can be summarized by the sum of the individual polarity values of each word or phrase present in the text. In this assumption, a sentiment lexicon identifies polar words and assigns polarity values to them (known as semantic orientations).

In our system, we used the sentiment lexicon provided by SentiStrength (Thelwall et al., 2010). This lexicon provides an emotion vocabulary, an emoticons list, a negation list and a booster word list.

In our algorithm, we sum the semantic orientations of each individual word in the text. If the word is negated, the polarity is inverted. If the word is intensified (boosted), we increase its polarity by a factor determined in the sentiment lexicon. A lexicon-based classifier usually assumes the signal of the final score as the sentiment class: positive, negative or neutral (score zero).

3.3 Machine learning classifier

The machine learning classifier uses labeled examples to learn how to classify new instances. The algorithm learns by using features extracted from these examples. In our classifier, we used the SVM algorithm provided by CLiPS Pattern. The features used by the classifier are bag-of-words, the part-of-speech set, and the existence of negation in the sentence.

4 Hybrid approach and tuning

The organization from *SemEval-2013 Task 2: Sentiment Analysis in Twitter* provided three datasets for the task (Wilson et al., 2013). A training dataset (TrainSet), with 6,686 messages[2], a development dataset (DevSet), with 1,654 messages, and two testing datasets (TestSets), with 3,813 (Twitter TestSet) and 2,094 (SMS TestSet) messages each.

As we said in the previous section, our system is a pipeline of classifiers where each classifier may assign a sentiment class if it achieves a particular confidence threshold. This confidence threshold is a fixed value we set for each system in order to have a decision boundary. This decision was made by inspecting the results table obtained with the development set, as shown below.

Table 1 shows how the rule-based classifier performed in the development dataset. The classifier score consists in the difference between the number of positive emoticons and the number of negative emoticons found in the message. For example, for score of -1 we had 22 negative, 4 neutral and 2 positive messages.

Table 1: Correlation between the rule-based classifier scores and the gold standard classes in the DevSet

Rule-based	Gold Standard Class		
classifier score	Negative	Neutral	Positive
-1	22	4	2
0	311	708	496
1	7	24	71
2		2	4
3 to 6		1	2

Inspecting the Table 1 we adjusted the rule-based classifier boundary to decide when the score is different from zero. For values greater than zero, the classifier will assign the positive class and, for values below zero, the classifier will assign the negative class. For values equal zero, the classifier will call the lexicon-based classifier.

Table 2 is similar to the Table 1, but it now shows the scores obtained by the lexicon-based classifier for the development set. This score is the message semantic orientation computed by the sum of the semantic orientation for each individual word.

Inspecting Table 2, we adjusted the lexicon-based classifier to assign the positive class when the total score is greater than 3 and negative class when the total score is below -3. Moreover, we evidenced that, compared to the other classifiers, the lexicon-based classifier had better performance to determine the neutral class. Therefore, we adjusted the lexicon-based classifier to assign the neutral class when the total score is zero. For any other values, the machine learning classifier is called.

Finally, Table 3 shows the confusion matrix for the machine learning classifier in the development

[2]The number of messages may differ from other participants because the data was collected by crawling

Table 2: Correlation between the lexicon-based classifier score and the gold standard classes in the DevSet

Lexicon-based classifier scores	Gold Standard Class		
	Negative	Neutral	Positive
-11 to -6	26	8	4
-5	15	6	4
-4	31	20	9
-3	32	24	5
-2	57	86	22
-1	25	31	20
0	74	354	115
1	26	70	42
2	28	87	103
3	12	29	81
4	8	9	56
5	2	6	42
6 to 13	4	9	72

dataset. The machine learning classifier does not operate with a confidence threshold, so no decisions were made for this classifier. We see that machine learning classifier does not have a good accuracy in general. Our hybrid approach proposed aims to overcome this problem. Next section shows the results achieved for the Semeval test dataset.

Table 3: Confusion matrix for the machine learning classifier in the DevSet

Machine learning classifier class	Gold Standard Class		
	Negative	Neutral	Positive
negative	35	6	11
neutral	232	595	262
positive	73	138	302

5 Results

Table 4 shows the results obtained by each individual classifier and the hybrid classifier for the test dataset. In the task, the systems were evaluated with the average F-Score obtained for positive and negative classes[3]. We see that the Hybrid approach could improve in relation to each classifier score, confirming our hypothesis.

[3] *Semeval-2013 Task 2: Sentiment Analysis in Twitter* compares the systems by the average F-score for positive and negative classes. For more information see Wilson et al. (2013)

Table 4: Average F-score (positive and negative) obtained by each classifier and the hybrid approach

Classifier	Twitter TestSet	SMS TestSet
Rule-based	0.1437	0.0665
Lexicon-Based	0.4487	0.4282
Machine Learning	0.4999	0.4029
Hybrid Approach	**0.5631**	**0.5012**

Table 5 shows the results in terms of precision, recall and F-score for each class by the hybrid classifier in the Twitter dataset. Inspecting our algorithm for the Twitter dataset, we had 277 examples classified by the rule-based classifier, 2,312 by the lexicon-based classifier and 1,224 the by machine learning classifier. The results for the SMS dataset had similar values.

Table 5: Results for Twitter TestSet

Class	Precision	Recall	F-Score
positive	0.6935	0.6145	0.6516
negative	0.5614	0.4110	0.4745
neutral	0.6152	0.7427	0.6729

6 Conclusion

We described a hybrid classification system used for *Semeval-2013 Task 2: Sentiment Analysis in Twitter*. This paper showed how a hybrid classifier might take advantage of multiple sentiment analysis approaches and how these approaches perform in a Twitter dataset.

A future direction of this work would be improving each individual classifier. In our system, we used simple methods for each employed classifier. Thus, we believe the hybrid classification technique applied might achieve even better results. This strengthens our theory that hybrid techniques might outperform the current state-of-art in sentiment analysis.

Acknowledgments

We would like to thank the organizers for their work constructing the dataset and overseeing the task. We also would like to thank FAPESP and CNPq for financial support.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *The Journal of Machine Learning Research*, 13:2063–2067.

Nicholas A. Diakopoulos and David A. Shamma. 2010. Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1195–1198, New York, NY, USA. ACM.

Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. 2007. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, New York, NY, USA. ACM.

Arnd Christian König and Eric Brill. 2006. Reducing the human overhead in text categorization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 598–603, New York, NY, USA. ACM.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA. ACM.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, pages 79–86, Morristown, NJ, USA, July. Association for Computational Linguistics.

Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307, June.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment in short strength detection informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, December.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov, and Alan Ritter. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13, June.