2017 IEEE/ACS 14th International Conference on Computer Systems and Applications

Sentiment lexicon enrichment using emotional vector representation

Hanen Ameur*, 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, Sfax-Tunisia 3021
*Email: ameurhanen@gmail.com
†Email: salma.jamoussi@isimsf.rnu.tn
‡Email: abdelmajid.benhamadou@isimsf.rnu.tn

***Abstract*—With the explosive growth of social media sites, an enormous amount of opinionated data has become numerically documented online. The explosion of this kind of data gives rise of new areas: sentiment analysis and opinion mining. Our purpose in this paper is to determine the polarity of Facebook comments "positive or negative" using machine learning-based approach. This approach requires an adequate vector representation of linguistic units (words or sentences) to obtain better classification performance. For this reason, we present a new emotional vector representation of words which can faithfully translate their semantic and sentimental characteristics, based on emotion symbols. Furthermore, we propose an enrichment step that aims to enlarge the lexicon and dynamically update word representation according to new contexts. Finally, by using these word representations, we present the comments in the same feature space to classify them and predict their polarity. We use Support Vector Machines to show that our emotional vector representation of lexicon's word and its enrichment significantly improves accuracy for sentiment analysis problem compared with the well-known bag-of-words vector representations, using dataset derived from Facebook. Our results are also consistent and effective.**

***Index Terms*—Sentiment classification; Emotional Tf-IDF; Dynamic enrichment; Emotion symbols; Facebook comments;**

I. Introduction

Social media sites (like Facebook, Twitter, etc.) has become an essential tool for people to express and share their sentiments and opinions about their daily lives and exchange with others their point of views towards different topics. Nowadays, the ever-increasing use of these social media sites leads to a very huge amount of online opinionated data in the form of tweets, comments, reviews, etc. The explosion of this kind of data gives opportunities to researchers to extract and analyze the user generated content. Recently, two main research areas, namely sentiment analysis and opinion mining, have been actively developing in order to assign a polarity label to a given texts (binary "positive or negative" or multivariate).

Sentiment analysis can be broadly classified into lexicon-based approach "linguistic" and machine learning-based approach "statistical". The lexicon-based approach requires the construction of an opinionated lexicon of labelled words to determine the semantic orientation "valence of polarity" of a larger text. There are five kind of lexicon-based methods, namely: manual method [1], dictionaries-based method [2], corpora-based method [3], hybrid method [4], concept-based method [5]. The statistical approach focuses on using machine learning methods (like Support Vector Machines, Naive Bayesian classifiers, Maximum Entropy, Neural Networks, etc.) to classify and identify the class labels for texts. The machine learning-based approach presents better capability than the lexicon-based approach [6]. However, this approach adopts machine learning classification techniques which have some limits [7], mainly the selection of effective features "best representation" that faithfully captures the semantic and sentimental characteristics of texts to obtain better classification performance.

In machine learning-based approach, feature selection is still a critical task. The majority of existing methods follows [8] using bag-of-words feature representation. The words can be represented for instance as binary (presence or absence of lexicon words) [6], frequency (number of cooccurrences), weighted by their Tf-IDF score (using lexicon words) [9], or TF-deltaIDF score (using opinionated positive and negative words) [10]. Moreover, it seems impossible to represent a word with a feature vector that covers all the words that exist in the language. Especially, in data gathered from social media site, the vocabulary is very dynamic and rapidly varied. So, the handling of new words is very difficult. For this reason, we address in this paper the problem how to represent words by sentimental feature vector representation that able to be update dynamically depending the context. Therefore, we introduce in this paper an emotional TF-IDF to represent words based on eight different emotional states that words can express for formulating the feature vector space. We present each emotional state as a collection of emotion symbols (e.g. emoticons). Furthermore, we propose an algorithm of dynamic enrichment lexicon that preserves the same emotional feature of words by updating and improving word representation accordingly their new context, as well as it allows to add all unknown words encountered at the arriving of new data. We perform the evaluation using Facebook comment dataset collected by [11] to see the effect of dynamic enrichment of sentiment lexicon using emotional vector representation at the level of polarity determination of word and sentence.

The paper is organized as follows: We review in the next section the related work for sentiment analysis using machine

2161-5330/17 $31.00 © 2017 IEEE
DOI 10.1109/AICCSA.2017.151

learning-based methods. We then elaborate on the principle of our proposed method for sentiment classification in section 3. We present the proposed emotional Tf-IDF representation and the technique of sentiment lexicon enrichment. Next, in section 4 we report the experimental results. Finally, we conclude the paper with future works.

II. Related Work

Before starting to propose and evaluate our sentiment classification method, we firstly study the existing works. In the literature, a fairly significant number of studies have been tried to address the problem of automatic sentiment analysis.

In this section, we present an overview of different methods proposed to perform sentiment analysis task using machine learning algorithms (Supervised methods), where sentiment analysis is formalized as a classification task. It involves classifying opinions in text into categories, like positive or negative, by using, for example Support Vector Machines ([6], [1] and [12]), Naive Bayes Classifiers ([6] and [13]), Maximum Entropy [6], Softmax [14], etc. 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.

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, various types of features are used by existing machine learning methods to sentiment analysis in order to construct vector representations of text. We distinguish two types of text representation; namely: the bag-of-words vector representation which considers the text as a set of words without taking into account the order of their appearance and the sequential representation which preserves the order of words contained in a sentence.

bag-of-words Representation

The most common and most useful feature is binary representation that indicates word presence or absence [6], [4]. **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 [8], [15] and [16], 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] use 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] and [16]). 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 [18], 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 [19]. To compute the vector representations of words for sentiment analysis, [20] used the skip-gram model of **Word2Vec** [19]. The Skip-gram model aims to find word representations that are useful for predicting the surrounding words in a sentence or document [19].

[21] 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 [22]. Other features capture the length of a text segment, and the extent to which it conveys opinions [23].

Sequential Representation

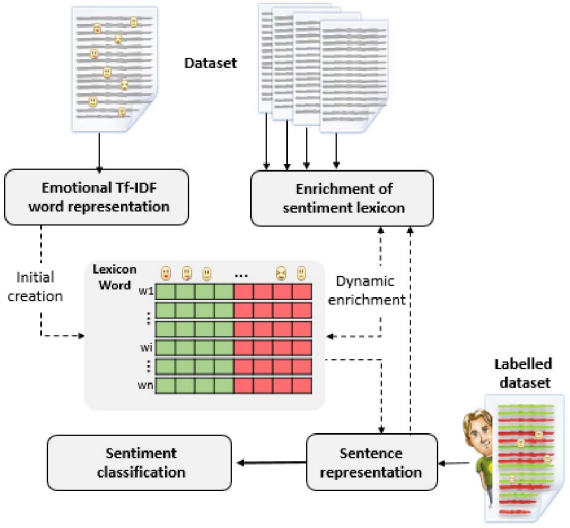
This representation emphasizes on the notion of **n-grams**. The n-gram is a notion derived from the model language and it is often used in the sentiment classification methods due to its simplicity and efficiency. In fact, in the literature, the employments of n-grams in sentiment classification has been debatable. The n-gram can be considered as a single word. At the film critics classification by polarity, [8] showed that the best result was obtained using only the uni-gram method. Nevertheless, [24] have demonstrated that in the same situation, the use of bi-grams and tri-grams give better results.

III. Proposed method

Machine learning-based sentiment analysis methods require essentially an adequate numerical representation of linguistic units (word, sentence or document) for classifying them into two categories (positive and negative). These units are represented with a feature vector space using generally the bag-of-words representation which focus often on presence or frequency of lexicon words or specific words (like frequent words, sentiment lexicon, opinion-conveying words, etc.). These representations consider the words as equivalent and

unordered entities without taken into account their semantic and sentimental aspects. Hence, in this paper, we propose a new feature vector representation (named, emotional Tf-IDF) that preserves the maximum of these two aspects based on words as well as emotions symbols, in order to determine the polarity of a given text (comments).

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



The proposed method, as shown in figure 1, is depicted in four major steps: (1) lexicon word representation using an improved emotional Tf-IDF weighting based on emotion symbols (:), :(, etc.), (2) dynamic enrichment of sentiment lexicon at each new arriving sentence, the word representation vector is updated and other words are added, (3) sentence representation using semantic compositional operation and (4) sentiment classification of sentences as positive and negative.

As we applied the semantic compositional operation to represent a sentence, we need to represent its words by vectors that can faithfully translate their semantic and sentimental aspects. Therefore, it is necessary to begin by associating a sentimental vector representing each word composing a given sentence.

In the following sections, we present our methodology to represent words. It is composed of two steps. The first one aims to generate the initial word representation using a new emotional Tf-IDF weighting based on comments containing emotion symbols. The second one is considered as a dynamic step that consist in enriching the lexicon by other words and updating the known words, at the arriving of a new comment. We use in the second step, the rest of comments that haven't emotions symbols.

A. Emotional TF-IDF weighting for word representation

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 impossible.

Due to the richness of the corpus with emoticons, we propose to represent each word according to the emotion symbols present in the comments. In fact, a word can have a different polarity degree with each emotion symbol. Therefore, we present lexicon words with vectors taking into account the relations between a word and all of emotion symbols. The challenge of our method is to choice the relevant emotion symbols that are appropriate to become features. To do so, we collect the emotion symbols most used in social media sites and we regroup them. We consider each set of emotion symbols as an emotional state that can reach every word. In reality, the human sentiments are not limited to positive and negative expressions, but they contain several emotional states [11]. In our case, we distinguish 8 emotional states (satisfied, happy, gleeful, romantic, disappointed, sad, angry and disgusted). Hence, we propose to represent lexicon word by an emotional vector which contains 8 feature values. Each value qualifies the degree of correlation (similarity) between the word w_i and the corresponding set of emotion symbols (emotional state), as shown in figure 2.

Fig. 2. The sets of used emotion symbols.

Satisfied	*Semot1*=	{(y)}
Happy	*Semot2*=	{:) , :-) , =) :] , :-] , ^_^, :D , :-D , ;D , =D , XD , xD , xd , mdr , h , haha , hihi , lol , :L , ه, ضع }
Gleeful	*Semot3*=	{;) , :3 , ;P , :-p , :p , =P , 8-| , 8| , B-| , B| , B-) , B) , 8=D , ♪ , ♫ , ★}
Romantic	*Semot4*=	{:-* , :*, <3 , ♥ , ج }
Disgusted	*Semot5*=	{ pf , pfet , pfr , يقف, يقط}
Angry	*Semot6*=	{>:O , >:-O , >:(, >:-(, 3:) , 3:-) , 3/> , (^^^) , :[ ] }
Sad	*Semot7*=	{:'(, :(, :-(, =(, :[}
Disappointed	*Semot8*=	{ :/ , :-/ , :\ , :-\ , :| }

Here, we describe our emotional vector representation, called emotional Tf-IDF, inspired from the Tf-IDF measure. It is calculated using the equation 1. The emotional Tf-IDF measure relies on calculating of a number of co-occurrences of w_i and $Semot_j$, noted $CoOcc(w_i, Semot_j)$ where it is weighted by the number of comments containing the emotion symbols $Semot_j$ in the corpus. In fact, the emotional Tf-IDF takes into account the distribution of emotion symbols in the comments of the corpus, see the equation 1.

$$EmotionalTFIDF(w_i, Semot_j) = CoOcc(w_i, Semot_j) \times \log \frac{N}{n_j} \quad (1)$$

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. Thus, if the $CoOcc(w_i, Semot_j)$ value is important, we can deduce, then, that w_i has the same polarity as $Semot_j$, otherwise, w_i has a different polarity. During the calculation of $CoOcc$ measure, it comes to cutting the comment into segments by considering the emotion symbols as separators, and to increment the number of co-occurrences of w_i and one of the $Semot_j$ by 1. 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 kind of information in the calculation of $CoOcc(w_i, Semot_j)$. To do so, we decrement the number of co-occurrences $CoOcc$ by 1 when we encounter the considered word preceded by a negation particle, with one of the emotion symbols in a comment segment. Thereby, the number of co-occurrences of the word w_i and the set of emotion symbols $Semot_j$ will be calculated using the following equation.

$$CoOcc = CoOcc(w_i, Semot_j) - CoOcc(w_i^{NEG}, Semot_j) \tag{2}$$

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

Up to this point, an initial lexicon words is generated from the comments containing emotion symbols, where each word is represented by a vector of 8-dimension. The element number j is the value of word's similarity to the set of emotion symbols $Semot_j$, with $j \in [1..8]$ (see equation 3).

$$w_i = \begin{pmatrix} EmotionalTFIDF(w_i, Semot_1) \\ EmotionalTFIDF(w_i, Semot_2) \\ EmotionalTFIDF(w_i, Semot_3) \\ \vdots \\ EmotionalTFIDF(w_i, Semot_j) \\ \vdots \\ EmotionalTFIDF(w_i, Semot_8) \end{pmatrix} \tag{3}$$

B. Enrichment of sentiment lexicon

At this step, we take the results of the previous step as a start-up to dynamically enrich the lexicon words at the arriving of new comments. In fact, we distinguish two essential objectives for the enrichment step. On the first hand, it aims to broaden the vocabulary by adding new words bearing sentiments and not taken into account in the previous step. On the other hand, it consists in improving and adjusting the word vector representation that present in the lexicon.

To achieve this dual objective, we propose to use the other comments that haven't emotions symbols as a new input batch-data. So, we cut the comment in segments by considering the punctuation symbols (like ".", "?") as separators. Thus, in each segment, a word is influenced and impacted by the neighbor words (its context). For this reason, we rely on the vector representation of a sentence (segments of a comment) based only on the vectors of the known words that present in the lexicon.

Adjustment of vector representation of existing words: it comes to highlight the new context where the examined word can be appeared. To do so, we propose to update the vector representing a word by applying the equation 4. It is a sort of weighting. In fact, we added the sentence vector representation V_C to the examined word vector W_i, multiplied by a β variable (see equation 4). Following a sequence of experiences, we were able to choose the value of α equals to 0.95 and the value of β equals to 0.05.

$$s_{ij} = \alpha \ s_{ij} + x \times \beta V_C^j \tag{4}$$

where:
- $\alpha + \beta = 1$
- $x = \begin{cases} -1 & \text{if the word is preceded by a negation particle.} \\ 1 & \text{else.} \end{cases}$
- $w_i = \begin{pmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ s_{ij} \\ \vdots \\ s_{i8} \end{pmatrix}$ and $V_C = \begin{pmatrix} V_C^1 \\ V_C^2 \\ \vdots \\ V_C^j \\ \vdots \\ V_C^8 \end{pmatrix}$

Adding new words: the new arriving comments are often containing new words that are not appear in the previous comments (not present in the lexicon). So, we propose a strategy to treat these new words by adding them to the lexicon of words. Generally, the word depends on the sentiment bearers by the other words present in the comments. For this reason, the new words must be represented by vectors which quantify their sentimental relation with the other words already present in the lexicon. Therefore, the idea is to calculate the distance between the new word (examined word) and the others known words using the equation 5. It comes to count the number of words that exist between the examined words and the other comment's words (see figure 3).

$$dist(w_i, w_m) = \frac{1}{n+1} \tag{5}$$

Where: n is the number of words present between the known word w_i and the examined word w_m.

In fact, when two words are more distant, the correlation and the degree of similarity between them are lower. Hence, it is necessary to weight each word of comment by its distance with the examined word in order to emphasize its correlation degree during the vector representation of the examined word. In the formal way, a new word will be represented by a vector w_m

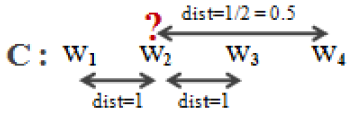


Fig. 3. The distance between the unknown word and the other comment's words.

in the same feature space (8-dimension) where each element is determined as follows.

$$s_{mj} = x \times \sum_{i=1}^{N} \frac{dist(w_i, w_m)}{\sum_{i=1}^{N} dist(w_i, w_m)} \times s_{ij} \quad (6)$$

Where:
- N is the number of words constituting the comment C which exist in the lexicon.
- $x = \begin{cases} -1 & \text{if the word is preceded by a negation particle.} \\ 1 & \text{else.} \end{cases}$

C. Sentence vector representation

So far, the lexicon words are represented with an emotional Tf-IDF representation that captures their semantic and sentimental characteristic, as well as it can dynamically enlarged at the arriving of new sentences. In this section, after obtaining a 8-dimensional vector for each word, we propose to represent sentences (comments) in the same vector space (with a vector containing 8 elements). We apply a simple semantic compositional operation to represent a sentence that involves attaching the vectors of the words that compose each comment. We represent the comments by averaging the vectors of all words which compose them and exist in the lexicon.

The jth ($j \in [1..8]$) element of comment vector V_C is thus calculated as expressed in the following equation:

$$V_C^j = \frac{\sum_{i=1}^{N} s_{ij}}{N} \quad (7)$$

where the comment C is composed of N words and each word w_i is represented by a vector $\overrightarrow{w_i} = (s_{i1}, s_{i2}, \ldots, s_{ij}, \ldots, s_{i8})$.

Finally, a comment is represented by vector that is defined by the equation 8.

$$V_C = \begin{pmatrix} V_C^1 \\ V_C^2 \\ \vdots \\ V_C^j \\ \vdots \\ V_C^8 \end{pmatrix} \quad (8)$$

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 directly represented by vectors whose components are null. The unknown words are automatically enriched in the lexicon (see section B). All these proposed sentence representations are used in order to apply sentiment classification methods which described in the following section.

D. Sentiment classification

This step consists in using the emotional vector representation of sentences, in order to classify and determine the polarity of them (positive or negative). Many supervised machine learning algorithms such as, Naive Bayes classifier and support vector machines, are used in the field of sentiment classification. In this paper, we adopt the SVM classifier. In fact, according to the literature, the SVMs is a well-known and powerful tool for classification of real-valued features which yield good results for the sentiment analysis task [25]. The SVMs algorithm requires a manually labeling data for training and testing using a learned SVM-model. SVM uses a function called a kernel to map a space of points in which the data is not linearly separable onto a new space in which it is, with allowances for erroneous classification [26].

In order to implement the SVMs classifier, we use the caret R-package[1] which provides several kernel types, kernel parameters and optimization parameters. We classify data with svm-radial-weights kernel method type. To choose the optimal tuning parameter values (sigma, cost and weight), we split using "train" function[2], the training set into training and validation sub-sets using 10-fold cross validation re-sampling function and we select the specific parameter values according to the performance measures of each model.

IV. Experiment results and discussion

In this paper, we are concerned about representing word lexicon with a compact representation "emotional Tf-IDF", based on emotion symbols. This representation can be updated and improved according new contexts using an enrichment step. We present here our experiments and the achieved results using our sentiment analysis method. We evaluate the proposed representation of words and comments compared to baseline representations. We also describe the impact of enrichment step on word representation.

A. Facebook dataset

The corpus used in the experiments was the comments derived from Facebook, collected by [11]. Facebook comment corpus is extracted from the political Tunisian pages in the period [1-Jan-2011, 1-Aug-2012] where it is written in multilingual form (Tunisian dialect, standard Arabic and French). Facebook corpus is decomposed into two disjoint sub-sets of comments. The first serves to generate the emotional vectors of lexicon words "learning corpus of lexicon" containing 57 millions of comments (7 millions contain emotion symbols and the rest haven't emoticons). The second sub-set is based on 3000 comments manually examined and annotated by an

[1]We used these two links: http://topepo.github.io/caret/index.html https://cran.r-project.org/web/packages/caret/caret.pdf

[2]The caret package offers us the "train" function which sets up a grid of tuning parameters for classification, fits each model and calculates a re-sampling based performance measure.

expert (1314 positive comments and 1686 negative comments). It is intended for the evaluation and validation of our sentiment classification system and the proposed representation (2100 comments used to perform the training step and 900 comments used to perform the test step).

Before beginning our sentiment classification process, we performed a preprocessing step of comments:

- Character normalization: we replace the unpronounced characters (that haven't any influence on the sentimental orientation like "◇") 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 Stop-words file containing the empty words as grammatical words and linking words.

B. Results and interpretations

This section is reserved to present the different experiments' conducted to evaluate our sentiment analysis method. To do so, we used the external evaluation techniques[3] (recall, precision and F-score) to measure the adequacy of our system classification and that made by the experts.

TABLE I
THE EFFICIENCIES "F-SCORE" ACHIEVED BY SVM CLASSIFIER USING INITIAL AND ENRICHED LEXICON.

Lexicon		Recall	Precision	F-score
Emotional TF-IDF (Initial)	**Wo/ Neg**	59.42%	60.81%	60.11%
	W/ Neg	79.92%	82.27%	81.08%
Emotional TF-IDF (Enriched)	**Wo/ Neg**	62.86%	67.12%	64.92%
	W/ Neg	82.88%	85.58%	84.21%

Table I presents the obtained results with (w/Neg) and without (wo/Neg) handling negation using initial and enriched sentiment lexicon. From these results, we can notice the interest of handling negation particles included in the comments, and more importantly, the usefulness of enrichment step. In fact, the enrichment step well improves the quality of sentiment classification, whatsoever with or without handling of negation particles. Thus, we found that the negation handling step was one of the factors that contributed significantly to the sentiment classification performance. Furthermore, the sentence representation based on enriched lexicon is more adequate than representation based on initial lexicon. So, we obtain a F-score of 84.21% using enriched lexicon.

Table II shows clearly the impact of enrichment step from a quantitative viewpoint. In fact, it allows to enrich the lexicon by other words.

TABLE II
THE NUMBER OF WORDS IN THE INITIAL AND ENRICHED LEXICONS.

Lexicon	Initial	Enriched
Number of words	17390	100753

To evaluate the effectiveness of our emotional representations using initial and enriched lexicon based on emotional state of words, we compared their results with those obtained by baseline representations that are conceptually similar to our own, as reported in section 2. In table III, we illustrate the experimental results, obtained using our proposed method and the following sentiment classification methods:

- **Binary**: we represented each sentence with binary representation [6]. So, we developed a program which takes, as input, the lexicon words and generates, as output, a binary sentence-word matrix (1: presence, 0: absent). Then, we built the sentiment classification with Supported Vector Machines.
- **LSA** (Latent Semantic Analysis): we used the package R "lsa[4]" which provides a sentence-term matrix.
- **TF-IDF** (Term Frequency-Inverse Document Frequency): we applied the TF-IDF score [17] to generate sentence-word matrix.
- **TF-ΔIDF**: 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 Java [5], to compute the vector representations of words. In addition, we chose to apply the skip-gram model of word2Vec [19]. In fact, the skip-gram model aims to find word representations that are useful for predicting the surrounding words in a sentence. We used the training corpus to train the word embedding and we have taken into account lexicon words that occur in the text more than a threshold frequency (equals to 4). We constructed 100-dimensional word vectors for 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 III shows the performance of baseline representation (based only on words) and our emotional representation (based on emotion symbols). In baseline representations, binary representation performs very poor as it loses the ordering of

[3]http://blog.onyme.com/apprentissage-artificiel-evaluation-precision-rappel-fmesure/

[4]https://cran.r-project.org/web/packages/lsa/lsa.pdf

[5]http://deeplearning4j.org/word2vec.html

TABLE III
THE EFFICIENCIES ACHIEVED BY SVM CLASSIFIER WITH DIFFERENT BASELINE VECTOR REPRESENTATIONS AND OUR EMOTIONAL VECTORS.

SVM Classifier	Recall	Precision	F-score
Baselines Features			
Bag of words (binary)	52.80%	78.30%	63.07%
LSA (Sentence-Word matrix)	66.84%	67.64%	67.24%
TF-IDF Weighting	68.84%	70.48%	69.65%
TF-ΔIDF Weighting	65.91%	78.79%	71.78%
Word2Vec + Average	66.41%	76.48%	71.09%
Proposed Features			
Emotional TF-IDF weighting (Initial)	79.92%	82.27%	81.08%
Emotional TF-IDF weighting (Enriched)	82.88%	85.58%	84.21%

words. It also ignores the semantics of words. TF-IDF and TF-ΔIDF perform better than binary representation because they are more informative than binary representation where the number of occurrences of words in a sentence (whatsoever positive or negative) is counted. We can see also that our emotional representation clearly outperform those baseline representation. We obtain consistent and encouraging results, 81.08% based on initial lexicon and 84.21% based on enriched lexicon. These results prove the performance of our grouping of emotion symbols that are used to form the vector features (emotional states) in initial lexicon. In addition to that, the efficiency of the enrichment step with allows to improve word representation depending new contexts and enlarge the lexicon by new encountered words.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a new method of sentiment lexicon enrichment using emotional vector representation. First of all, we started by representing lexicon's words by an emotional Tf-IDF vector based on comments containing emotions symbols. Thereafter, using the rest of comments, we proposed an enrichment step to enlarge the lexicon by other words and update the representations of existing words depending new contexts. Then, we represented the comments by an emotional vector in the same feature space by averaging the vector representing all words that compose them and exist in the lexicon. We used SVMs classifier to determine the polarity "positive or negative" of comments. Finally, we discussed the experimental results obtained by our proposed method. Our results are also effective and consistent. In future work, we would like to consider other sentimental classes. We propose to use more sentimental corpus out of the experimental datasets and treat different languages without translating data into a unified language. We aims also to apply our word representation on data streem analysis task.

REFERENCES

[1] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? finding strong and weak opinion clauses," in *Proceedings of the 19th National Conference on Artifical Intelligence*, San Jose, California, 2004, pp. 761–767.

[2] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke, "Using wordnet to measure semantic orientations of adjectives," in *Proceedings of LREC-04, 4th international conference on language resources and evaluation*, vol. 4, 2004, pp. 1115–1118.

[3] H. Ameur and S. Jamoussi, "Dynamic construction of dictionaries for sentiment classification," in *13th IEEE International Conference on Data Mining Workshops, ICDM Workshops*, TX, USA, 2013, pp. 896–903.

[4] A. Pak and P. Paroubek, "Construction dun lexique affectif pour le franais partir de twitter." in *TALN 2010*, Université de Paris-Sud, Orsay Cedex, France, 2010.

[5] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, vol. 28, pp. 15–21, 2013.

[6] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42Nd Annual Meeting on ACL*, Stroudsburg, PA, USA, 2004, pp. 271–278.

[7] W. Rong, Y. Nie, Y. Ouyang, B. Peng, and Z. Xiong, "Auto-encoder based bagging architecture for sentiment analysis," *Journal of Visual Languages and Computing*, vol. 25, no. 6, pp. 840–849, 2014.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02, Stroudsburg, PA, USA, 2002, pp. 79–86.

[9] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA, 2006, pp. 423–430.

[10] J. Martineau and T. Finin, "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, 2009, pp. 258–261.

[11] H. Ameur, S. Jamoussi, and A. B. Hamadou, "Exploiting emoticons to generate emotional dictionaries from facebook pages," in *KES International Conference on Intelligent Decision Technologies (KES-IDT 2016)*, Tenerife, Spain, 2016, pp. 39–49.

[12] G. Gezici, B. Yanikoglu, D. Tapucu, and Y. Saygn, "New features for sentiment analysis: Do sentences matter?" pp. 5–15, 2012.

[13] A. Harb, M. Plantié, G. Dray, M. Roche, F. Trousset, and P. Poncelet, "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, 2008, pp. 211–217.

[14] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, N. Andrew, and P. Christopher, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, 2013, pp. 1631–1642.

[15] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[16] D. Poirier, "Des textes communautaires à la recommandation," Ph.D. dissertation, Université d'Orléans, 2011.

[17] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, pp. 141–188, 2010.

[18] H. Schütze, "Dimensions of meaning," in *Proceedings of Supercomputing '92*, 1992.

[19] T. Mikolov, W. tau Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of NAACL-HLT*, 2013, pp. 746–751.

[20] A. Alghunaim, M. Mohtarami, S. Cyphers, and J. Glass, "A vector space approach for aspect based sentiment analysis," in *Proceedings of NAACL-HLT*, 2015.

[21] H. Wang, Y. Lu, and C. Zhai, "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, 2010, pp. 783–792.

[22] L. Qu, G. Ifrim, and G. Weikum, "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, 2010, pp. 913–921.

[23] A. Hogenboom, M. Bal, F. Frasincar, D. Bal, U. Kaymak, and F. de Jong, "Lexicon-based sentiment analysis by mapping conveyed sentiment to

intended sentiment," *Int. J. Web Eng. Technol.*, vol. 9, pp. 125–147, 2014.
[24] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th International Conference on World Wide Web*, 2003.
[25] S. Wang and C. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ser. ACL '12, Stroudsburg, PA, USA, 2012, pp. 90–94.
[26] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, July 2004, pp. 412–418, poster paper.