

# EXPLORE THE EFFECTS OF EMOTICONS ON TWITTER SENTIMENT ANALYSIS

Katarzyna Wegrzyn-Wolska<sup>1</sup>, Lamine Bougueroua<sup>1</sup>, Haichao Yu<sup>2</sup>,  
Jing Zhong<sup>2</sup>

<sup>1</sup>Esigetel, Groupe Efrei Paris-Sud, Villejuif, France

katarzyna.wegrzyn@groupe-efrei.fr,

lamine.bougueroua@groupe-efrei.fr

<sup>2</sup>Allianstic, Groupe Efrei Paris-Sud, Villejuif, France

haichao.yu.20150767@efrei.net, jing.zhong.20150772@efrei.net

## ABSTRACT

*In recent years, Twitter Sentiment Analysis (TSA) has become a hot research topic. The target of this task is to analyse the sentiment polarity of the tweets. There are a lot of machine learning methods specifically developed to solve TSA problems, such as fully supervised method, distantly supervised method and combined method of these two. Considering the specialty of tweets that a limitation of 140 characters, emoticons have important effects on TSA. In this paper, we compare three emoticon pre-processing methods: emotion deletion (emoDel), emoticons 2-valued translation (emo2label) and emoticon explanation (emo2explanation). Then, we propose a method based on emoticon-weight lexicon, and conduct experiments based on Naive Bayes classifier, to validate the crucial role emoticons play on guiding emotion tendency in a tweet. Experiments on real data sets demonstrate that emoticons are vital to TSA.*

## KEYWORDS

*Social Media, Social Network Analysis, Text Mining, Sentiment analysis, Tweets, Emoticon*

## 1. INTRODUCTION

Sentiment Analysis (SA) [1] is a computational study of how opinions, attitudes, emoticons and perspectives are expressed in language. With the development of social network and dramatic development of big data, SA has been applied to a variety of domains to solve practical problems, such as understanding customer feedback, brand analysis, understanding public opinions, financial prediction, etc. Therefore, SA has become an important and hot research topic, which has attracted a large number of researchers from domains of machine learning, data mining and natural language processing (NLP). Theoretically, there are 3 classes of sentiment: positive, negative and neutral. However, most of the researchers usually focus on polarity classification: classifying sentence or document as positive or negative, which is two-way classification problem. Since SA has been formulated as machine learning based text classification problem by [2] [3] [4], machine learning methods have become the most important methods to solve SA problem.

Twitter is one of the most popular online social networking service today, which allow users to send and read short messages called tweets. With tweets, people can share with other people what they are doing and thinking [5]. According to recent statistical data<sup>1</sup>, as of March 2016, there have been more than 310 million monthly active users and 330 million tweets are generated every day. The most important feature of Twitter is that every tweet is a message up to 140 characters. It is because of this character limitation that emoticon become very important in tweets, since emoticon can help people better express their emotion in a short message. However, most of the researchers have dismissed emoticons as noisy information and delete them in the pre-processing process. Nevertheless, we will explore the influence of emoticons on SA in this paper.

Very often SA is applied on movies review and news article [3] [4] [6]. Compared with movie reviews and news articles, tweets have a lot of difference [7]. On the one hand, tweets are shorter and more ambiguous than movie reviews and news articles because of the limitation of words. On the other hand, tweets contain much more misspelled words, slang, modal particles and acronyms because of the casual form. Considering these difference, the traditional SA methods for movie reviews and news articles are not appropriate for Twitter Sentiment Analysis (TSA) problem. Actually, many novel SA methods have been specifically developed for TSA, which include fully supervised method and distantly supervised method. With manually labelled data, fully supervised methods like Multinomial Naive Bayes (MNB) and support vector machine (SVM) are more accurate, but labelling data manually is more labour-intensive and time consuming. With data collected by Twitter API, distantly supervised methods are more efficient but less accurate. [8] even combined these two methods and developed the emoticon smoothed language models (ESLAM) for TSA.

In this study, we explore the effects of emoticons on TSA. At first, we compare three emoticon pre-processing methods: emotion deletion (*emoDel*), emoticons 2valued translation (*emo2label*) and emoticon explanation (*emo2explanation*). After that, we propose a method based on emoticon-weight lexicon to explore the influence of emotion on TSA. Experiments on real data sets demonstrate that emoticons are vital to TSA.

## 2. RELATED WORK

SA [1] has been a popular research topic over the past decades. Before [2], knowledge-based method dominated this domain. However, in [2], authors show that machine learning techniques like naive Bayes, maximum entropy and support vector machine can outperform the knowledge-based baselines on movie reviews. After that, machine learning based methods have become the most important methods for SA.

With the rapidly growth of Twitter, more and more researchers started to focus on TSA. Most of earlier works on TSA are fully supervised methods. In [9] [10], authors use traditional SA methods on normal text form to solve TSA problems. Authors propose target-independent SA based on SVM in [11]. In [12], authors present a dynamic artificial neural network to handle TSA.

Recently, different supervised methods are proposed. Authors in [13] utilize Twitter API to get training data which contain emoticons like :) and :( . They use these emoticons as noisy labels.

---

<sup>1</sup> <https://about.twitter.com/company>

Tweets with :) are thought to be positive training data and tweets with :( are thought to be negative training data.

In [8], authors present the ESLAM which combine fully supervised methods and distantly supervised methods. Although a lot of TSA methods have been presented, few of them explored the influence of emoticons on TSA, which motivates our work in this paper.

### 3. EXPLORE EFFECTS OF EMOTICONS

In this section, first we present our basic TSA classifier based on Naive Bayes (NB). Then, we introduce an emoticon lexicon which contain 50 most commonly used emoticons. After that, we present 3 emoticon pre-processing methods: *emoDeletion*, *emo2label* and *emo2explanation*.

Finally, we propose a method based on emoticon-weight lexicon and introduce a strategy to integrate emoticon-weight lexicon method with naive Bayes method.

#### 3.1. Naive Bayes (NB) Model for SA

In this paper, we use a Twitter-aware tokenizer<sup>2</sup> combined with a Naïve Bayes model as our basic classifier. Refer to the Stanford Classifier<sup>3</sup>, here is the basic idea for the Naive Bayes:

We assume that:

- $n$  is the number of words appeared in training set  $T$ ,
- $n_{c_j}$  is the number of feature which belong to class  $j$  ( $c_j$ ) in training set  $T$  ( $j$  can be positive or negative),
- $n_{f_i}$  is the number of times feature  $i$  appeared in training set  $T$ ,
- $n_{f_i c_j}$  is the number of times feature  $i$  appeared in class  $j$ .

Then, we use the following equations to compute the probabilities  $p_{c_j}$  and  $p_{f_i c_j}$ :

$$p_{c_j} = \frac{n_{c_j} + \varepsilon}{n + |\text{classes}| \times \varepsilon} \quad (1)$$

$$p_{f_i c_j} = \frac{n_{f_i c_j} + \sigma}{n_{f_i} + |\text{classes}| \times \sigma} \quad (2)$$

While we have two classes (positive and negative), so  $|\text{classes}| = 2$ .

In (1) (2), the parameters  $\varepsilon$  and  $\sigma$  are smoothing parameters to avoid assigning zero weight to unseen feature. In our experiment, we choose  $\varepsilon = 10^{-30}$  and  $\sigma = 1.0$  (Laplacian smoothing).

With (1) (2), we can compute negative weight and positive weight of every feature:

$$W_{i,j} = \log \left( \frac{p_{f_i c_j}}{p_{c_j}} \right) \quad (3)$$

<sup>2</sup> <http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py>

<sup>3</sup> <http://nlp.stanford.edu/software/classifier.shtml>

After get weights of all features, we can compute the weights of sentences according to Naive Bayes assumption.

Assuming that tweet  $t$  consists of  $n$  features, then the weights of the tweet  $t$  will be:

$$W_{sentence_{t,j}} = \sum_{i=1}^n W_{i,j} \quad (4)$$

Finally, we will compute the possibilities of the sentence belonging to negative class and positive class:

$$P(t | neg) = \frac{e^{W_{t,neg}}}{e^{W_{t,neg}} + e^{W_{t,pos}}} \quad (5)$$

$$P(t | pos) = \frac{e^{W_{t,pos}}}{e^{W_{t,neg}} + e^{W_{t,pos}}} \quad (6)$$

### 3.2. Emoticon Lexicon

Our emoticon lexicon is based on a Twitter emoticon analysis<sup>4</sup> which collected a large number of most commonly used emoticons. We choose the top 50 emoticons as our emoticon lexicon.

For every emoticon, we give a polarity value which can be negative or positive, a specific translation and a weight. This lexicon is showed in Table 1. We will use this emoticon lexicon in subsequent parts.

Table 1. Emoticon Lexicon

Emoticon	Value	Translation	Weight
:) :D :-) ;) XD :] =) (: ;-) =D =] :-D ^_^ (8 :o) (;=o 8) ;o) (= [: 8D :]	POSITIVE	happy	1
:o ;O o:	POSITIVE	surprise	1
=P :-P ;P =P	POSITIVE	playful	1
;D ;]	POSITIVE	wink	1
\m/	POSITIVE	salute	1
:( D: =( ) : ; : [ ;( =[	NEGATIVE	sad	-1
=/ :-/ \ ;/ :-/ =\	NEGATIVE	annoyed	-1
:'(	NEGATIVE	crying	-1
:@	NEGATIVE	angry	-1
:	NEGATIVE	indifferent	-1

### 3.3. Emoticon Pre-processing Methods

**EmoDeletion:** In this emoticon pre-processing method, we just delete all the emoticons defined in emoticon lexicon in TABLE 1 from the training data.

<sup>4</sup> <http://www.datagenetics.com/blog/october52012/index.html>

**Emo2label:** This emoticon pre-processing method is pretty simple and straightforward. We give all the emoticons a 2-valued label: NEGATIVE or POSITIVE. We give a label of NEGATIVE to those emoticons with negative meanings and give a label of POSITIVE to those emoticons with positive meanings. This kind of translation is not so close to natural language, but it is more intuitive and robust because it could avoid some translation errors. For both training data and test data, when we find any emoticon defined in emoticon lexicon, we replace it with its 2-valued labels in pre-processing.

**Emo2explanation:** When two people communicate face to face, they could notice the expression like “smile” or “frown” made by the other. For example, A is frowning and says to B “I’m fine”. If C asks B the recent situation of A, B will not ignore A’s expression but translate A’s expression naturally. B will say: “I saw some days ago. She said she was fine but I noticed she was frowning. So I think maybe she met some trouble.” Such like that, almost every emoticon can be described as a verbal word and it is much easier for a computer to recognize a word rather than an emoticon since most of the features extracted by classifier are words. Because of the similarity of some emoticons, we organize emoticons into emoticon synonymy sets, which we define as groups of emoticons with the same translation (see TABLE 1). From both training data and testing data, when we find any emoticon defined in emoticon lexicon, we replace it with its translation in pre-processing. For example, a tweet “This movie so cool!! :)” are translated into “This movie so cool!! happy” after pre-processing.

### 3.4. Emoticon-Weight Lexicon Model (EWLM) for SA

In polarity classification, we place a text into negative or positive class. Similarly, we use a polar weight to define an emoticon which is a character sequences. For an emoticon with positive meaning, we give it the value 1, otherwise, we give it the value -1 [14]. The format of an emoticon-weight lexicon is (emoticon, weight), for example, (:), 1), (:, -1).

When classifying a text, we consider both emoticons and verbal cues, and combine the two factors to get an integrated assessment to the text. The framework is as below [Figure 1]: Firstly, we load a set of tweets for analysing sentiment. Then, the classifier split it into different tweets. For each tweet, we check if this tweet contains emoticon.

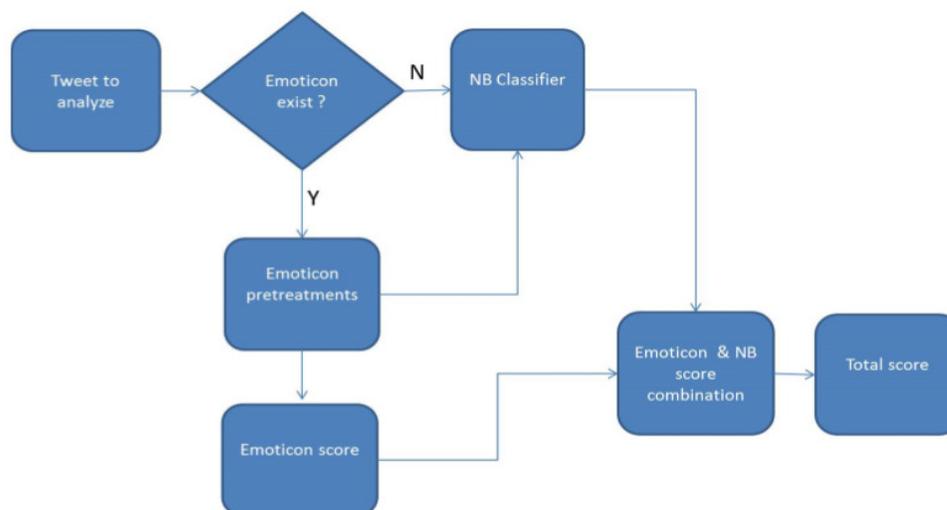


Figure 1. Framework architecture

We compare each word in the tweet with the emoticon lexicon entries. If there exist emoticons which match the emoticons in lexicon, we compute the emoticon score of this tweet and combine this score with words score. Otherwise, we just use the words score which is given by the NB classifier. When the tweet  $i$  contains emoticon,  $e_i = 1$ , otherwise  $e_i = 0$ . i.e.

$$e_i = \begin{cases} 0, & \text{no emoticon in tweet } i \\ 1, & \text{exist emoticon in tweet } i \end{cases} \quad (7)$$

For every tweet, the NB classifier gives us two probabilities  $p_{iw}(neg)$  and  $p_{iw}(pos)$  for classifying verbal cues. If  $p_{iw}(neg) > p_{iw}(pos)$ , the NB classifier places the tweet into negative class. Otherwise, the tweet is placed into positive class. When  $e_i = 1$ , the emoticon score of  $i^{th}$  tweet  $s_{ie}$  equals the sum of weights of each emoticon. Assuming that the number of emoticons in  $i^{th}$  tweet is  $N_i (N_i > 0)$ , and the weight of  $j^{th}$  emoticon is  $W_{emoj}$ , we have:

$$S_{ie} = \sum_{j=1}^{N_i} W_{emoj} \quad (8)$$

The emoticon-weight lexicon helps us to deal with only emoticons. The NB classifier deals with verbal cues. Hence, we need a combination strategy to combine EWLM with NB classifier, to get a final classification result.

As above,  $s_{ie}$  is the sum of weight of emoticons in tweet  $i$ , which is not in the range of  $(0, 1)$ . We use the Sigmoid function to convert the range of  $s_{ie}$  into a new range which is between 0 and 1, because we need to combine this value with a probability value which is between 0 and 1 given by the NB classifier. With Sigmoid function, we can compute P\_EWLM:

$$P_{EWLM}(t|pos) = Sigmoid(S_{ie}) \quad (9)$$

$$P_{EWLM}(t|neg) = 1 - Sigmoid(S_{ie}) \quad (10)$$

The sentiment of both emoticons and verbal cues can be computed as a probability of being negative or positive. We use  $\alpha$  as a factor, which decides the importance of the emoticon in a tweet, to integrate these two probabilities and get the final probabilities.  $p_i(pos)$  is the probability of the  $i^{th}$  tweet being positive, and  $p_i(neg)$  is the probability of the  $i^{th}$  tweet being negative. If  $\alpha \geq 0.5$ , verbal cues play a more important role. Otherwise, the emoticon occupies a greater proportion on analysing sentiment.

$$P_i(neg) = \alpha \times P_{NB(neg)} + (1-\alpha) \times P_{EWLM}(neg) \quad (11)$$

$$P_i(pos) = \alpha \times P_{NB(pos)} + (1-\alpha) \times P_{EWLM}(pos) \quad (12)$$

The classification  $c_i$  of  $i^{th}$  tweet is defined as a function of its final probabilities  $p_i(neg)$  and  $p_i(pos)$ :

$$c_i = \begin{cases} \text{positive,} & \text{if } P_i(pos) \geq P_i(neg) \\ \text{negative,} & \text{if } P_i(pos) < P_i(neg) \end{cases} \quad (13)$$

## 4. EXPERIMENT DESIGN

### 4.1. Data Set

We use the publicly available Sanders Corpus<sup>5</sup> as our experiment data, which consist of 5513 manually labelled tweets. These tweets involved with four different topics: Apple, Google, Microsoft, and Twitter. After removing the no English tweets, spam tweets, re-tweets and duplicate tweets, and setting the classes to be balanced, we get 952 tweets for polarity classification, including 476 negative tweets and 476 positive tweets. There are 200 tweets which contain emoticons in the whole data set (which means approximately 21% tweets contain emoticons).

We take the following measures to pre-process the data:

1. Replace the Twitter usernames which start with @ with USERNAME.
2. Replace urls in tweets with URL.
3. All words are changed to their lower cases. With these pre-processing measures, we can reduce the influence of meaningless strings and extract more representative features.

### 4.2. Experiment Setting

We assume that the total number of data, including training data and test data, is  $X$  ( $= 952$ ). For every experiment, we randomly sample the same amount of tweets (say  $Y$ ,  $Y = 16, 32, 64...$ ) for both negative class and positive class as our training set, and use the rest  $X - 2Y$  tweets as our test set. In order to avoid the experiment contingency, every time we will conduct 60 times experiments independently and get the average performance, which is more accurate.

### 4.3. Evaluation

We evaluate the performance of our experiments by the values of accuracy and Macro-level F1-score. Accuracy is the percentage of correctly predicted data in all test data. The Macro-level F1-score is the average of the F1-scores of the positive and negative classifiers, where F1-score is the harmonic mean of precision and recall. F1-score is related with precision and recall calculated by the simplified formula [14]:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (14)$$

## 5. EXPERIMENT RESULTS

### 5.1. Effects of emoticon pre-processing methods

We conduct experiments based on NB model to compare with and without emoticon pre-processing methods and explore the influence of emoticons. In this experiment, we use different number of training data (i.e.  $2Y = 32, 64, 128, 256, 512, 768$ ). The results are illustrated by Figure 2 with accuracy and Figure 3 with Macro-level F1-score.

---

<sup>5</sup> <http://www.sananalytics.com/lab/twitter-sentiment/>

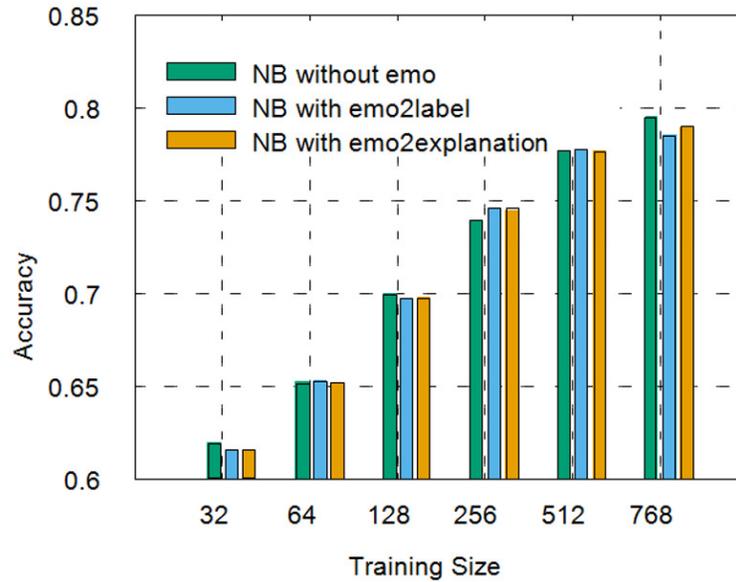


Figure 2. Effects of emoticon pre-processing methods measured by Accuracy

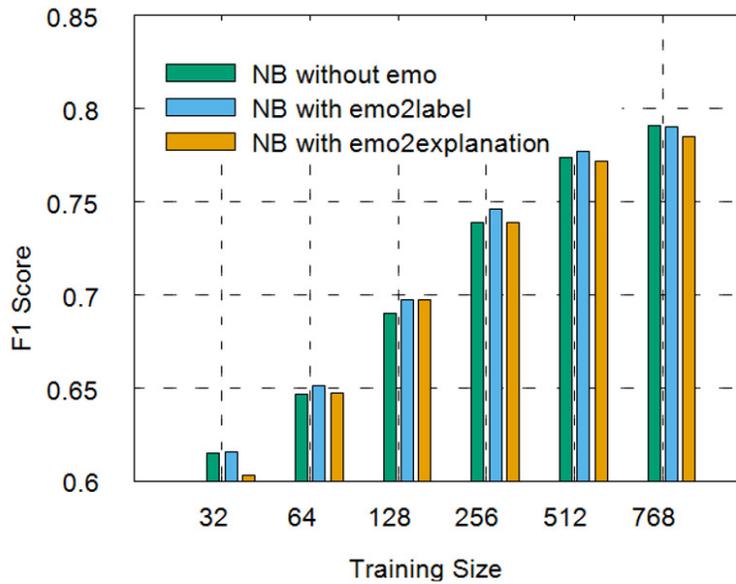


Figure 3. Effects of emoticon pre-processing methods measured by Macro level F1 score

From Figure 2 and Figure 3, we can easily see that emo2label has the best performance among the proposed emoticon pre-processing methods.

## 5.2. Effects of Emoticon-Weight Lexicon Model

We compare the performance of the NB model with and without EWLM to judge if EWLM can help the NB model to raise the performance on TSA. In this experiment, we also use different

training size to train the classifier and utilize accuracy and Micro-level *F1* score to evaluate the classifier.

The experiment result is showed in Figure 4 and Figure 5.

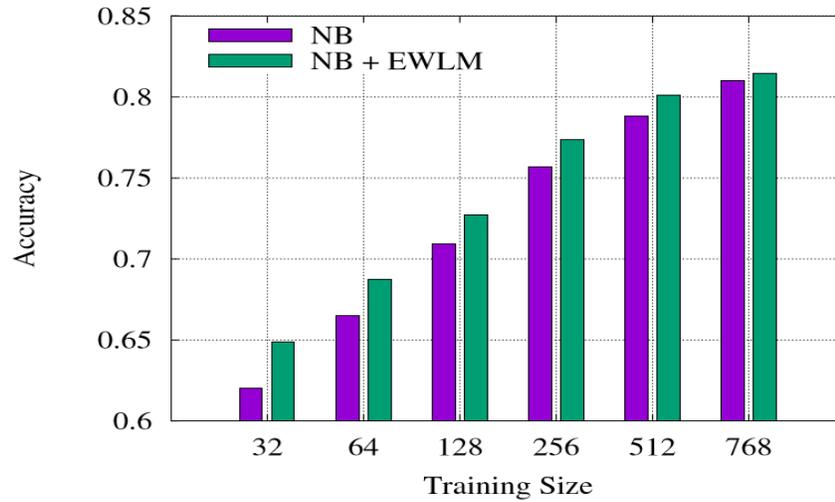


Figure 4. Effects of EWLM measured by accuracy

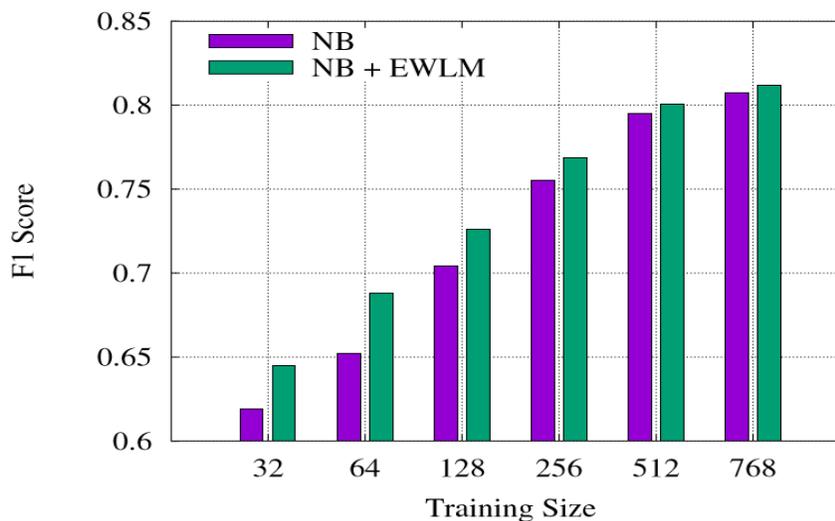


Figure 5. Effects of EWLM measured by Macro-level F1 score

From Figure 4 and Figure 5, it is obvious that EWLM can help the NB model to raise the performance on TSA, especially when the training size is small. When the training size is big enough, the data can provide more discriminating information for training the NB classifier, and the NB classifier could achieve a better performance. In this condition, the improvement brought by EWLM will become smaller. Anyway, the experiment results imply that the emoticons do have important information which could help the NB classifier to achieve better performance on TSA tasks.

### 5.3. Effects of the Combination Parameter Alpha

Alpha is a significant factor to combine NB model with EWLM. When alpha equals 1, there will be only NB model to conduct TSA task. When alpha is smaller, the EWLM will play a more important role in the combined classifier. In this experiment, we try different value of alpha to check which value of alpha is best. The experiment results can be seen in Figure 6 (training size equals 128) and Figure 7 (training size equals 512).

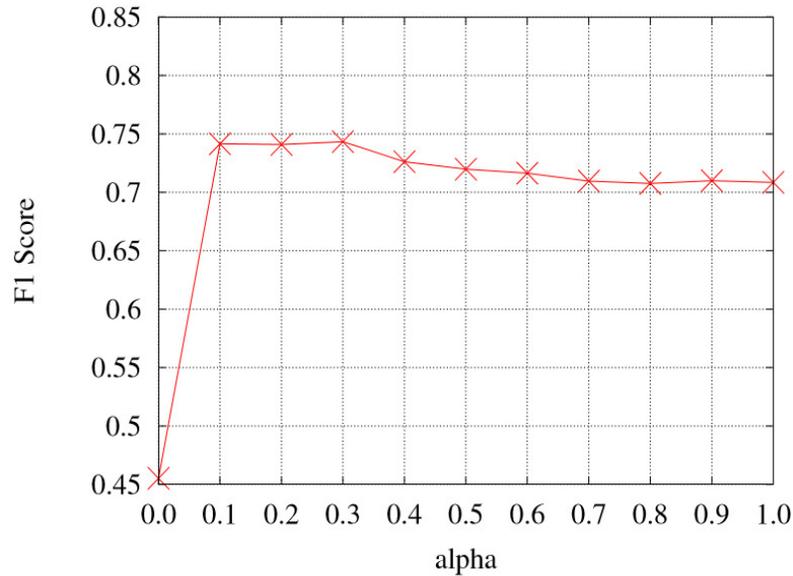


Figure 6. Effect of combination factor alpha with 128 training data

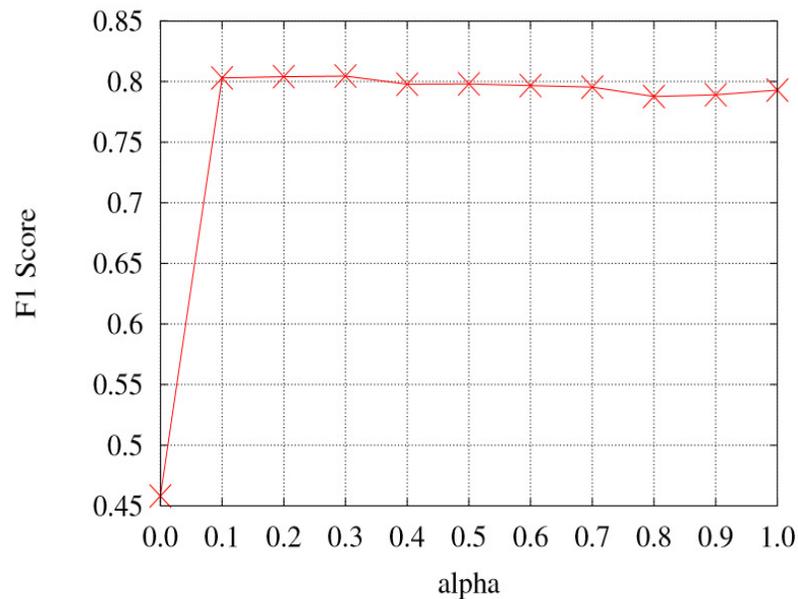


Figure 7. Effect of combination factor alpha with 512 training data

The experiments result in Figure 6 and Figure 7 clearly show that the combination strategy is better than the single NB model or single EWLM. Furthermore, we can see that when alpha take values from 0.1 to 0.3, the classifier can achieve the best performance. Also, in the experiment with 512 training example, we can notice that when alpha becomes larger, the performance of the classifier will not be influenced a lot. This is because a large manually labelled training data can provide enough discriminating information for the TSA classifier.

Our results could clearly indicate that considering emoticon into classifier is a necessary addition on sentiment analysis and whether a tweet contains emoticon or not, our methods will not weaken the performance. If no emoticon in data, the performance of our methods is same with NB classifier.

## 6. CONCLUSIONS

With the significance of sentiment analysis being recognized and the popularity rate of emoticon in social network getting higher and higher, the role of emoticon cannot be ignored on polarity classification. Our key contribution in this paper lies in validating the important role emoticon plays in conveying overall sentiment of a text in TSA though a series of experiments.

We compare 3 emoticon pre-processing methods and emoticon-weight lexicon method on the base of Twitter aware tokenizer and NB Model. We propose a combination strategy using factor alpha to integrate the Emoticon-Weight Lexicon with classifier. The result shows that the usage of emoticon-weight lexicon model improves the performance of NB model on TSA task. We can get the conclusion that some emoticons dominate the sentiment of a tweet and conquer the emotion of verbal cues.

As our results are very promising, we assume several directions for further work. First, we will look for some authoritative help to improve our emoticon dictionary and set more detailed score for emoticon weight to show its intensity of emotion. Second, we will study the impact of number of emoticons in experimental data on our emoticon weight lexicon.

## REFERENCES

- [1] Pang, Bo and Lee, Lillian, Opinion mining and sentiment analysis, *Journal Foundations and trends in information retrieval*, volume 2, number 1-2, pages 1–135, 2008.
- [2] Pang, Bo and Lee, Lillian and Vaithyanathan, Shivakumar, thumbs up? sentiment classification using machine learning techniques, *Proceedings of the ACL-02 conference on Empirical methods in natural language Processing-Volume 10*, pages 79–86, 2002.
- [3] Dzikowski, G., & Wegrzyn-Wolska, K. 2007b. Rcss - rating critics support system purpose built for movies recommendation. In: *Advances in Intelligent Web Mastering*. Springer.
- [4] Dzikowski, G., & Wegrzyn-Wolska, K. 2008a. An autonomous system designed for automatic detection and rating of film. Extraction and linguistic analysis of sentiments. IN *Proceedings of WIC*, Sydney.

- [5] Janik Lthi, Lamine Bougueroua and K. Wegrzyn-Wolska, Sentiment Polarity on Twitter messages with geolocation, in proceedings of the International Workshop on Computational Social Networks (IWCSN 2014) within the 15th International Conference on Web Information System Engineering WISE 2014, Thessalonique, Greece, October 2014, Springer Lecture Notes in Computer Science (LNCS).
- [6] Dzikowski, G., & Wegrzyn-Wolska, K. 2008b. Tool of the intelligence economic: Recognition function of reviews critics. In: ICSOFT 2008 Proceedings. INSTICC Press.
- [7] Wegrzyn-Wolska, K., Bougueroua, L.: Tweets mining for French Presidential Election, In proceeding of the 4th IEEE/WIC International conference on computation aspects of social networks - CASoN 2012, SaO Carlos, Brazil, November (2012).
- [8] Liu, Kun-Lin and Li, Wu-Jun and Guo, Minyi, Emoticon Smoothed Language Models for Twitter Sentiment Analysis., AAI, 2012.
- [9] Jansen, Bernard J and Zhang, Mimi and Sobel, Kate and Chowdury, Twitter power: Tweets as electronic word of mouth, Journal of the American society for information science and technology, volume 60, number11, pages 2169–2188, 2009, Wiley Online Library.
- [10] Bermingham, Adam and Smeaton, Alan F, Classifying sentiment in microblogs: is brevity an advantage? Proceedings of the 19th ACM international conference on Information and knowledge management, pages 1833–1836, 2010, ACM.
- [11] Jiang, Long and Yu, Mo and Zhou, Ming and Liu, Xiaohua and Zhao, Tiejun, Target-dependent twitter sentiment classification, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 151–160, 2011.
- [12] Ghiassi, M and Skinner, J and Zimbra, D, Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, Journal Expert Systems with applications, volume 40/16, pages 6266–6282, 2013, Elsevier.
- [13] Go, Alec and Bhayani, Richa and Huang, Lei, Twitter sentiment classification using distant supervision, journal CS224N Project Report, Stanford, volume 1, pages 12, 2009.
- [14] Hogenboom, Alexander and Bal, Daniella and Frasincar, Flavius and Bal, Malissa and De Jong, Franciska and Kaymak, Uzay, Exploiting Emoticons in Polarity Classification of Text., J. Web Eng., volume 14, num1&2, pages22–40, 2015.

## AUTHORS

**Katarzyna Węgrzyn-Wolska** received M.Sc. from the Silesian Technical University of Gliwice (Poland), and a further M.Sc. in Computer Science from the University of Val Essonne (France), her Ph.D. (2001) in Automatics, Real Time Computing and Computer Science from the Ecole Supérieur des Mines de Paris, France and the habilitation (H.D.R), to become Full Professor in 2012. She is the Principal Professor and head of an SITR team at ESIGETEL, France. She is editor-in-chief of the International Journal on Social Informatics edited in ICST Transactions Series. She is involved in the organization of several International Conference as well as an expert for the group Information Society Technologies (IST) active in the European Community. Her main interests are Information Retrieval, Search Engines, Web Based Support Systems, Web Intelligence and Social Networks.



**Lamine Bouguroua:** He is a Research Associate Professor at school of Computer Science and Engineering ESIGETEL (Ecole Supérieure d'Informatique et Génie des Télécommunications), Villejuif, France. He received the Ph.D. degree in Sciences from the University of Paris XII, Paris, France, in March 2007. His research interests include programming languages, software architecture, object-oriented software systems, program analysis, scheduling, embedded systems, real time systems and fault tolerance. Today, his activity is concerned with scheduling and fault tolerance in Real-Time Systems, social network, multi-agent system.

