



A New Emotional Vector Representation for Sentiment Analysis

Hanen Ameer^(✉), Salma Jamoussi, and Abdelmajid Ben Hamadou

Multimedia InfoRmation Systems and Advanced Computing Laboratory,
MIRACL-Sfax University, Sfax-Tunisia Technopole of Sfax:
Av.Tunis Km 10 B.P. 242, 3021 Sfax, Tunisia
ameurhanen@gmail.com,
{salma.jamoussi, abdelmajid.benhamadou}@isimsf.rnu.tn

Abstract. With the advent of Web 2.0, social networks (like, Twitter and Facebook) offer to users a different writing style that's close to the SMS language. This language is characterized by the presence of emotion symbols (emoticons, acronyms and exclamation words). They often manifest the sentiments expressed in the comments and bring an important contextual value to determine the general sentiment of the text. Moreover, these emotion symbols are considered as multilingual and universal symbols. This fact has inspired us to research in the area of automatic sentiment classification. In this paper, we present a new vector representation of text which can faithfully translate the sentimental orientation of text, based on the emotion symbols. We use Support Vector Machines to show that our emotional vector representation significantly improves accuracy for sentiment analysis problem compared with the well known bag-of-words vector representations, using dataset derived from Facebook.

Keywords: Emotional TFIDF vector · SVM classifier
Sentiment analysis · Facebook dataset · Emoticons

1 Introduction

With the development of the internet and the advent of community sites like social networks (e.g., Facebook and Twitter) and blogs, more and more people share their opinions and express their sentiments towards a considerable variety of topics. Every day, there is an ever-increasing number of textual data expressing users' sentiments. The explosion of online opinionated texts, gives rise to new research areas; opinion mining and sentiment analysis. These areas have been actively developing recently. An important and motivating research direction of sentiment analysis boils down to an automatic classification problem. Sentiment classification aims to group the subjective texts according to the opinion or judgment which they express by assigning a polarity label to texts (binary: positive or negative or multivariate).

There are two main kinds of sentiment classification approach. The first one is the linguistic approach (lexicon-based), which consists of calculating the semantic orientation (valence) of texts by obtaining word polarities from a lexicon. Thus, this approach requires to construct a properly labelled sentiment lexicon in order to classify texts (using the manual method [19], the dictionaries-based method [8], the corpora-based method [2], the method combining the last two ones [12] or the concept-based method [3]). The second one is the statistical approach, (learning-based), which involves using the machine learning methods (like Support Vector Machines, Naïve Bayesian classifiers, Maximum Entropy, Neural Networks, etc.) to produce automated classifiers and generate the class labels for opinionated texts.

In this paper, we focus on the statistical approach to classify texts into positive or negative. This approach adopts machine learning classification techniques with feature selection. A critical task of learning-based sentiment analysis system is how to find a representation (set of features) which can faithfully capture and translate the sentimental characteristics of a text. [13] were the first who applied a machine learning in sentiment analysis. They evaluated the contribution of different features including uni-grams, bi-grams and part-of-speech tags. Alternatively, words can be counted as booleans as shown in [13], or weighted by their TF-IDF score like [9], TF- Δ IDF score [10]. We introduce a novel way to represent words by distinguishing eight different emotional states which formulate the vector space. We propose vector representations of words and sentences based on the notion of emoticons, in order to represent numerically their semantic and sentimental characteristics. In fact, the emoticons are considered as multilingual and universal symbol. We perform the evaluation using the Facebook dataset to see the effect of emoticons as sentence feature on polarity determination. Throughout the evaluation, we show how this improves accuracy.

The content of this paper is organized as follows: First, in Sect. 2, we review the related work for sentiment analysis. In Sect. 3, we elaborate on our proposed method for sentiment classification. We present the proposed emotional vector representations. Next, we report the experimental results in Sect. 4. First, we provide the datasets collected from Facebook and the subsequent process of its pre-treatment. Finally, we conclude the paper with future works.

2 Related Works

Owing to the limited space and our contribution which is a statistical approach (learning-based), we only describe relevant related work at this approach. Most studies have been focused on the statistical methods based on machine learning algorithms (Supervised methods). This kind of algorithm requires a set of well-classified sentences (manually labeled data: training corpus). Supervised methods aim to discover a model using labeled examples which must be able to generalize the classification learned on a wider dataset. Then, it comes to learning a machine how to assign a class to a new unlabeled sentence among the predefined classes in the learned model. The popular algorithms in machine

learning, such as Support Vector Machines [5, 13, 19] and Naive Bayes [6, 13], are used to train the sentiment classifier.

In order to perform machine learning, it is necessary to transform the text into numerical representations that may lead to correct classification. In the literature, the most common and useful feature is binary representation that indicates word presence or absence [12, 13]. **Binary** representation is very simple, but it is not very informative since it does not inform about the frequency of words which can be an important information. **Frequency-based** feature representation is a natural extension of the binary representation, where the number of occurrences of words in a sentence is counted. In sentiment classification, frequency representation has been used in several works, such as [14], but it presents the disadvantage of not taking into account the length of processed sentences and hence a long sentence may be represented by a vector whose norm is greater than that of the representation of a short sentence. It is therefore very interesting to work with a normalized frequency representation where each sentence is presented by a vector weighted by its size whose each component code the proportion of the term in the sentence.

[17] used Latent Semantic Analysis **LSA** which learns semantic word vectors by applying singular value decomposition to factor a term-document co-occurrence matrix. The **TF-IDF** representation attempts to be more informative than the previous representations (used by [9, 14]). The value TF (Term Frequency) corresponds to the frequency of a term in a sentence. It essentially refers to the importance of the term in this sentence. Nevertheless, the value IDF (Inverse Document Frequency) measures its importance in the set of sentences by calculating the logarithm of the inverse documentary frequency. [10] proposed a supervised variant of IDF weighting for sentiment analysis, named Δ **IDF**, in which the IDF calculation is done for each text class and then one value is subtracted from the other. They assigned feature values for a document by calculating the difference of those words TFIDF scores in the positive and negative training corpus.

Recently, another type of representation is proposed by [16], named Distributed vector representations which associate similar vectors with similar words and phrases. These vectors provide useful information for the learning algorithms to achieve better performance in Natural Language Processing tasks [11]. To compute the vector representations of words for sentiment analysis, [1] used the skip-gram model of **Word2Vec** [11]. The Skip-gram model aims to find word representations that are useful for predicting the surrounding words in a sentence or document [11].

[18] proposed **bag-of-sentiwords** vector representations of text which capture the presence of sentiment-carrying words derived from a sentiment lexicon. In other work, text has been represented as a bag-of-opinions, where features denote occurrences of unique combinations of opinion-conveying words, amplifiers, and negators [15]. Other features capture the length of a text segment, and the extent to which it conveys opinions [7].

3 Proposed Method

Most of the existing methods for sentiment classification based on machine learning algorithms use the bag-of-words representation of a text. Thus, these representations focus often on presence or frequency of lexicon words or specific words (frequent words, sentiment lexicon, opinion-conveying words,...). In this paper, we aim to determine the polarity of text by using a vector representation that uses words as well as emoticons.

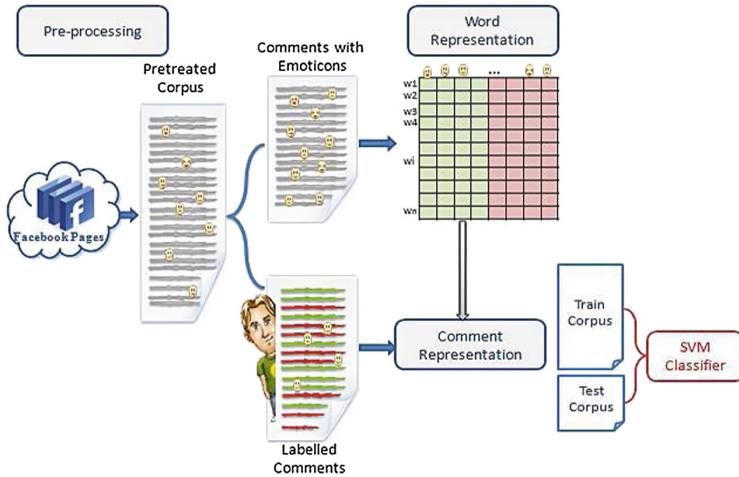


Fig. 1. The different steps of our sentiment classification method.

3.1 Vector Representation of Words

The statistical approach that we adopted in this paper, revolves essentially around the numerical representations of linguistic units (words and comments). These units are generally represented in a vector form which is characterized by associating each unit with its dimension within the vector space. In the statistical approach, the text is presented with a set of words that are considered as equivalent and unordered entities (bag-of-words). Thus, their semantic and sentimental aspects are not taken into account. Therefore, the main objective of our work is to find a representation that preserves the maximum of these two aspects.

In order to preserve the sentimental characteristic of comment, we need to represent it by an adequate vector which can faithfully translate the sentimental aspect of its words. Therefore, It is necessary to begin by associating a vector representing each word composing this comment.

Emotional Vector. In the text bearers sentiments, a word can have many features, but it is very difficult to select those which can give it a complete

sentimental representation. Unfortunately, the exact algorithm for finding best features does not exist. It is thus required to rely on our intuition and the domain knowledge for choosing a good feature. It is impossible to represent a word with a feature vector that contains all the words that exist in the language. For the simple reason that the vocabulary in Web 2.0 is very dynamic, handling new words is still not possible.

Due to the richness of the corpus with emoticons, we propose to represent each word according to the emotion symbols present in the comment. In fact, a word can have a different polarity degree with each emotion symbol. Therefore, we present word with a vector which takes account of the relations of a word with all of these symbols. The challenge of our method is the choice of emotion symbols that are appropriate to become features.

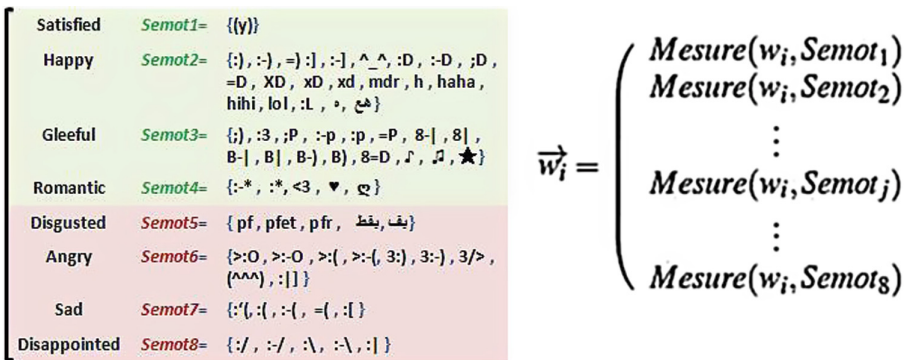


Fig. 2. The sets of used emotion symbols.

In the proposed vector representation, we granted to element number j of the vector representing a word, the value of its similarity to the set of emotion symbols $Semot_j$, with $j \in [1..8]$ (see Fig. 2). In the Fig. 2, we consider that each set of emotion symbols is an emotional state that can reach every word. Indeed, we distinguish 8 emotional states (satisfied, happy, gleeful, romantic, disappointed, sad, angry and disgusted). To do this, we collect the emotion symbols which determine each emotional state. The Fig. 2 shows that the 4 first sets of symbols *Semot* express positive sentiments, and the 4 following *Semot* express negative sentiments. Therefore, our emotional vector contains 8 dimensions. It represents the similarity degree between the word w_i and each set of emotion symbols (emotional state).

Eventually, it's noteworthy to define the effective similarity measures allowing to quantify sentimental relations between words and the emotion symbols. Here, we describe two similarity measures: Normalized co-occurrence and Emotional TF-IDF weighting.

– Normalized cooccurrence

This measure relies on the calculating of a number of co-occurrences of w_i and $Semot_j$ which is equal to the number of times that w_i and one of the $Semot_j$ appear together in the same comment. In order to highlight the similarities between the examined word and the emotion symbols having the same polarity as $Semot_j$, we normalized the number of co-occurrences of w_i and $Semot_j$ by the number of co-occurrences of all words and the emotion symbols having the same polarity as $Semot_j$ by using the following formula:

$$NormCO(w_i, Semot_j) = \frac{CoOcc(w_i, Semot_j)}{\sum_{k=f}^n CoOcc(w_i, Semot_k)} \quad (1)$$

where: $CoOcc(w_i, Semot_j)$ is the number of co-occurrences of the word w_i and the set of emotion symbols $Semot_j$. $f = 1$ and $n = 4$ (if $j \in [1..4]$), $f = 5$ and $n = 8$ (if $j \in [5..8]$).

– Emotional TF-IDF weighting

This measure is inspired from the TF-IDF measure (see Sect. 2) and it attempts to be more informative than the previous measure. Indeed, we propose this measure in order to take into account the distribution of emotion symbols in the comments of our corpus. Thus, it weights the $CoOcc(w_i, Semot_j)$ by the number of comments containing the emotion symbols $Semot_j$ in the corpus. It is calculated by using the formula 2.

$$Emotional_TFIDF(w_i, Semot_j) = CoOcc(w_i, Semot_j) \times \log \frac{N}{n_j} \quad (2)$$

where: $CoOcc(w_i, Semot_j)$ is the number of times that w_i and one of the $Semot_j$ appear together in the same comment. N is the total number of comments in the corpus containing emotion symbols. n_j is the number of comments that contain only emotion symbols of the $Semot_j$ set.

Thus, we can notice that, as in the emotional TF-IDF representation, $\log \frac{N}{n_j}$ will have a high value with the emotion symbols that appear in a few comments, and vice versa.

Up to this point, we proposed two emotional vector representations for words. These representations are based on similarity measures that are calculated according to the number of co-occurrences of word and emotion symbols, noted $CoOcc(w_i, Semot_j)$. In fact, if this number is important, we can deduce, then, that w_i has the same polarity as $Semot_j$, otherwise, w_i has a different polarity.

Negation Handling. The lexicon words are represented by emotional vectors containing 8 elements. Each element means the degree of similarity of the word with a set of emotion symbols $Semot$. The general principle of these representations is based on calculating the number of times where we encounter the considered word with an emotion symbol in a comment. However, a word can

be sometimes preceded by a negation particle (such as, ne, n, pas, ni, jamais, aucun, no, none, not, neither, never, ever). Hence, it is necessary to consider the presence of this type of information in the calculation of the number of co-occurrences, $CoOcc(w_i, Semot_j)$. It comes to cutting the comment in segments by considering the emotion symbols as separators. We thus assume that the number of co-occurrences will be decremented when we encounter the considered word preceded by a negation particle, with one of the emotion symbols in a comment segment. Thereby, we propose the Eq. 3 to calculate the number of co-occurrences of the word w_i and the set of emotion symbols $Semot_j$.

$$CoOcc = CoOcc(w_i, Semot_j) - CoOcc(w_i^{NEG}, Semot_j) \quad (3)$$

where: w_i^{NEG} is the word w_i when it is preceded by a negation particle.

3.2 Vector Representation of Comments

Once the vector representations of the words are ready, we can represent the comments in the same vector space (with a vector containing 8 elements). We proposed a simple strategy that involves attaching the feature vectors of the words that compose each comment of the corpus. We proceeded to represent the comments by averaging the vectors of all words which compose them and exist in the lexicon. The component V_C^j of the vector representing the comment is thus given by the following formula:

$$V_C^j = \frac{\sum_{i=1}^N s_{ij}}{N} \quad (4)$$

where N is the number of words constituting the comment C and each word w_i is represented by a vector $\vec{w}_i = (s_{i1}, s_{i2}, \dots, s_{ij}, \dots, s_{i8})$.

The comments can contain emotion symbols, it is therefore very important to consider them at the level of the vector representation step. For this reason, we considered the emotion symbol as a word by associating it with a binary vector. Thus, we assign the value 1 to the position $Semot_i$ of the vector, where an emotion symbol of the set of symbols i exists in the treated comment, ($i \in [1..8]$). For example, the emoticon :) will be represented by the vector: $(0, 1, \dots, 0)$.

Sometimes, a comment can contain unknown words that do not exist in the lexicon and are not represented by feature vectors. However, these words are indirectly represented by vectors whose components are null. At this point, we come to represent the comments of the corpus by two representations (Normalized co-occurrence and Emotional TF-IDF weighting). These representations are used in order to apply sentiment classification methods which described in the following section.

3.3 Comment Classification

In this paper, our main objective is to classify subjective texts (comments) into two classes (positive and negative). To achieve this objective, we propose two

classification methods: by summation of elements of its vectors and by using a supervised algorithm SVM.

Classification by Summation. We start by using the elements constituting the vector of the comment in a simple and intuitive way without recourse to classical classification methods. The idea is to combine the elements of the vector having the same polarity, in order to calculate the positive and negative scores (see 5). Then, we compare the two scores to determine the polarity of the comment.

$$Score_{pos} = \sum_{j=1}^4 s_{Cj}, \quad Score_{neg} = \sum_{j=5}^8 s_{Cj} \quad (5)$$

Supervised Classification with SVM. According to the literature, Naive Bayes classifier and Support Vector Machines are the most celebrated algorithms in the field of supervised classification which provide better results in most cases. We introduce in this paper a method which aims to classify texts (comments) as positive or negative using Support Vector Machines, a well-known and powerful tool for classification of vector of real-valued features [13]. To do this, we must perform labelling manually in advance and prepare the comments that will be used for the training step to obtain a model. The goal is to automatically identify the class of a new comment of the test corpus using the learned SVM-model.

SVM is a machine learning classification technique which uses a function called a kernel to map a space of data points in which the data is not linearly separable onto a new space in which it is, with allowances for erroneous classification. At the implementation level, we used the libsvm tool¹ developed and updated by [4]. This tool provides several kernel types, kernel parameters and optimization parameters. In order to achieve best performing SVM, we empirically adjust these parameters. We choose to classify data with polynomial kernel (defined as $(\gamma * u' * v + c)^\text{degree}$) type and change the parameter values related to this type (degree, gamma and coefficient).

4 Experiment Results and Discussion

To implement our proposed methods, we firstly prepare the dataset derived from Facebook. Indeed, we collect a set of comments from the political Tunisian pages. Then, we evaluate and validate the proposed sentiment classification methods by using external evaluation measure (F-score).

4.1 Facebook Dataset

Collection. The first step of our work is to collect huge amount of data, as shown in the Fig. 1. We chose to collect our corpus from social network Facebook in order to achieve a specific sentiment classification system to Facebook comment.

¹ SVM library available on the net (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>).

The choice of this corpus is not due to coincidence, but to the fact that the Facebook user uses the emotion symbols in the majority of their comments. Facebook provides an API² that brings us a simple and consistent information about the comment object. We extracted the comments from the political Tunisian pages in the period [1-Jan-2011, 1-Aug-2012]. We used 22 political pages among the most popular in Tunisia during the revolution period.

We present our collected corpus as a set of multilingual comments (Tunisian dialect, standard Arabic, French, etc.) organized in a well structured XML file. We decompose our corpus into two sub-sets of comments. The first is based only on comments containing emotion symbols (7000 comments), for generating the emotional vectors of lexicon words. The second is based on 3000 comments manually examined and annotated by an expert (1314 positive comments and 1686 negative comments). It serves to the evaluation and validation of our system classification and the proposed representations. We divided the set of comments into two disjoint sub-sets: 2100 comments used to perform the training step, and 900 comments used to perform the test step.

Pre-processing Corpus. Before beginning our sentiment classification process, it is necessary to pre-process and homogenize the corpus. This step aims to avoid the noise (presence of the words deemed unnecessary). Thereby, it allows us to select the most significant and relevant words which express the sentiment clearly. In this step, we performed: **Character normalization:** we replace specific characters with a space. Indeed, we prepared a set of unpronounced characters that haven't any influence on the sentimental information. These characters can be attached to words or differently listed as sequences. For this reason, we replace them with a space. **Filtering:** we avoid the hyperlinks to external resources and @target user in order to keep only the useful words that reflect the semantic and sentimental content of the comments. **Translation into French:** We prepare an automatic program which uses the translation tool (Google translator which is the most popular), in order to render all the comments written in the same language French and unify the future treatment. **Lemmatization:** It consists in encompassing the words which have the same primary entity (lemma). **Stopwords removal:** we prepared our own Stopwords file containing the empty words as grammatical words and linking words.

4.2 Evaluation Results

In this paper, we propose two emotional vector representations for text based on emotion symbols, namely: normalized co-occurrence and emotional TFIDF. In order to evaluate these representations and test our proposed sentiment classification methods, we used the external evaluation techniques by measuring the global efficiency F-score³ (see Table 1). From Table 1, we conclude that the best results are obtained with the handling of negation particles step. This step was

² <https://developers.facebook.com/docs/reference/apis/>.

³ <http://blog.onyme.com/apprentissage-artificiel-evaluation-precision-rappel-f-mesure/>.

one of the factors that contributed significantly to the efficiency of our classifier. In fact, negation handling aims to reverse the polarity of all words directly preceded by one of these particles. Despite its simple idea, we found that the negation handling step improves the results with the two proposed representations of text. We can also notice that the emotional TFIDF weighting provided superior results when compared against normalized co-occurrence.

Using the method of comment classification by summation of the elements of the vector having the same polarity, we obtained an efficiency of 72.71% with the emotional TFIDF weighting and 70.46% with the normalized co-occurrence vector. This shows that the performance of our grouping of emotion symbols that are used to form the vector features (emotional states which represent the positive and negative sentiments). However, when we use the SVM classifier, the vector representation of comments plays its proper role in the statistical classification approach. We achieve more important results than using the classification by summation. We obtained an efficiency of 81.08% with the emotional TFIDF weighting and 75.18% with the normalized co-occurrence vector.

Table 1. The efficiencies achieved by the summation **SUM** and **SVM** methods using different vector representations of words with and without negation handling.

	Normalized co-occurrence		Emotional-TFIDF weighting	
	Wo/Neg	W/Neg	Wo/Neg	W/Neg
SUM	35.50%	70.46%	45.44%	72.71%
SVM	54.64%	75.18%	60.11%	81.08%

To evaluate the effectiveness of our emotional vector representation based on emoticons, we compared them with baselines representations previously reported in the literature. To do this, we implemented several alternative bag-of-words vector representations that are conceptually similar to our own, as discussed in Sect. 2 (see Table 1):

Latent Semantic Analysis LSA: we used the package R ([lsa](https://cran.r-project.org/web/packages/lsa/lsa.pdf)⁴) which provides us with a sentence-term matrix.

Binary: We developed the program which takes, as input, the lexicon words and generates, as output, a binary sentence-word matrix (1: presence, 0: absent).

TF*IDF: we used the TFIDF score [17] to generate the sentence-word matrix.

TF* Δ IDF: we used the sentimental score proposed by [10] to generate the sentence-word matrix.

Word2Vec: we used the open-source distributed deep-learning library written in Java⁵, to compute the vector representations of words. In addition, we chose to apply the skip-gram model of word2Vec [11]. In fact, the skip-gram model aims to find word representations that are useful for predicting the surrounding

⁴ <https://cran.r-project.org/web/packages/lsa/lsa.pdf>.

⁵ <http://deeplearning4j.org/word2vec.html>.

words in a sentence. We constructed 100-dimensional word vectors for all lexicon words. Then, we compute the averaging of vectors representing the words which constitute a given comment. After this, supervised learning with SVM was performed using these vector representations.

Table 2. The efficiencies achieved by **SVM** classifier with different baseline vector representations and our emotional vectors.

SVM classifier	F-score
<i>Baselines features</i>	
LSA (Sentence-Word matrix)	67.24%
Bag of words (binary)	63.07%
TF*IDF Weighting	69.65%
TF* Δ IDF Weighting	71.78%
Word2Vec + Average	71.09%
<i>Our features</i>	
Normalized co-occurrence + Negation	75.18%
Emotional TF-IDF weighting + Negation	81.08%

Table 2 shows the obtained results (F-score) using SVM classifier with different representations of text based only on words (baseline feature) and our emotional vector representations based on emoticons. In baseline representation, we find that the TF* Δ IDF and word2vec work well in sentiment analysis problem. We also notice that our emotional vector representation clearly outperform those baseline representations. This shows that the usage of the emotion symbols in the text representation increases the performance of sentiment classification.

5 Conclusion and Future Work

In this paper, we presented a new vector representation of text for sentiment classification. First of all, we started by representing the words in an emotional vector based on the emoticons. The ultimate goal is to represent the different comments by numerical vectors capable of faithfully translate their sentimental orientations. Then, we proposed comment classification methods, namely: the classification by summing the vector elements and the supervised classification with the SVM. Finally, we presented the experimental results obtained by our proposed method. Our results are also effective and consistent. In the word representation step, we used only the comments having emotion symbols. Thus, we kept all infrequent words because these words have the chance to be known from other comments and to have higher frequencies. For this reason, it is very interesting to think of an enrichment step. Therefore, we thought, in future work, to take advantage of the other comments which haven't got any emotion symbol in order to adjust the feature vectors of words and comments and to add new words to the lexicon.

References

1. Alghunaim, A., Mohtarami, M., Cyphers, S., Glass, J.: A vector space approach for aspect based sentiment analysis. In: Proceedings of NAACL-HLT (2015)
2. Ameer, H., Jamoussi, S.: Dynamic construction of dictionaries for sentiment classification. In: 13th IEEE International Conference on Data Mining Workshops, ICDM Workshops, TX, USA, pp. 896–903 (2013)
3. Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. *IEEE Intell. Syst.* **28**, 15–21 (2013)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011)
5. Gezici, G., Yanikoglu, B., Tapucu, D., Saygn, Y.: New features for sentiment analysis: do sentences matter? (2012)
6. Harb, A., Plantié, M., Dray, G., Roche, M., Troussel, F., Poncelet, P.: Web opinion mining: how to extract opinions from blogs? In: Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology, New York, NY, USA, pp. 211–217 (2008)
7. Hogenboom, A., Bal, M., Frasinca, F., Bal, D., Kaymak, U., de Jong, F.: Lexicon-based sentiment analysis by mapping conveyed sentiment to intended sentiment. *Int. J. Web Eng. Technol.* **9**, 125–147 (2014)
8. Kamps, J., Marx, M., Mokken, R.J., de Rijke, M.: Using wordnet to measure semantic orientations of adjectives. In: Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation, vol. 4, pp. 1115–1118 (2004)
9. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, pp. 423–430 (2006)
10. Martineau, J., Finin, T.: Delta TFIDF: an improved feature space for sentiment analysis. In: Proceedings of the Third International Conference on Weblogs and Social Media, San Jose, California, USA, pp. 258–261 (2009)
11. Mikolov, T., tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of NAACL-HLT, pp. 746–751 (2013)
12. Pak, A., Paroubek, P.: Construction dun lexique affectif pour le franais partir de Twitter. In: TALN 2010. Université de Paris-Sud, Orsay Cedex, France (2010)
13. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on ACL, Stroudsburg, PA, USA, pp. 271–278 (2004)
14. Poirier, D.: Des textes communautaires à la recommandation. Ph.D. thesis, Université d’Orléans (2011)
15. Qu, L., Ifrim, G., Weikum, G.: The bag-of-opinions method for review rating prediction from sparse text patterns. In: Proceedings of the 23rd International Conference on Computational Linguistics, Stroudsburg, PA, USA, pp. 913–921 (2010)
16. Schütze, H.: Dimensions of meaning. In: Proceedings of Supercomputing 1992 (1992)
17. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. *J. Artif. Int. Res.* **37**, 141–188 (2010)
18. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 783–792 (2010)
19. Wilson, T., Wiebe, J., Hwa, R.: Just how mad are you? Finding strong and weak opinion clauses. In: Proceedings of the 19th National Conference on Artificial Intelligence, San Jose, California, pp. 761–767 (2004)