Combining a Rule-based Classifier with Weakly Supervised Learning for Twitter Sentiment Analysis

Umme Aymun Siddiqua*, Tanveer Ahsan*, and Abu Nowshed Chy†
Department of Computer Science & Engineering
*International Islamic University Chittagong
Chittagong-4314, Bangladesh
†Toyohashi University of Technology
Toyohashi, Aichi, Japan
aymun@kdml.org, tanveer.ahsan@gmail.com, and nowshed@kde.cs.tut.ac.jp

***Abstract*—Microblog, especially Twitter, have become an integral part of our daily life, where millions of user sharing their thoughts daily because of its short length characteristics and simple manner of expression. Monitoring and analyzing sentiments from such massive amount of twitter posts provide enormous opportunities for companies and other organizations to learn about what user think and feel about their products and services. But the ever-growing unstructured and informal user-generated posts in twitter demands sentiment analysis tools that can perform well with minimum supervision. In this paper, we propose an approach for sentiment analysis on twitter, where we combine a rule-based classifier with weakly supervised Naive-Bayes classifier. To classify the tweets sentiment, we introduce a set of rules for the rule-based classifier based on the occurrences of emoticons and sentiment-bearing words, whereas several sentiment lexicons are applied to train the Naive-Bayes classifier. We conducted our experiments based on the Stanford sentiment140 dataset. Experimental results demonstrate the effectiveness of our method over the baseline in terms of recall, precision, F_1 score, and accuracy.**

Keywords: Microblogs, sentiment analysis, sentiment classification, twitter, sentiment lexicons, emoticons.

I. INTRODUCTION

Nowadays, microblog websites play an important role in maintaining social relationships as well as act as a valuable information source. Every day lots of users utilizing microblog sites for expressing their views, opinions, experiences, and feelings about specific topics or entities. Among several microblog sites, Twitter[1] is now the most popular, where millions of users spread millions of tweets (posts on twitter) on a daily basis. As these tons of tweets reflect people's opinions and attitudes, sentiment analysis in twitter has made a hit with a lot of complaisance. Monitoring and analyzing tweets sentiment from twitter provide enormous opportunities for the public and private sectors as well as individual users. For example, people are eager to know the feelings of others about Apple's new product "iPhone 7" and it will provide great convenience for them if the opinions are extracted from massive tweets. As the reputation of a certain product is highly affected by the people's opinions posted in twitter, companies are now monitoring and detecting public opinions from twitter to estimate the extent of product acceptance and to determine the strategies for improving product quality. Fans of famous personalities are always fascinated about what is going on with their favorite person and the reaction from other people. Moreover, there is a strong correlation between the users' posts on twitter and the outcomes of political elections. Both voters and political parties are increasingly seeking ways to get an overview of the support and opposition candidates before the elections [1]. Therefore, a comprehensive sentiment analysis in twitter will provide an effective solution to address these scenarios. In this paper, we have proposed a method for sentiment analysis on twitter. To classify the tweets sentiment either positive or negative, we combine a rule-based classifier with weakly supervised Naive-Bayes classifier. Our rule-based classifier is based on the occurrences of emoticons and sentiment-bearing words, whereas several sentiment lexicons are used to train the Naive-Bayes classifier. Experimental results with Stanford sentiment140 dataset showed that our method improves the sentiment classification performance.

The main contributions of this paper include: (1) We propose a rule-based classifier based on the occurrences of emoticons and sentiment-bearing words and combine it with the weakly-supervised classifier. (2) We investigate the impact of utilizing sentiment lexicons for training the supervised Naive-Bayes classifier instead of the large training dataset. For this, we separately train our own developed Naive-Bayes classifier by using the ensemble of sentiment lexicons and sentiment140 dataset to compare the performances.

The rest of the paper is structured as follows: **Section II** describes the state-of-the-art of sentiment analysis task while our proposed framework is described in detail in **Section III**. Next, in **Section IV**, the experimental setup and results are discussed to show the effectiveness of our proposed method. Finally, some concluded remarks and future directions of our work described in **Section V**.

II. RELATED WORK

Sentiment analysis in a microblogging environment, such as twitter is one of the state of the art research tasks in information retrieval domain, where the major goal is to classify the polarity of a tweet sentiment either positive, negative or neutral based on its contents. Usually sentiment analysis tasks conducted into different levels of graininess, including word level, a phrase or sentence level, document level, and feature level [2]. Prior works on twitter sentiment analysis mostly based on the noisy labels or distant supervision, for example, considering emoticons to decide the tweets sentiment, to train the supervised classifiers, and so on [3][4]. After that, several researchers explore feature engineering along with the combination of machine learning methods like Naive-Bayes (NB), Support Vector Machine (SVM), Maximum Entropy (Max Ent.) etc. to improve the classification performance [5].

[1]https://twitter.com

978-1-5090-6122-8/16/$31.00 ©2016 IEEE

Since it is very costly to obtain sentiment labels for large training data and expressions in twitter are unstructured, informal, and fast-evolving, some researcher introduced the unsupervised methods based on lexicons [6] as well as the combination of lexicons and emotional signals [7]. Along with this direction, current studies showed that results of twitter sentiment analysis is widely used in various social applications, including trend identification of political elections [8], natural disaster management [9], etc.

III. PROPOSED FRAMEWORK

In this section, we describe the details of our proposed framework. The goal of our proposed sentiment classification task is to assign positive or negative sentiment label to a tweet document based on its contents. The overview of our proposed sentiment analysis framework illustrated in Fig. 1.

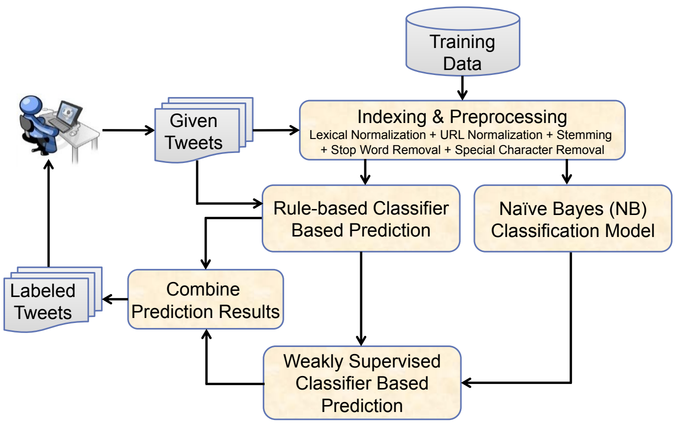

Fig. 1: Proposed sentiment analysis framework

At first, our system fetches the set of tweets from the user and indexed them for further processing. In the preprocessing stage, we perform the tokenization, lexical normalization, URL normalization, stemming, stop-word removal, and special character removal. Next, our proposed rule-based classifier is applied to classify the tweets sentiment as positive, negative or unknown. Tweets that are labeled as unknown are then considered as bag-of-words and classified by using weakly supervised Naive-Bayes (NB) classifier. Results of both rule-based classifier and weakly supervised classifier are then combined and set of labeled tweets return to the user.

A. Lexical Resources for Sentiment Analysis

For our sentiment analysis task, we construct a strong sentiment lexicon. A sentiment lexicon is a collection of sentiment words such as “affluent”, “brilliant”, “crime”, and “hate”, that are used to imply the positive or negative sentiments. We consider seven publicly available sentiment lexicons, including the Bing Liu lexicon [10], subjectivity clues from [11], EffectWordNet [12], WordStat Sentiment Dictionary[2], NRC emotion lexicon [13], SentiStrength lexicon [14], and SentiWordnet [15]. We extract the positive and negative sentiment words separately from these lexicons, which are then used to count the sentiment-bearing words in our rule-based classifier. We also used these lexicons to train our weakly supervised Naive-Bayes classifier.

[2]http://provalisresearch.com/Download/WSD.zip

Like sentiment lexicon, we also construct an emoticon lexicon, as peoples are increasingly interested in using emoticons on twitter in order to express their feelings. For our emoticon lexicon, we combine four publicly available emoticon lists, including the SentiStrength emoticon lexicon [14], emoticon list from Wikipedia[3], Sharpened-text based emoticons[4], and DataGenetics[5] popular emoticons list in twitter.

B. Rule-based Classifier

In a rule-based classifier, a set of rules usually constructed to determine a certain combination of patterns, that are most likely to be related to the different classes. Each rule consists of two parts: the antecedent part and the consequent part. The antecedent part corresponds to a word pattern and the consequent part corresponds to a class label. We can define a rule as follows:

$$R_k : if\ x_1\ is\ A_{k1}\ and\ ....\ x_n\ is\ A_{kn}$$
$$then\ Class = C_k, \quad k = 1, ..., N$$

where R_k is a rule label, A_{k1} is an antecedent set, C_k is a consequent class, k is a rule index, and N is the total number of rules. Our unsupervised rule-based classifier casts the sentiment analysis problem as a multi-class classification problem and labeled each tweet as positive, negative or unknown. We define the following set of rules based on the emoticons and occurring of positive and negative sentiment-bearing words:

$$R_1 : if\ N_{PE} > 0\ and\ N_{NE} = 0\ then,\ Class = Positive$$
$$R_2 : if\ N_{NE} > 0\ and\ N_{PE} = 0\ then,\ Class = Negative$$
$$R_3 : if\ N_{PW} - N_{NW} > P\ then,\ Class = Positive$$
$$R_4 : if\ N_{NW} - N_{PW} > P\ then,\ Class = Negative$$

where N_{PE} is the number of positive emoticons, N_{NE} is the number of negative emoticons, N_{PW} is the number of positive words, N_{NW} is the number of negative words, and P is the positive integer. These rules are applied sequentially. If a tweet satisfies any of these rules, it is labeled with the corresponding class and the remaining rules are ignored. Whereas, tweets that do not satisfy any of these rules are labeled with unknown class. We use the original tweet for R_1 and R_2 rule, while performing the preprocessing described in section III-C1 for R_3 and R_4 rule.

C. Weakly Supervised Learning Approach

Weakly supervised sentiment classification uses the prior word polarity knowledge, where a small number of seed words with known polarity are used to infer the polarity of a target document. The weakly supervised learning process described in detail, as follows:

1) Internal Preprocessing: Internal data preprocessing step is initiated with tokenization, which is the process of forming tokens from an input stream of characters. As we are working on tweet documents, there are lots of emoticons and other special characters. But meaningful English words do not contain these characters. We remove these characters from tweet documents as well as removing the single-letter

[3]https://en.wikipedia.org/wiki/List_of_emoticons
[4]http://www.sharpened.net/emoticons/
[5]http://www.datagenetics.com/blog/october52012/index.html

word. Moreover, in sentiment classification task, stop-words play a negative role because they do not carry any sentiment information and may actually damage the performance of the classifier. For stop-word removal, we utilize the refined form of Indri's standard stop-list[6], because it contains some words that have sentiment information, such as like, thank, useful, downward, etc. So, we manually inspect the list and discard these words from the list. After performing stop-word removal, the remaining words are then stemmed by using Krovetz stemmer[7] to reduce the word variants to a common form.

Lexical Normalization: As tweets are informal user generated texts, it often contains non-standard word forms and some domain-specific entities. Some used non-standard word examples are: "thnx" instead of "thanks", "plzzzzzz" instead of "please", "happpppppy" instead of "happy", etc. To normalize the non-standard words into their canonical forms, we utilize two publicly available lexical normalization dictionaries, collected from [16] and [17]. The first one contains 41,181 words and the second one contains 3,802 words. For URL normalization, we replace all the links present with just the word "URL".

2) Classification Model: We make use of well-known Naive-Bayes (NB) classification model for our sentiment classification system. For a given tweet document, the Naive-Bayes (NB) classifier make use of the joint probabilities of words and categories to estimate the probabilities of categories. The assumption of word independence makes the computation of this classifier far more efficient than the exponential complexity of non-Naive-Bayesian approaches [18].

D. Combining the Classifiers

After developing our proposed rule-based classifier and training the Naive-Bayes classifier with sentiment lexicons described in section III-A, we combine them to classify the tweets sentiment. At first, our rule-based classifier is applied to classify the tweets sentiment as positive, negative or unknown. But, our goal is to classify the tweets sentiment only positive or negative class. That is why; for the tweets that are labeled as unknown by the rule-based classifier, we consider the predictions of weakly supervised Naive-Bayes classifier as the final labels.

IV. EXPERIMENTS AND EVALUATION

A. Dataset Collection

For conducting our experiments, we used the Stanford Twitter Sentiment140 dataset [3][8]. The training dataset has 1.6 million tweets with the equal number of positive and negative tweets, while the test set consists of 182 positive and 177 negative tweets, which were manually annotated. The test set was collected with specific queries arbitrarily chosen from different categories including products' name, companies, people, etc. Table I presents the categories and the total number of corresponding tweets, positive tweets, and negative tweets. Each training and test tweet are composed of the tweet_polarity, tweet_id, date, query, screen_name, and tweet_text.

[6]http://www.lemurproject.org/stopwords/stoplist.dft

[7]http://sourceforge.net/p/lemur/wiki/KrovetzStemmer/

[8]http://help.sentiment140.com/for-students

Table I: Categories for test set

Category	#Tweets	#Positive	#Negative
Company	119	33	86
Misc.	67	26	41
Person	65	48	17
Product	63	47	16
Movies	19	16	3
Location	18	4	14
Events	8	8	0
Total	359	182	177

B. Parameter Setting for Rule-based Classifier

To determine the optimal value of parameter P for the $R3$ and $R4$ rule in our rule-based classifier, we examine the accuracy of our *Run4* method (described in Section IV-C) for different values of P. Based on the result illustrated in Fig. 2, the parameter P is set to as 3.

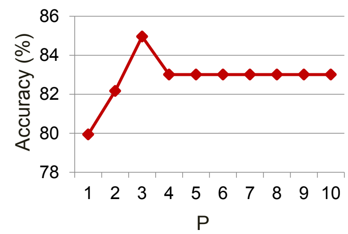

Fig. 2: Accuracy estimation for different values of P

C. Results with Sentiment Classification

At first, we used the training dataset of 1.6 million tweets to train our own developed Naive-Bayes classifier using bag-of-words model for classifying the sentiment of the test set, which we considered as our baseline. To evaluate the performance, we considered four standard evaluation measures including recall, precision, F_1 score, and accuracy.

Table II: Classification results (in %) on sentiment140 dataset

Method	Recall	Precision	F_1 Score	Accuracy
Baseline	63.19	80.99	70.99	73.82
Run1	80.77	81.67	81.22	81.06
Run2	89.56	74.43	81.30	79.11
Run3	85.16	82.45	83.78	83.29
Run4	**91.76**	81.07	**86.08**	**84.96**
SentiStrength [14]	80.77	79.46	80.11	81.70
Semantria[9]	70.88	**83.23**	76.56	78.10

The summarized results of our experiments and comparison with two known related systems (SentiStrength and Semantria) are presented in Table II. At first, we showed the classification performance based on the baseline. Then performance incorporating preprocessing with baseline method resulted in *Run1*. After that, we trained the Naive-Bayes classifier with several sentiment lexicons, which has resulted in *Run2*. Next, to improve the classification performance, we combined our rule-based classifier with the setup of *Run1* and *Run2*. Results of these experiments are articulated in *Run3* and *Run4* respectively. Results showed that both *Run3* and *Run4* outperform the baseline and *Run1* (baseline+preprocessing), which indicates the complementary importance of our proposed rule-based classifier. As we achieved the best result in terms of recall, F_1 score, and accuracy at *Run4*, we can deduce that combining a rule-based classifier with weakly supervised Naive-Bayes classifier improves the sentiment classification performance.

[9]https://www.lexalytics.com/semantria

We also evaluate the performance of our several runs by receiver operating characteristic (ROC) curve analysis. The method, which curve is closer to the y-axis than the other in ROC curve, is treated as the best. Fig. 3 illustrated that the curve of *Run4* method is closer to the y-axis that means it performs best i.e.; yielded the most desirable ratio between false positives and true positives.

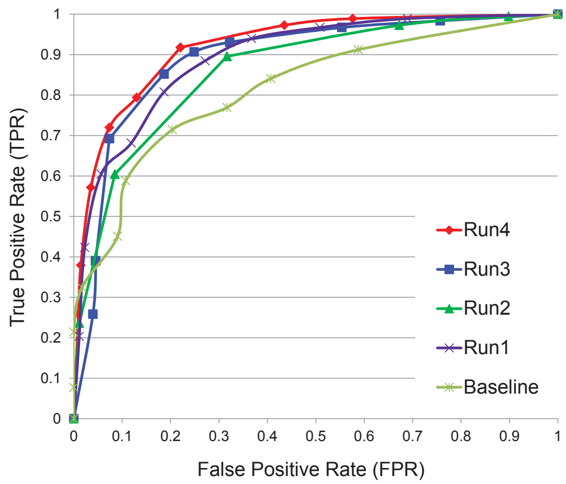

Fig. 3: Receiver operating characteristic (ROC) curve analysis

We also compare the category-wise performance of our proposed method (*Run4*) with baseline. Fig. 4 showed the category-wise comparative results. For 5 types of tweet categories, our proposed *Run4* outperforms the baseline significantly in terms of accuracy and for 2 types of categories, our system yielded similar results. This indicates the applicability of our proposed system for all types of tweets.

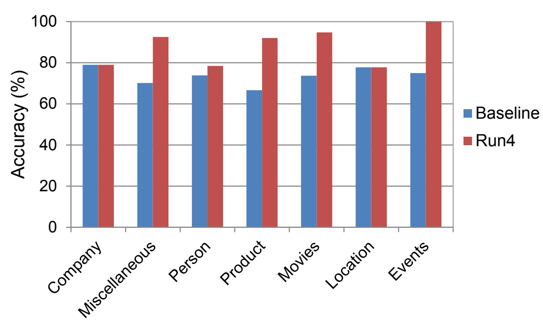

Fig. 4: Category-wise performance comparison

V. Conclusion and Future Direction

In this paper, we proposed an efficient and effective method for sentiment analysis on twitter. We introduced a rule-based classifier based on emoticons and sentiment-bearing words and combined it with our weakly supervised Naive-Bayes classifier to classify the tweets sentiment. To alleviate the construction of large annotated training dataset, we utilized several sentiment lexicons to train the Naive-Bayes classifier and found that it is effective rather than using large training dataset. Experimental results demonstrated that our proposed method achieved the significant improvements in sentiment analysis task.

In future, we have a plan to incorporate several rules based on complex semantics along with the ensemble of standard machine learning techniques to improve the classification efficacy. We also have a plan to consider the neutral sentiment class along with the positive and negative sentiment class.

References

[1] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in *Proceedings of the 20th International Conference on Information & Knowledge Management*. ACM, 2011, pp. 1031–1040.

[2] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM (CACM)*, vol. 56, no. 4, pp. 82–89, 2013.

[3] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[4] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING): Posters*. Association for Computational Linguistics (ACL), 2010, pp. 36–44.

[5] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Combining strengths, emotions and polarities for boosting twitter sentiment analysis," in *Proceedings of the 2nd International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*. ACM, 2013, p. 2.

[6] G. Paltoglou and M. Thelwall, "Twitter, myspace, digg: Unsupervised sentiment analysis in social media," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 66, 2012.

[7] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in *Proceedings of the 22nd International Conference on World Wide Web (WWW)*. International World Wide Web Conferences Steering Committee, 2013, pp. 607–618.

[8] O. Almatrafi, S. Parack, and B. Chavan, "Application of location-based sentiment analysis using twitter for identifying trends towards indian general elections 2014," in *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. ACM, 2015, p. 41.

[9] D. Buscaldi and I. Hernandez-Farias, "Sentiment analysis on microblogs for natural disasters management: a study on the 2014 genoa floodings," in *Proceedings of the 24th International Conference on World Wide Web (WWW) Companion*. International World Wide Web Conferences Steering Committee, 2015, pp. 1185–1188.

[10] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the web," in *Proceedings of the 14th International Conference on World Wide Web (WWW)*. ACM, 2005, pp. 342–351.

[11] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*. Association for Computational Linguistics, 2005, pp. 347–354.

[12] Y. Choi, L. Deng, and J. Wiebe, "Lexical acquisition for opinion inference: A sense-level lexicon of benefactive and malefactive events," in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, 2014, pp. 107–112.

[13] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[14] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *Journal of the American Society for Information Science and Technology (JASIST)*, vol. 63, no. 1, pp. 163–173, 2012.

[15] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining." in *LREC*, vol. 10, 2010, pp. 2200–2204.

[16] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics (ACL), 2012, pp. 421–432.

[17] F. Liu, F. Weng, and X. Jiang, "A broad-coverage normalization system for social media language," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 1035–1044.

[18] A. N. Chy, M. H. Seddiqui, and S. Das, "Bangla news classification using naive bayes classifier," in *Proceedings of the 16th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2014, pp. 366–371.