Hindawi
Complexity
Volume 2021, Article ID 6152494, 11 pages
https://doi.org/10.1155/2021/6152494

Review Article

A Training-Optimization-Based Method for Constructing Domain-Specific Sentiment Lexicon

Maokang Du, Xiaoguang Li, and Longyan Luo

Key Laboratory of Electronic Commerce and Logistics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Xiaoguang Li; lixiaoguang.self@gmail.com

Received 4 July 2020; Revised 27 September 2020; Accepted 25 January 2021; Published 8 February 2021

Academic Editor: Zong Woo Geem

Sentiment analysis has been widely used in text mining of social media to discover valuable information from user reviews. Sentiment lexicon is an essential tool for sentiment analysis. Recent research studies indicate that constructing sentiment lexicons for special domains can achieve better results in sentiment analysis. However, it is not easy to construct a sentiment lexicon for a specific domain because most current methods highly depend on general sentiment lexicons and complex linguistic rules. In this paper, the construction of sentiment lexicon is transformed into a training-optimization process. In our scheme, the accuracy of sentiment classification is used as the optimization objective. The candidate sentiment lexicons are regarded as the individuals that need to be optimized. Then, two genetic algorithms are specially designed to adjust the values of sentiment words in lexicon. Finally, the best individual evolved in the presented genetic algorithms is selected as the sentiment lexicon. Our method only depends on some labelled texts and does not need any linguistic knowledge or prior knowledge. It provides a simple and easy way to construct a sentiment lexicon in a specific domain. Experiment results show that the proposed method has good flexibility and can generate high-quality sentiment lexicon in specific domains.

1. Introduction

Currently, it has become very convenient for people to express opinions and share knowledge through the Internet. With the continue increasing of online reviews on the Internet, sentiment analysis is becoming a research hot issue in natural language processing. To provide better personalized service, sentiment analysis has been applied to extract user opinions from the comments on the Internet [1]. In most methods of sentiment analysis, sentiment lexicon is an essential tool. Sentiment lexicon includes sentiment words or phrases and their intensity value and polarity [2]. Some recent research studies find that different domains have their special sentiment words, and these special sentiment words usually play an important role in sentiment analysis [3]. Accordingly, how to construct the sentiment lexicon in specific domains is very important to sentiment analysis, which becomes a hot issue in recent years.

The traditional unigrams sentiment lexicon is primarily collected by experts and annotated manually. Some generally used lexicons have been proposed, such as General Inquirer (GI) [4] and SentiWordNet [5]. Although these methods provide a good foundation to construct sentiment lexicon, the number of sentiment words in the lexicon is small and the coverage of sentiment words is limited. To improve the coverage of sentiment words, researchers proposed some methods to extend the traditional sentiment lexicons [6, 7]. Unigrams sentiment lexicons, degree adverbs, and negative words are combined together to construct n-grams sentiment lexicons [2]. However, the n-grams lexicons constructed by these methods only provide the sentiment polarity of n-grams but do not quantitatively describe their sentiment intensity. Moreover, due to the limitation of basic lexicon, these methods cannot generate new sentiment words, which have never appeared in the basic lexicon.

In some cases, the value and polarity of the same sentiment word may change in different domains. In order to better satisfy the requirement of sentiment analysis, it is necessary to construct sentiment lexicons by considering domain traits. There are some efforts to construct domain-specific sentiment lexicons. A common idea is to use the value of the sentiment word in a general lexicon as the basic value and then automatically adjust the basic value according to the corpus of the target domain [8, 9]. However, most of methods based on this idea is related to the background knowledge of domain and require a large number of extra annotations by linguists. Thus, these methods have no universal property and can hardly construct lexicon for other domains.

In recent years, some methods based on machine learning have been proposed to construct sentiment lexicon in specific domain [10–12]. These methods can directly extract sentiment words from the corpus of a specific domain by learning the features of the text. The main advantage of machine learning methods is that they do not depend on the linguistic knowledge and the domain knowledge and have good universality. Moreover, these machine learning methods can generate new sentiment words that are not related to the basic lexicon and break the limitation of basic lexicon. Inspired by the idea of machine learning, we proposed a training-optimization-based method to automatically construct domain-specific sentiment lexicons. Different to the machine learning methods, we regard the process of learning as an optimization process and design genetic algorithms to optimize the intensity values of sentiment words. Compared with machining learning methods, our method has a simple computational structure and high running speed. The main innovation and contributions of this research are as follows:

(i) A framework of constructing the sentiment lexicons for specific domains is proposed, in which the construction of sentiment lexicon is converted into a training and optimization process

(ii) Our method extracts sentiment words from the short texts collected in the target domain, which breaks the limitation of seed lexicon and effectively improves the coverage of sentiment words of specific domains

(iii) We specially design two genetic algorithms to optimize the sentiment lexicon, which makes it possible to automatically adjust the intensity values of sentiment words according to the domains. Our method has good universality and can be used in any domains

The remaining parts of this paper are organized as follows. In Section 2, a brief review of related works in sentiment analysis is given. Section 3 represents the framework of training-optimization and the proposed algorithms for constructing sentiment lexicon. The advantage of our scheme is described in Section 4. In Section 5, we present the experimental results. Finally, conclusions are drawn in Section 6.

2. Related Work

Sentiment lexicon is an important basic component for sentiment analysis, which also has an important influence on the results of sentiment classification. There are some famous sentiment lexicons built by the experts before, such as General Inquirer [4] and SentiWordNet [5]. Although these general sentiment lexicons have achieved certain success in text sentiment analysis, the coverage of universal sentiment lexicon is not comprehensive for some specific domains. To extend the coverage of sentiment lexicon, Wang et al. selected some sentiment words from the existing sentiment lexicon as seed and used the PU learning method to generate new sentiment words [6]. Zhang et al. constructed a comprehensive sentiment lexicon by collecting network words and emoticons widely used in Chinese microblogs [13]. The polarity of sentiment word in this lexicon is determined according to the sentiment polarity of the texts. Milagros et al. created an emoji lexicon in an unsupervised way [7]. The initial sentiment value of each emoji is set as the value assigned by the emoji creators. Then, the value of emoji is adjusted based on the texts containing the emojis. Although these methods effectively increase the number of sentiment words and provide some good ways to set the intensity values of sentiment words, they do not address the problem that the values of sentiment words vary from domains.

In recent years, some methods have been proposed to construct domain-specific sentiment lexicons. In order to adapt the sentiment classification for specific domains, Deng et al. extracted candidate words from the unannotated corpus [9]. Then, the sentiment orientation of the candidate word is determined by measuring the relations between the candidate word and the sentiment words in a seed lexicon. Nuno et al. presented an approach based on statistical measures to construct a stock market lexicon [14]. The value of sentiment word is calculated according to a lot of labelled stock market microblogs. Wei Li et al. proposed a method to detect new words in specific domains, which incorporates manually calibrated sentiment scores, semantic information, and statistical similarity information derived from word2vec [15]. Frank et al. adapted word polarities to target domain by training a sentiment classifier [8]. In the training process, the wrongly predicted sentences are used as the feedback to correct sentiment words, which effectively improves the accuracy of sentiment classification. However, most of these methods require external human annotations and complex linguistic knowledge. To automatically construct domain-specific sentiment lexicons, researchers proposed some methods. William et al. constructed a lexical graph and defined the graph edges and propagated sentiment labels over the graph by a random walk method [16]. Then, a bootstrap-sampling approach is utilized to obtain confidence regions over the sentiment scores. Sixing Wu et al. utilized syntactic relations and semantic similarities to extract opinion pairs. The opinion pairs are utilized in a one-class SVM classifier to automatically construct a sentiment lexicon for the specific domain [17]. However, the methods in [16, 17] do not consider the influence of n-grams. It has been proven by researchers that *n*-grams are very important to sentiment classification [18]. Authony et al. optimized n-gram-based text feature selection to improve the accuracy of sentiment analysis [19]. In [20], intensifiers and negations are extracted to construct n-gram features for cross-domain sentiment classification. The limitation of these two methods [19, 20] is that they depend on the existed unigrams sentiment lexicons and do not provide sentiment values for the n-gram features.

Based above analysis, the main problems of constructing sentiment lexicons can be summarized as follows. (i) For the traditional methods, they depend on linguistic knowledge and update sentiment words slowly. They are unable to add new sentiment words generated by the Internet to the lexicon on time. (ii) Some unigrams lexicons and n-grams lexicons only provide the polarities of sentiment words but do not provide the intensity values of sentiment words. (iii) The existing methods of constructing domain-specific lexicons are specially designed according to the background knowledge of domain, which are not universal.

Today, machine learning-based methods are also widely used in sentiment analysis. Veny et al. utilized three methods of decision tree, Naïve Bayes, and random forest to do the classification of social media Twitter [21]. Saerom Park et al. proposed a semisupervised distributed representation to describe the difference of document for sentiment analysis [22]. Some deep learning models such as CNN, LSTM, and GRU are applied to sentiment analysis [23, 24]. The attention mechanism is applied in the deep learning framework to improve the sentiment analysis performance [25]. Inspired by these ideal of self-learning, we propose a novel scheme to automatically construct sentiment lexicon based on corpuses. The proposed scheme can overcome the above problems that have existed in the previous studies, which provide a new way to construct sentiment lexicon with the help of network text resources.

3. The Proposed Scheme

3.1. Framework of Training-Optimization Scheme. In Web forums, there are plenty of user reviews, which usually contain the sentiment orientation of users. Plenty of user comments gathered in Web forums are valuable sources to extract sentiment words in specific domains. However, although the sentiment orientation in user comments is mainly determined by the intensity and polarity of sentiment words, it is still difficult to describe the relationship between them by an explicit formula. Inspired by the idea of machine learning, we propose a training-optimization framework to solve this problem, in which the construction of sentiment lexicons in a specific domain is transformed into a process of supervised learning. Thus, the presented framework provides a novel and effective way to construct sentiment lexicon and determine the value of sentiment word according to the sentiment orientations of texts.

The proposed framework is shown in Figure 1, which mainly includes four parts. The first part is to extract the sentiment words from the training dataset. In the second part, randomly initialize the values of sentiment words to construct the initial sentiment lexicon. In the third part, classify the texts according to the sentiment lexicon and judge whether the results of text classification meet the requirement. If it does not satisfy the requirement, the training-optimization-based algorithm is used to adjust the sentiment lexicon in part four. The detailed descriptions on these four parts are as follows:

(i) *Part (i).* Sentiment words extraction: The review texts in a specific domain are collected from Web forums and used as the corpuses to construct the sentiment lexicon. Since the comment texts such as tweets are very short, each word and n-gram feature in the texts have a certain impact on the result of sentiment classification. Thus, the words and the n-grams features appearing in the corpuses more than twice are selected as the candidate sentiment words.

(ii) *Part (ii).* Sentiment values initialization: The value of sentiment word is set according to a ten-point system. We initialize each sentiment word a random value in the interval $\{-10, -9, \ldots, -1, 0, 1, \ldots, 9, 10\}$. The positive and negative signs indicate the sentiment polarity, and the value of sentiment word represents the sentiment intensity.

(iii) *Part (iii).* Sentiment classification and evaluation: A selected text is classified according to its sentiment polarity. The sentiment polarity of a text is determined as follows. Get the value of each sentiment *t* words according to the sentiment lexicon and accumulate the values of all sentiment words. If the sum of sentiment values is larger than 0, the text is classified as positive. Otherwise, the text is classified as negative. Since all the testing texts have been annotated manually, the accuracy of text classification can be evaluated by comparing with the text labels. Finally, if the accuracy of text classification meets the requirement or the optimization algorithm is converged, output the current lexicon as the final sentiment lexicon. Otherwise, go to part (iv) and adjust the value of sentiment words in the lexicon.

(iv) *Part (iv).* Adjusting sentiment lexicon: Since there is no explicit rule to guide the adjustment of sentiment

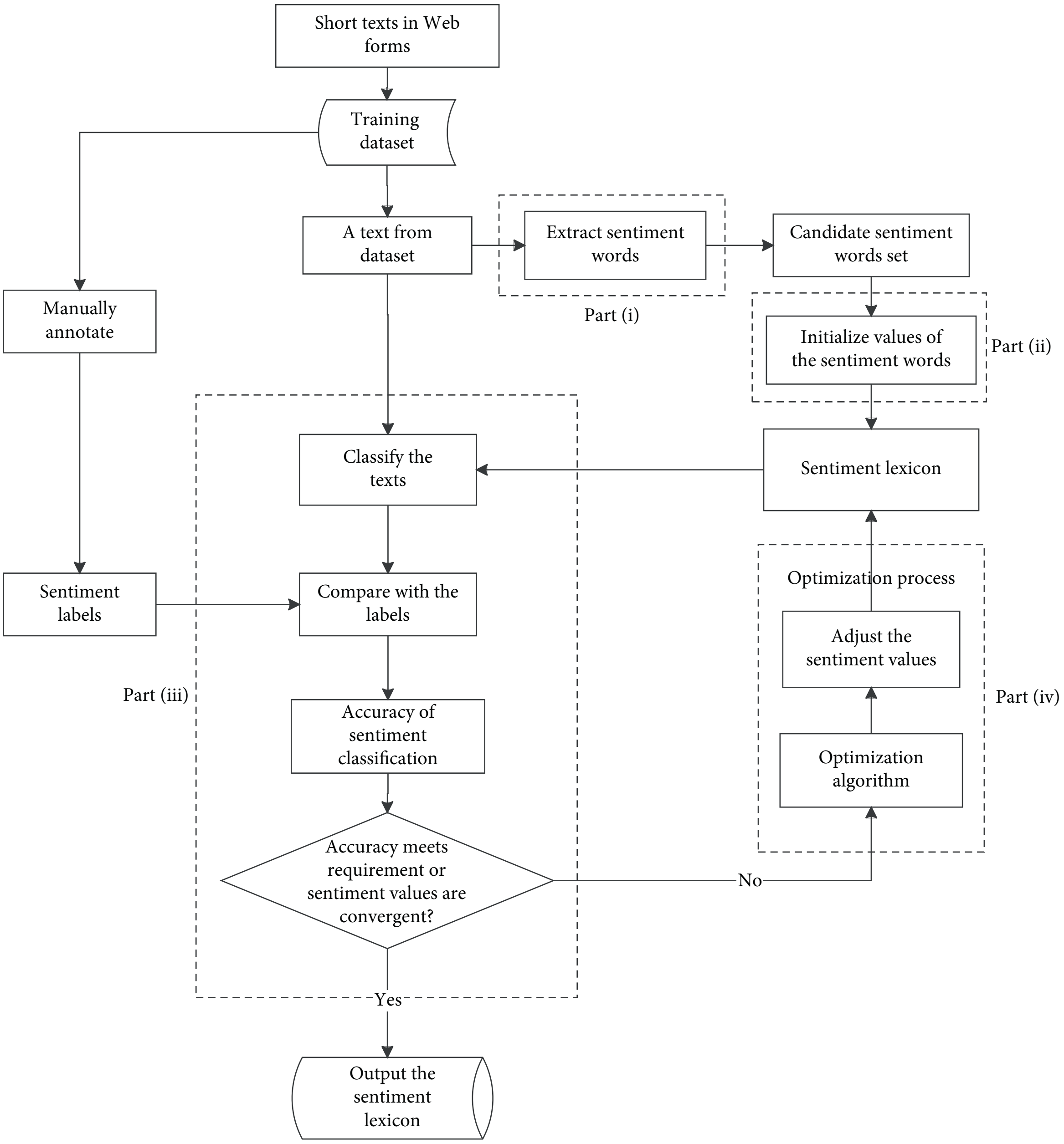

Figure 1: The proposed framework for constructing sentiment lexicon, which includes four main parts. Part (i): sentiment words extraction; Part (ii): sentiment values initialization; Part (iii): sentiment classification and evaluation; Part (iv): adjusting sentiment lexicon.

value, we randomly adjust the value of emotion words. In order to ensure the effectiveness of the adjustment, we further transform the process of adjusting the values of sentiment words into an optimization process. The accuracy of text classification is used as the optimization objective. The optimization of sentiment lexicon will be implemented by a genetic algorithm, which will be described in Section 3.2.

3.2. The Proposed Genetic Algorithms. In the framework, how to adjust the values of sentiment word in the lexicon is the core part of the framework. We adjust the intensity values of sentiment words based on the idea of the genetic algorithm. In this section, we firstly design the basic operations in the genetic algorithms, and then the framework is implemented by two algorithms, which are, respectively, called Algorithm 1 and Algorithm 2.

3.2.1. Basic Operations

(i) Population initialization

In order to optimize the lexicon by a genetic algorithm, a sentiment lexicon is regarded as an individual. Correspondingly, each sentiment word in a lexicon is regarded as a gene of the individual. Each sentiment word in a lexicon is initialized as a random integer between −10 and 10. By the same way, we initialize the whole population. Table 1 is an example of the initialized population.

(ii) Fitness value calculation

The fitness value is designed based on whether a sentiment lexicon can correctly classify the texts. If a text is correctly classified by the lexicon, the fitness value of the lexicon is increased by a reward value. Otherwise, the fitness value is subtracted by a penalty value. The reward/penalty function is described as follows:

$$R(L,D,T_i) = \begin{cases} 1, & \text{if correctly classified,} \\ -\dfrac{|P|}{\omega}, & \text{if incorrectly classified,} \end{cases} \tag{1}$$

where L is an individual, i.e., a lexicon, T_i is the i^{th} text in the training dataset D, ω is a penalty factor, and P is the sentiment value of a text that is computed by the sentiment lexicon according to the following equation:

$$P = \sum_{i=0}^{n} w_i, \tag{2}$$

where w_i is the value of the i^{th} sentiment word and n is the total number of sentiment words in the text.

According to the reward and penalty function, the fitness value of a lexicon is calculated as follows:

$$\text{fitness}(L,D) = \sum_{i=1}^{n} R(L,D,T_i). \tag{3}$$

(iii) Crossover

Let L_1 and L_2 be the two individuals who will perform the crossover operation. In the process of crossover, a position j is randomly selected. Then, the sentiment values of the j^{th} gene in L_1 is exchanged with that of the j^{th} gene in L_2. We repeat randomly selecting a new position and exchanging the corresponding sentiment values. The total number of exchanged genes is determined by the crossover ratio p_c. Finally, we can generate two new individuals by the crossover operation, which are denoted as C_1 and C_2, respectively. Here, for better understanding of our crossover scheme, an example is given in Figure 2. In Figure 2, the second and fifth genes of individual L_1 and L_2 are exchanged.

(iv) Mutation

In general, the sentiment polarity of the text is closely related to the polarity of sentiment words in the text. If the probability that a sentiment word exists in the positive texts is greater than its probability in the negative texts, the polarity of the sentiment word is also positive with a high probability, and vice versa. According to this idea, we design a new mutation strategy, which is different to the traditional mutation strategy that randomly changes the value of gene.

Table 1: An example of initialized population.

	Hello	Good	None	New	......
Individual-1	+2	+8	0	−8	......
Individual-2	−3	−1	+6	−7	......
Individual-3	−1	+7	−4	+9	......
......	......	......	......	......	......

Define r_w as the probability that sentiment word appears in the positive texts. We present the following function to guide the mutation of word w:

$$f(r_w) = 80(r_w - 0.5)^3, \quad r_w \varepsilon [0,1]. \tag{4}$$

Function f is plotted in Figure 3. We can see from Figure 3 that the value of function f is around 0 and changes slowly when $r_w \in (0.4, 0.6)$. It means that probability of word w appearing in a positive or negative text is almost the same. Thus, function f plays little influence on the sentiment value of w. When $r_w \in [0, 0.2]$, it means that word w appears in the negative texts with a probability more than 80%. Thus, the output of function f is a negative value and influences the final sentiment value greatly. Similarly, when $r_w \in [0.8, 1]$, the output of function f is a positive value and plays important impact on the final sentiment value. When $r_w \in (0.6, 0.8)$ or $r_w \in (0.2, 0.4)$, they are the transition intervals and function f plays an medium effect on the final sentiment value.

According to function f, we propose the mutation formula as follows:

$$v_w = R + f(r_w), \tag{5}$$

where v_w is the value of sentiment word w after mutation and R is a random number in interval $[-10, 10]$. In equation (5), R guarantees the diversity of population and function f makes the value of sentiment word changed along the polarity direction. It should be noted that if v_w is out of the interval [−10, 10], assign the corresponding boundary value to it.

Based on above analysis, our mutation scheme is proposed as follows. Randomly select one gene in the individual and change its sentiment value according to equation (5). Then, we repeat randomly selecting a new gene and changing its sentiment values. The total number of mutational genes is determined by the mutation ratio p_m.

3.2.2. Algorithm 1. According to the idea of evolution, Algorithm 1 is proposed to construct a sentiment lexicon. The details of Algorithm 1 are as follows:

Step 1. According to the target domain, short texts such as microblogs and tweets are collected from the

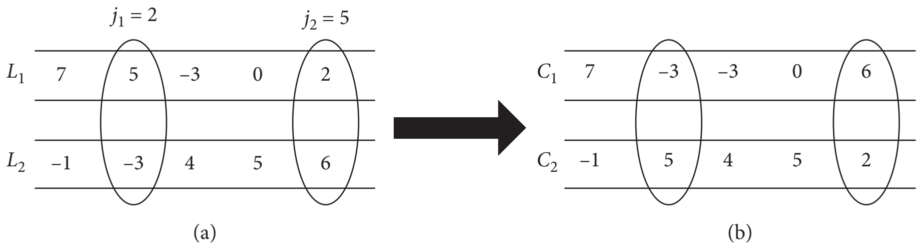

FIGURE 2: An example of crossover: the 2^{nd} and the 5^{th} genes in $(L)_1$ are exchanged with the genes of $(L)_2$ in the same position: (a) before crossover; (b) after crossover.

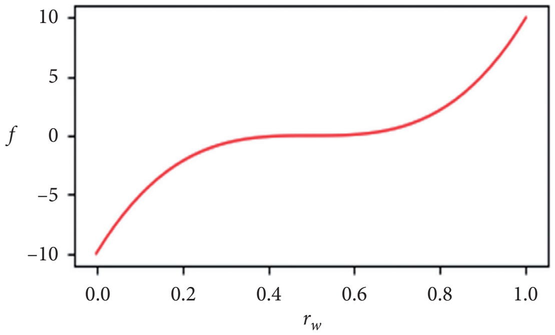

FIGURE 3: The curve of function (f).

Internet. Pick out the texts with emotions and manually annotate them to obtain a training dataset.

Step 2. For each text in the training dataset, the words and the n-grams features appearing more than twice are selected as the words of the sentiment lexicon.

Step 3. Initialize the population according to the scheme in Section 3.2.1.

Step 4. Calculate the fitness value of each individual according to the scheme in Section 3.2.1. Then, the roulette wheel policy is used to select two individuals from the population. Denote the two selected individuals as L_1 and L_2, respectively.

Step 5. Set the value of crossover ratio p_c and execute the crossover operation between L_1 and L_2 according to the crossover strategy in Section 3.2.1. Correspondingly, we get two individuals C_1 and C_1 after crossover operation.

Step 6. Set the value of mutation ratio p_m and perform the mutation operation on C_1 and C_2 according to the mutation strategy in Section 3.2.1. Correspondingly, we get two individuals M_1 and M_2 after mutation.

Step 7. According to the roulette wheel selection policy, select two worst individuals from the population, which are denoted as W_1 and W_2, respectively. Compare the fitness values of W_1, W_2, M_1, and M_2, select the two individuals with higher fitness value back to the population, and remove the other two individuals.

Repeat Step 4 to 7 until the population is converged. Finally, the best individual in the population is selected as the sentiment lexicon.

3.2.3. Algorithm 2. In Algorithm 2, we optimize the sentiment lexicon by the evolution of population from one generation to the next. Moreover, we introduce an elite strategy into Algorithm 2 to improve the speed of convergence. Algorithm 2 is described as follows:

Steps 1–3. These steps are the same as Algorithm 1.

Step 4. Set the value of the proportion of elites in the population as p_e. The first p_e% individuals with larger fitness value are regarded as the elites of the current generation. These elites are directly selected as the individuals of the next generation.

Steps 5–7. These steps are the same as Steps 4–6 of Algorithm 1.

Step 8. The two new generated individuals M_1 and M_2 are used as the individuals of the next generation. Repeat Steps 5–8 until all individuals in the next generation are produced.

Step 9. Repeat Steps 4–8 until the population is converged. Output the best individual as the sentiment lexicon.

4. The Advantage of Our Scheme

Our scheme provides a simple way to construct sentiment lexicon for a specific domain. In our scheme, the sentiment words are extracted from the corpuses collected from a specific domain. Our scheme can be regarded as a corpus-driven method. Getting enough high-quality corpora is very important to our scheme. Nowadays, microblogs, twitters, and Web forums provide plenty of resources to create corpuses. The sentiment lexicon constructed by our scheme can achieve high coverage if enough corpuses are obtained. The newly generated sentiment words are also easy to collect by our scheme if the corpora are updated from time to time. As we know, under current technical conditions, it is not difficult to automatically obtain and update the corpus from the Internet, which laid a good foundation for our scheme. Moreover, in the process of constructing the sentiment lexicon, our scheme is not related to the background knowledge of domain or the linguistic knowledge. Thus, our scheme has good universality and can be used to construct sentiment lexicon for any domains.

In our scheme, we have adopted the working method of machine learning. The intensity and polarity of sentiment words are determined by a training-optimization process. Our scheme provides an indirect and effective way to solve the problems of infer the value of sentiment words according to the polarities of short texts. Thus, the sentiment lexicon constructed by our scheme not only includes emotion word and their polarity but also includes their intensity values, which can better support sentiment analysis of texts.

In our scheme, the quality of sentiment lexicon heavily depends on the corpora collected from a certain domain. Using network searching technique or network spider technique, our scheme can continually collect corpora from the Internet and automatically obtain new sentiment words, including unigram and n-grams, and update their intensity values. Thus, the sentiment lexicon constructed by our method has the ability of continually updating sentiment words and improving its quality.

5. Experiments

5.1. Datasets. Our experiments are performed on five public datasets of different domains, which are OMD, SOMD, HCR, SemEval2013, and STS-Test. Since the polarities of corpuses in the datasets have already been annotated by the publishers, they can be used to test our scheme directly. The details on these datasets are described as follows:

(i) OMD1 is the short text dataset about Obama-McCain debate, which was captured from the TV debate between Obama and McCain in 2008 US elections. It has 710 positive texts and 1,196 negative texts, which can be regarded as a dataset in political domain.

(ii) SOMD2 is the Strict Obama-McCain Dataset. SOMD is another version of the OMD. It contains 569 positive and 347 negative tweets. It is also used as a dataset in political domain.

(iii) HCR3 is a comment dataset on Healthcare Reform, which was constructed in 2010. The HCR dataset contains 1,286 tweets, of which 369 are positive text and 917 are negative text. It is used as a dataset in healthcare reform domain.

(iv) SemEval20134 is a short text dataset on hot issues or products, which is dedicated to Twitter sentiment analysis. It contains 3,640 positive tweets and 1,458 negative tweets. It can be regarded as a general dataset related to people's lives.

(v) STS-Test5 is the Stanford Twitter Dataset, which includes the comments on hot issues. The texts in the STS-Test dataset are manually annotated; among them, 177 texts are negative and 182 texts are positive.

5.2. Evaluation Metrics. The following performance metrics are used in the experiments of this paper, including Accuracy and F1-measure. Accuracy is a common and intuitive evaluation index shown in equation (6), which indicates the proportion of the correctly classified number of texts to the total number of the texts:

$$\text{accuracy} = \frac{a+d}{a+b+c+d}, \tag{6}$$

where a is the number of positive texts classified correctly, d is the number of negative texts classified correctly, b is the number of positive texts classified incorrectly, and c is the number of negative texts classified incorrectly.

However, in the cases when the dataset has an unbalanced distribution, the Accuracy cannot reflect the classifier's performance. In this work, F1-measure is adopted as another indicator, which is the weighted average of Precision and Recall. F1-measure is defined as follows:

$$F1 - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \tag{7}$$

wherePrecision = $(a/(a+d))$ and Recall = $(a/(a+b))$.

5.3. The Parameters of Algorithms. According to the general experiences, we set the population size as 2,000 and the penalty factor ω as 60. Since the crossover ratio p_c and the mutation ratio p_m have important impact on the performance of the genetic algorithm, we determine them by an experiment way. The HCR dataset is selected as the test datasets because it has a lot of texts. F1-measure is used as the indicator for choosing p_c and p_m. The test results are shown in Table 2. According to Table 2, we set $p_c = 0.7$ and $p_m = 0.1$. For Algorithm 2, we need to set the value of p_e. According to our experience, we set p_e for all datasets except for the STS-test. The main reason is that the labelled texts in the STS-test are very small. For this dataset, we set p_e as 0.025.

5.4. Results and Analysis

5.4.1. The Result of Sentiment Lexicon. According to our scheme, the sentiment lexicon can be generated based on the training set, which is constructed by randomly selecting 80% corpora from the dataset. Here, we take the HCR dataset as an example and construct the sentiment lexicon for Healthcare Reform. In this sentiment lexicon, there are 1,787 unigrams, 1,772 bigrams, and 843 trigrams. Some unigrams, bigrams, and trigrams in the sentiment lexicon for Healthcare Reform are listed in Table 3. It can be seen from Table 3 that the word "obstruction" is used as a negative word in our sentiment lexicon. In general, the word "obstruction" does not be regarded as a sentiment word. In our sentiment lexicon, quite a few words are similar to the word "obstruction." Thus, it also confirms that our scheme does provide an effective way to obtain sentiment words in the specific domain.

In the sentiment lexicon constructed based on the HCR dataset, there are 1,687 positive words and 2,500 negative words, which provide a good coverage of sentiment words. However, in the sentiment lexicon, there are 214 words that their sentiment values are 0. It means that these words are regarded as neutral words in our scheme, which can be removed from the lexicon. In our scheme, to improve the coverage of lexicon, we add candidate sentiment words

Table 2: The F1-measure values with different p_c and p_e.

Parameters		F1-measure (%)
p_c	p_e	
0.70	0.05	80.49
0.70	0.10	84.71
0.70	0.15	80.00
0.80	0.05	76.92
0.80	0.10	82.76
0.80	0.15	71.79
0.90	0.05	69.88
0.90	0.10	80.49
0.90	0.15	72.29
1.00	0.05	65.82
1.00	0.10	77.89
1.00	0.15	70.89

according to the frequency of word appearing in the corpora. It is possible that some words that are not emotional words have also been selected into the sentiment lexicon. Finally, the intensity values of these words are set close to 0 by the optimization process. Thus, although some neutral words are selected into the sentiment lexicon at the beginning, these words still can be excluded by setting their intensity values, which guarantees the high quality of our sentiment lexicon. Nevertheless, it is worth further studying that how to select sentiment words from corpora.

The other sentiment lexicons constructed from other datasets have the similar situation to the sentiment lexicon based on the HCR dataset. Therefore, the experiment results confirm that our scheme can automatically construct sentiment words (unigrams and n-grams) for specific domains and reasonably set their intensity values.

5.4.2. Performance Analysis and Comparison. We firstly test the performances of Algorithm 1 and Algorithm 2 in the five datasets. Then, some different conditions are also considered in Algorithm 1 and Algorithm 2. Algorithm-SW means running the algorithms without considering stop words. Algorithm+1, 2, 3-grams gives the results by combining the algorithms and n-gram features. Algorithm + Lex computes the results by combining Bing Liu's lexicon [26]. All the test results for each dataset are listed in Table 4.

In [16], the authors present a method to construct domain-specific sentiment lexicons based on unlabelled corpora. We implement this method and test its performances in Accuracy and F1-measure. The results are also listed in Table 3 for comparison. Moreover, some similar schemes [27–29] are selected and tested in the same datasets. To save the article length, we only list the best result from literatures [27–29], which is denoted as BRFL in Table 4.

According to Table 4, we can get that the best Accuracy and F1-measure of Algorithm 1 and Algorithm 2 are always better than the results in Ref. [16]. It means that using labelled corpora is beneficial to improve the quality of sentiment lexicon. In the four datasets, i.e., STS-Test, HCR, OMD, and SOMD, the best results of Algorithm 1 and Algorithm 2 are better than the best results of BRFL. Only in the dataset SemEval2013, the BRFL has a little better performance than Algorithm 1 and Algorithm 2.

Meanwhile, Algorithm 1 has a little better performance than Algorithm 2 in most cases. In our experiments, the unigrams and n-gram features are also considered. Here, the n-gram features include bigrams and trigrams. According to Table 4, we can get that the sentiment lexicon has better performance when combining the n-gram features. The test results also confirm that the n-gram features play an important role in sentiment analysis.

According to Table 4, we can get that combining Bing Liu's lexicon with our scheme can achieve better performance in most cases. The main reason is that adding Bing Liu's lexicon can increase the coverage of sentiment words. However, in some cases, combining Bing Liu's lexicon with our scheme cannot improve the performance and even get worse results. The main reason is that for some sentiment words, their sentiment values are conflict between our lexicon and Bing Liu's lexicon. Bing Liu's lexicon is a general sentiment lexicon. The sentiment word in Bing Liu's lexicon cannot reflect the traits of specific domain. Therefore, using the sentiment words in Bing Liu's lexicon to replace the corresponding sentiment words in our lexicon will lower the performance of sentiment analysis. Therefore, how to combine different sentiment lexicons is a problem worth studying in the future.

In addition, Algorithm-SW shows the results of using the algorithm without stop words. We can see that the results get worse in this case. It also confirms the conclusion in [30] that omitting the stop words has the negative influence on sentiment analysis.

5.4.3. Adaptability Analysis. To test the adaptability of our scheme, we construct a Chinese dataset on stock market by collecting more than 4,000 short comments from some financial web forums, such as "https://www.weibo.com/." Then, we manually annotated the texts in the dataset. The Chinese dataset on stock market is used to test whether our scheme is suitable to construct Chinese sentiment lexicon. We generate a Chinese sentiment lexicon based on this dataset and test the results on the Accuracy and F1-measure, which are shown in Table 5.

Moreover, some famous Chinese sentiment lexicons, such as DUTIR[6] and TSING[7], are also used to test the results in the same dataset and compare with our scheme. In [6], the sentiment lexicon is expanded based on a neural learning method combing dictionary lookup and polarity association. This method is also used to compared with our scheme. All the test results are also listed in Table 5 for better comparison.

According to Table 5, we can see that our scheme has better Accuracy and F1-measure than the other scheme. The test results show that our scheme is suitable for constructing Chinese sentiment lexicon, which also confirms that our scheme has good adaptability.

5.4.4. Efficiency Analysis. For Algorithm 1, only two individuals are updated in each iteration. It means that the speed of convergence is not very fast. The largest fitness value in the

Table 3: Some example words in the sentiment lexicon, which is generated based on the HCR dataset.

Sentiment words	Serious	Sick	Awesome	Cancer	Death	Drug	Hope	Moved	Obstruction	Urgent
Value	−9	−8	9	−7	−9	−5	3	9	−2	−6
Sentiment words	Better than		Get health		Give up		Raise taxes		Walk away	
Value	3		5		−4		−10		−4	
Sentiment words	Abuse of power		Best health system		Make no mistake		Get it done		Lack of health	
Value	−10		8		3		7		−5	

Table 4: Comparison of accuracy and F1-measure (the bold values represent the best values).

Methods	Accuracy (%)		F1-measure	
	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
STS-test dataset				
Algorithm	77.78	75.00	76.47	74.26
Algorithm-SW	77.78	75.00	78.95	79.07
Algorithm + 1,2,3-grams	**88.89**	86.11	**87.50**	85.71
Algorithm + Lex	83.33	77.78	82.35	77.78
Algorithm + 1,2,3-grams + Lex	86.11	83.33	83.87	82.35
BRFL	88.40		83.90	
Ref. [16]	80.22		80.25	
HCR dataset				
Algorithm	88.37	83.72	**82.76**	76.40
Algorithm –SW	83.72	79.84	70.42	66.67
Algorithm + 1,2,3-grams	86.82	86.05	80.46	80.43
Algorithm + Lex	**89.15**	84.50	82.50	77.27
Algorithm +1,2,3-grams + Lex	86.82	85.27	80.00	79.12
BRFL	80.04		68.20	
Ref. [16]	69.91		63.69	
OMD dataset				
Algorithm	82.20	79.58	80.00	75.78
Algorithm –SW	78.53	76.96	72.11	75.00
GA+1,2,3-grams	79.59	80.63	78.69	78.36
Algorithm + Lex	82.72	83.25	80.47	80.25
Algorithm +1,2,3-grams + Lex	**83.25**	82.72	**82.80**	80.92
BRFL	82.90		77.85	
Ref. [16]	70.25		67.20	
SemEval2013 dataset				
Algorithm	83.53	83.13	89.53	89.28
Algorithm –SW	86.67	72.94	91.56	81.79
Algorithm + 1,2,3-grams	84.31	79.61	89.42	86.56
Algorithm + Lex	83.34	83.73	89.34	89.66
GA+1,2,3-grams + Lex	84.12	80.00	89.24	86.82
BRFL	**93.70**		**93.70**	
Ref. [16]	77.13		70.35	
SOMD dataset				
Algorithm	86.96	84.78	85.01	81.58
Algorithm –SW	79.35	78.26	75.32	72.97
Algorithm +1,2,3-grams	91.30	81.52	90.48	79.01
Algorithm + Lex	89.13	89.13	87.18	87.18
Algorithm +1,2,3-grams + Lex	**91.30**	84.78	**90.48**	82.93
BRFL	80.35		79.25	
Ref. [16]	71.40		68.73	

population is used as a criterion to evaluate the convergence. We use the HCR dataset as an example to evaluate the efficiency of the proposed algorithms. Figure 4 shows the convergence process of fitness value in Algorithm 1. It can be seen that Algorithm 1 is converged after 128,000 iterations. Since each iteration updates two individuals, the converging time can be regarded as the time for Algorithm 1 to update 256,000 individuals.

In Algorithm 2, the evolution of the population is carried out from generation to generation. The convergence process of Algorithm 2 is illustrated in Figure 5. Algorithm 2 is converged at the 140th generation. To compare the efficiency

Table 5: The test results in a Chinese dataset on stock market.

Sentiment lexicon	Accuracy (%)	F1-measure (%)
Our scheme	**85.70**	**78.83**
DUTIR	61.33	54.74
TSING	62.90	61.80
Ref. [6]	67.13	57.39

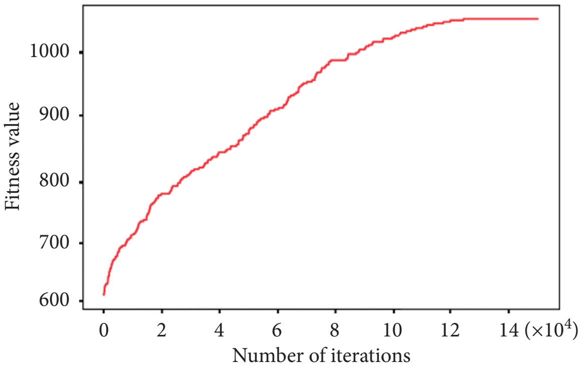

Figure 4: The convergence process of Algorithm 1. Algorithm 1 is converged after 128,000 iterations and totally updates 256,000 individuals.

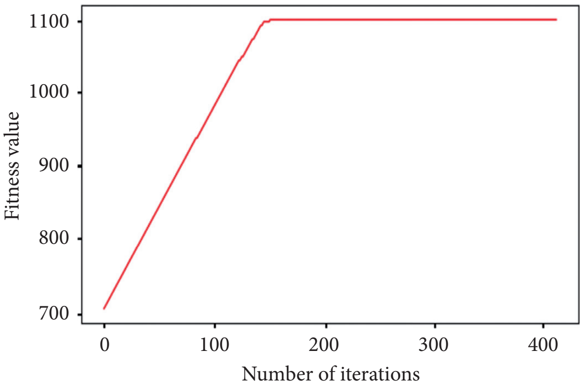

Figure 5: The convergence process of Algorithm 2. Algorithm 2 is converged at 140 generations and totally updates 252,000 individuals.

with Algorithm 1, we use the number of updating individuals as the criterion. For Algorithm 1, the total numbers of updating individuals are 256,000, respectively. In Algorithm 2, the population size is 2,000 and the elite rate is 0.1. So, the number of updating individuals in each generation is 1,800. In the 140 generations, the total number of updating individuals is 252,000. Comparing the number of updating individuals, Algorithm 2 has the higher efficiency than Algorithm 1.

6. Conclusion

Sentiment lexicon is an important component for text sentiment analysis. In this paper, we proposed a framework based on training and optimization to construct sentiment lexicon for a specific domain. According to this framework, our presented method provides a way to automatically generate sentiment words and their intensity values for a specific domain by using the labelled corpora. Especially, we design two genetic algorithms, which can find suitable value for sentiment words by optimizing the accuracy of sentiment classification. Since the fast development of Internet, it is not difficult to find plenty of texts in a specific domain, which provides enough corpuses for our method. Thus, our method can be easily implemented in practice and effectively construct the specific sentiment lexicon for a given domain. Our method does not depend on the domain knowledge and has good adaptability and universality. The test results in five datasets from different domains also confirm that the sentiment lexicons generated by our method has good performance and can effectively support the sentiment analysis of short texts.

The main limitation in our method is that we simply choose the words that appear twice in a short text as sentiment words. Quite a few words among them are not really sentiment words. Although most of the sentiment values of these word are close to 0 after optimization and their impacts are very weak, these words still have negative effect on the results and efficiency of sentiment classification. Thus, it is necessary to design a method on how to filter the unnecessary words.

In the future, we think it is still worth studying that how to merge multiple sentiment lexicons together to improve the coverage of sentiment words and avoid the conflicts between sentiment words. In addition, how to collect the high-quality corpus for a specific domain from the Internet is still worth studying.

Data Availability

The data used to support the findings of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61876200), the MOELayout Foundation of Humanities and Social Sciences (no. 20YJAZH102), and the Chongqing Social Science Planning Project (no. K2015-59).

References

[1] L. Wei and L. Shi, “Investor sentiment in an artificial limit order market,” *Complexity*, vol. 2020, Article ID 8581793, 10 pages, 2020.

[2] A. Dey, M. Jenamani, and J. J. Thakkar, “Senti-N-Gram: an n -gram lexicon for sentiment analysis,” *Expert Systems with Applications*, vol. 103, pp. 92–105, 2018.

[3] P. Agathangelou, I. Katakis, I. Koutoulakis, F. Kokkoras, and D. Gunopulos, “Learning patterns for discovering domain-

oriented opinion words," *Knowledge and Information Systems*, vol. 55, no. 1, pp. 45–77, 2018.

[4] J. R. Ragini, P. M. R. Anand, and V. Bhaskar, "Big data analytics for disaster response and recovery through sentiment analysis," *International Journal of Information Management*, vol. 42, pp. 13–24, 2018.

[5] F. Smarandache, M. Colhon, Ş. Vlăduţescu, and X. Negrea, "Word-level neutrosophic sentiment similarity," *Applied Soft Computing*, vol. 80, pp. 167–176, 2019.

[6] W. Y. Sheng, Y. Zhang, and B. Liu, "Sentiment lexicon expansion based on neural PU learning, double dictionary lookup, and polarity association," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 553–563, Copenhagen, Denmark, 2017.

[7] M. Fernández-Gavilanes, J. Juncal-Martínez, S. García-Méndez, E. Costa-Montenegro, and F. J. González-Castaño, "Creating emoji lexica from unsupervised sentiment analysis of their descriptions," *Expert Systems with Applications*, vol. 103, pp. 74–91, 2018.

[8] F. Z. Xing, F. Pallucchini, and E. Cambria, "Cognitive-inspired domain adaptation of sentiment lexicons," *Information Processing & Management*, vol. 56, no. 3, pp. 554–564, 2019.

[9] S. Deng, A. P. Sinha, and H. Zhao, "Adapting sentiment lexicons to domain-specific social media texts," *Decision Support Systems*, vol. 94, pp. 65–76, 2017.

[10] H. J. Heng, C. Y. Hua, and W. C. Yao, "Constructing sentiment dictionary with deep learning: case study of financial data," *Data Analysis and Knowledge Discovery*, vol. 2, pp. 95–102, 2018.

[11] J. C. Qing, G. Y. Bo, and L. Yao, "Constructing a domain sentiment lexicon based on Chinese social media text," *Data Analysis and Knowledge Discovery*, vol. 3, pp. 98–107, 2019.

[12] M. Ghiassi and S. Lee, "A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach," *Expert Systems with Applications*, vol. 106, pp. 197–216, 2018.

[13] S. Zhang, Z. Wei, Y. Wang, and T. Liao, "Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary," *Future Generation Computer Systems*, vol. 81, pp. 395–403, 2018.

[14] N. Oliveira, P. Cortez, and N. Areal, "Stock market sentiment lexicon acquisition using microblogging data and statistical measures," *Decision Support Systems*, vol. 85, pp. 62–73, 2016.

[15] W. Li, K. Guo, Y. Shi, L. Zhu, and Y. Zheng, "DWWP: domain-specific new words detection and word propagation system for sentiment analysis in the tourism domain," *Knowledge-Based Systems*, vol. 146, pp. 203–214, 2018.

[16] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 595–605, Austin, TX, USA, 2016.

[17] S. Wu, F. Wu, Y. Chang, C. Wu, and Y. Huang, "Automatic construction of target-specific sentiment lexicon," *Expert Systems with Applications*, vol. 116, pp. 285–298, 2019.

[18] D. Gamal, M. Alfonse, E.-S. M. El-Horbaty, and A.-B. M. Salem, "Implementation of machine learning algorithms in Arabic sentiment analysis using N-gram features," *Procedia Computer Science*, vol. 154, pp. 332–340, 2019.

[19] M. A. Cabanlit and K. J. Espinosa, "Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons," in *Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications*, pp. 94–97, Chania, Greece, 2014.

[20] A. Dey, M. Jenamani, and J. J. Thakkar, "Sentiment wEight of N-grams in dataset (SEND): a feature-set for cross-domain sentiment classification," in *Proceedings of the 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pp. 1–6, Bangalore, India, 2017.

[21] V. A. Fitri, R. Andreswari, and M. A. Hasibuan, "Sentiment analysis of social media twitter with case of anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm," *Procedia Computer Science*, vol. 161, pp. 765–772, 2019.

[22] S. Park, J. Lee, and K. Kim, "Semi-supervised distributed representations of documents for sentiment analysis," *Neural Networks*, vol. 119, pp. 139–150, 2019.

[23] C. A. Martín, J. M. Torres, R. M. Aguilar, and S. Diaz, "Using deep learning to predict sentiments: case study in tourism," *Complexity*, vol. 2018, Article ID 7408431, 9 pages, 2018.

[24] Z. Zhang, L. Wang, Y. Zou, and C. Gan, "The optimally designed dynamic memory networks for targeted sentiment classification," *Neurocomputing*, vol. 309, pp. 36–45, 2018.

[25] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharrya, "An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems Special Issues*, vol. 115, pp. 279–294, 2020.

[26] H. M. Qing and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 168–177, Washington, DC, 2004.

[27] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Systems*, vol. 66, pp. 170–179, 2014.

[28] X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 537–546, Rome, Italy, 2013.

[29] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for twitter sentiment analysis," in *Proceedings of the 2nd Workshop on Making Sense of Microposts*, pp. 2–9, Lyon, France, 2012.

[30] H. Saif, M. Fernandez, Y. He, and H. Alani, "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," in *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*, pp. 810–817, Reykjavik, Iceland, 2014.