# Sentiment Analysis with Automatically Constructed Lexicon and Three-Way Decision

Zhe Zhou, Weibin Zhao, and Lin Shang

State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China
zhouzhenjucs@gmail.com zhaowb@njupt.edu.cn shanglin@nju.edu.cn

**Abstract.** An unsupervised sentiment analysis method is presented to classify user comments on laptops into positive ones and negative ones. The method automatically extracts informative features in testing dataset and labels the sentiment polarity of each feature to make a domain-specific lexicon. The classification accuracy of this lexicon will be compared to that with an existing general sentiment lexicon. Besides, the concept of three-way decision will be applied in the classifier as well, which combines lexicon-based methods and supervised learning methods together. Results indicate that the overall performance can reach considerable improvements with three-way decision.

**Keywords:** sentiment analysis, opinion mining, sentiment lexicon, three-way decision.

## 1 Introduction

Sentiment analysis, also known as opinion mining, refers to detecting the sentiment polarity or sentiment strength of a given piece of text[1]. Nowadays people can freely post their opinions and comments on the Internet and receive others' views at the same time[2]. Therefore, sentiment analysis becomes popular and urgent for some particular groups of Internet users. For example, commodity producers may collect reviews written by consumers and try to obtain the overall sentiment tendency in order to know whether their products are popular or not and what advantages and disadvantages they have[3, 4]. On the other hand, consumers can as well search their peers' opinions and reviews in order to know whether the product they want is worth buying[5]. In such cases, techniques similar to traditional topic-based classification algorithms can be used to automatically assign sentiment labels to product reviews. However, such methods may run into difficulty due to the speciality of product review sentiment analysis. Firstly, a major difference between traditional topic-based classification and sentiment analysis is that sentiment is often expressed in a subtle way, which will pose challenges in the classification work[6]. Besides, reviews on products or services often pay attention to detailed features or aspects[3, 4, 7], so feature-level analysis must be taken into consideration in the analysis process.

D. Miao et al. (Eds.): RSKT 2014, LNAI 8818, pp. 777–788, 2014.
DOI: 10.1007/978-3-319-11740-9_71 

In this article, we will design an unsupervised sentiment analysis algorithm to deal with these problems. The algorithm can help to understand the sentiment of product reviews better by utilizing the feature-level information. Also, we will apply a three-way-decision-like concept into the scheme in order to boost its performance and get a finer-grained system. The proposed scheme will be applied on a dataset made up by laptop reviews to test its efficiency. The remainder of this article is organized as follows. Sect. 2 will list some related work about sentiment analysis. Sect. 3 will explain the proposed method in detail. Sect. 4 will list and analyse the experiment results. Sect. 5 will conclude the whole work above and look into the future work.

## 2 Related Work

There have been many contributions studying text sentiment analysis during the past decade. Pang et al.[6] collected over a thousand movie reviews for binary sentiment classification and compared performances of three different machine learning algorithms including Naive Bayesian, Max Entropy and the Support Vector Machine. These movie reviews have been one of the most well-known benchmark datasets for sentiment analysis since this contribution. Besides, Pang and Lee[8] also focused on extracting only the subjective sentences for feature selection aiming to improve the performance of sentiment analysis. The process of subjectivity summarization was based on the minimum-cut algorithm and proved to be beneficial to the classifier's performance. Recent works include Hu et al.[9], who provided a supervised sentiment analysis approach in microblogging by utilizing users' social relation information to tackle noises. Besides those works based on supervised learning, there have been enormous unsupervised-learning-based contributions as well. Turney[10] calculates the semantic orientations of a large number of 2-grams with the help of search engines and use them to classify the given text. Li and Liu[11] introduced an clustering-based approach in sentiment analysis and obtained satisfying results by applying TF-IDF weighting, voting mechanism and important term scores. Taboada et al.[12] used different general sentiment lexicons in their lexicon-based sentiment analysis approaches and made a comparison between those lexicons. Hogenboom et al.[13] manually created a lexicon consisting emoticons to aid the traditional classification work. With the problem that current expressions on social media are usually unstructured and informal, Hu et al.[14] incorporated emotional signals into the unsupervised sentiment analysis framework and experimental results proved the effectiveness of emotional signals.

Current sentiment analysis works mostly focus on adjective features as adjectives are believed to be more informative in indicating sentiment orientations. However, Zhang and Liu[15] pointed out that in sentiment analysis on product reviews, it is often necessary to apply domain-specific noun features into the feature space. There have been many contributions concerning this aspect. For example, Yin and Peng[3] build semantic orientations between product features and sentiment words in reviews written in Chinese. Hu and Liu[7] addressed sentiment analysis on customer reviews by extracting frequent product features and using them to summarize the sentiment of the whole review. Similarly, Zhang et al.[4] addressed feature-level sentiment analysis by combining association rules and point-wise mutual information together to extract product features and identify their sentiment orientations. Riloff et al.[16] proposed a method to extract subjective nouns, which again proved that noun features can help sentiment analysis quite well, especially in product reviews.

## 3 Proposed Method: Feature Lexicon Construction and Three-Way Decision

In this section we will explain the proposed scheme to tackle the potential difficulty in traditional sentiment analysis methods. We extract informative patterns from the dataset and calculate sentiment scores of those patterns with the help of a general lexicon. The newly formed product feature lexicon and the general lexicon will be separately used in a lexicon-based sentiment analysis algorithm and return two different results. Finally, we will introduce the concept of three-way decision and use a similar method to reach a better classification accuracy.

### 3.1 Data Preprocessing

Our work aims to run a sentiment analysis on comments written in Chinese, so the most crucial parts in preprocessing step will be word segmentation and part-of-speech tagging. Word segmentation refers to cutting every sentence into its component words and part-of-speech tagging means using natural language processing techniques to obtain the part of speech of each word.

### 3.2 Lexicon-Based Sentiment Analysis

A sentiment lexicon, which can be considered as a special kind of dictionary, is a data structure containing different words and their sentiment orientations. Typically, the sentiment orientation is represented by a numerical value. A value greater than zero refers to a positive orientation and a value smaller than zero indicates a negative one. At the moment there are plenty of public sentiment lexicons on the Internet. Those general lexicons are integrated by other people's manual work and can be applied into sentiment analysis works of any domain.

In our proposed scheme, a general lexicon of HowNet[17], which contains about 9000 Chinese words and their sentiment polarities as positive or negative, will be utilized as the lexicon to classify a piece of text into two sentiment categories. The pseudo-code of classification algorithm is shown below.

As [1, 6] have mentioned, sometimes lexicon-based sentiment analysis may encounter a large amount of ties where the sentiment score will be 0. It's usually because the times of occurrences of positive and negative words are equal (usually both are 0 when the text is not long enough). According to [12], people tend to favour positive expressions when they make comments and negative languages are often expressed in a obscure and indirect way which is actually hard to detect.

```
Input: text, lexicon
Output: sentiment label for text

score = 0
negation = 1
for i = 0 : text.length
    if text[i] is "不"        //"not"
        negation *= -1
    if text[i] is punctuation
        negation = 1
    if text[i] is in lexicon
        polarity = lexicon.get(text[i]) == positive ? 1 : -1
        score += negation * polarity
if score > 0
    classify text as positive
else
    classify text as negative

return score
```

**Fig. 1.** Lexicon-based algorithm for sentiment analysis

This phenomenon makes negative oriented comments much easier to be ignored and incorrectly classified. As a solution to the "positive bias", [12] gives negative words additional weights to reach a balance. For the same reason, out tie-breaker will always label tie comments as negative ones.

### 3.3 Automatically Constructed Feature Lexicon

General sentiment lexicons can make contributions to many sentiment analysis works, but our work focuses on the categorization of comments on laptops, which will be more challenging due to their unique traits. For example, the comment "the cost-performance is low" expresses a negative sentiment, although neither "cost-performance" nor "low" can be found in general sentiment lexicons. From the example, it's easy to see that when customers comment on electronic devices, they tend to express their opinions on product features rather than directly use sentiment-carrying words, especially when they want to show their negative views. So it will be inefficient to run analysis only with general lexicons. In order to solve this problem, we design an algorithm to automatically extract the product-feature-related phrases out of the whole corpus. These phrases together form a laptop feature lexicon.

The process of constructing a laptop feature lexicon is based upon an assumption that is consistent with people's general intuitions: the sentiment expressed by a word is to some extent correlated with the sentiment of words co-occurring with it. In [10], similar assumptions were used to label target phrases by calculating mutual information between the phrase and seed words. The detailed

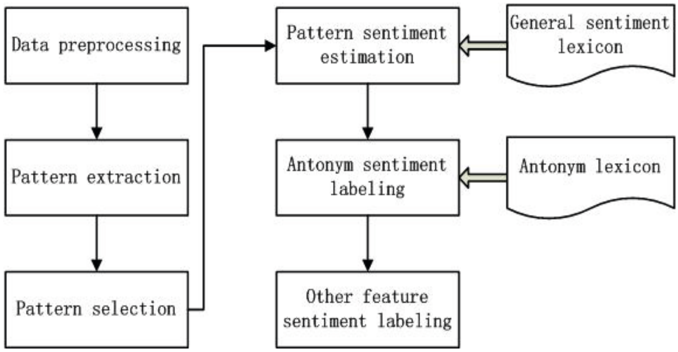


**Fig. 2.** Steps of building a laptop feature lexicon

scheme is shown below. The whole process consists of two main steps: pattern extraction and sentiment polarity assignment.

In the patterns extraction step, the algorithm detects all the "n-adj" patterns (a "n-adj" pattern is a noun and an adjective with no punctuations between them) from the corpus and stores them in pre-defined data structure. Afterwards, if a pattern's noun part occurs at least 100 times in the corpus and the occurrence of the pattern itself is greater than 10, then it will be put into the laptop feature lexicon. Otherwise the pattern will be removed.

**Table 1.** Patterns after selection

| pattern | noun freq | n.+adj. freq |
|---|---|---|
| cost-performance+high | 685 | 233 |
| laptop+hot | 181 | 67 |
| camera+clear | 200 | 14 |
| price+high | 632 | 33 |
| memory+small | 491 | 67 |
| speed+high | 873 | 168 |
| speed+low | 873 | 321 |
| ... | ... | ... |

Table 1 shows part of the terms in the final laptop feature lexicon. Of course in our experiment all those patterns are written in Chinese. Then our algorithm will automatically assign a sentiment label to every pattern in the laptop feature lexicon. When a pattern is found in the pattern extraction process, our algorithm will extract the n words before and following its noun part and store them in another data structure (in this work we let n=5). We call those words "neighbourhood sentences". For every pattern, we extract all the neighbourhood sentences near its occurrences and use the same method as Fig. 2 to compute their sentiment scores. After that we will be able to get the average sentiment

**Table 2.** Patterns and their sentiment polarities

| pattern | sentiment score | sentiment polarity |
|---|---|---|
| cost-performance+high | 1.70 | 1 |
| laptop+hot | -0.88 | -1 |
| camera+clear | 1.14 | 1 |
| price+high | -0.52 | -1 |
| memory+small | -0.91 | -1 |
| speed+high | 1.72 | 1 |
| speed+low | 0.07 | -1 |
| ... | ... | ... |

score of all the neighbourhood sentences, we see that as an estimation of the sentiment score of the pattern itself. Table 2 shows part of the selected patterns, their corresponding average sentiment scores are shown in the middle column.

Lastly we transform every sentiment score to +1/-1. This is done with the help of an antonym lexicon which contains a large number of antonym pairs (see Table 3). We traverse through all the frequent patterns to find the pattern pairs whose noun parts are the same but adjective parts are antonyms. When such pairs are encountered, the pattern with larger sentiment score is given the sentiment polarity of +1 and the other -1. The polarity of remaining patterns will be decided by the sign of their sentiment scores.

**Table 3.** An antonym lexicon

| word1 | word2 |
|---|---|
| front | back |
| forward | backward |
| high | low |
| public | private |
| cold | hot |
| die | alive |
| ... | ... |

After all the patterns are given a sentiment polarity label (see the last column in Table 2), the laptop feature lexicon is finally constructed. Now we can run the algorithm in Fig. 2 again with the laptop feature lexicon in place of the general HowNet lexicon and expect the new algorithm to reach satisfying results.

### 3.4 Three-Way Decision

Most sentiment analysis problems are treated as binary classification tasks[3, 4, 6–8, 10–12, 14, 15], in which a piece of text is either labelled as "positive" or "negative". Such idea is simple and direct, but sometimes can not reflect the nature of the real world. Take a real world problem into consideration, if a classifier is trained to predict whether an incoming Email is a spam one or not, it will encounter some Emails which are hard to classify into either of the two categories. For example, assume the classifier is trained by logistic regression, and there is an Email whose probability of being negative (spam mail) is estimated as 0.55. Of course the Email will be classified as a spam, but the classifier will take great risks doing so because the probability of the mail being legitimate is up to 0.45 as well. Therefore, it is necessary to introduce a third way of decision into the classification task, where the classifier can refuse to classify emails if it's not confident enough of the emails' categorization. A rejected email will be labelled as "suspicious" and presented to the user, who will make his own judgements whether it's a spam or not. Such concept is called three-way decision.

Three-way decision has been widely studied in previous contributions and is usually associated with the rough set theory, as Yao have introduced in [18, 19]. According to the three regions in the rough set, Yao concluded three rules shown in (1)-(3) for decision. The values of alpha and beta are computed from six predefined losses when different actions are taken on different objects. Zhou et al.[20] put the three-way decision into application by designing a Email spam filtering algorithm. In their work, an email may be rejected if the risk of labelling it as either "legitimate" or "spam" is high enough. From experimental results, the three-way classifier reached a better weighted accuracy than a normal one.

$$\text{If } \Pr\left(X|[x]\right) \geq \alpha, \text{decide } x \in POS\left(X\right) \tag{1}$$

$$\text{If } \beta < \Pr\left(X|[x]\right) < \alpha, \text{decide } x \in BND\left(X\right) \tag{2}$$

$$\text{If } \Pr\left(X|[x]\right) \leq \beta, \text{decide } x \in NEG\left(X\right) \tag{3}$$

In this work we will use a method which is similar to the process of three-way decision to provide another sentiment classification algorithm on the given dataset, which is combined by the two lexicon-based methods introduced above. First, we apply the algorithm based on general lexicon and the algorithm based on laptop feature lexicon separately on the dataset and get two different results about the sentiment polarity of every piece of comment. Then we combine the two results together and let them vote for a final one. The rule is simple: if two results are the same, then the sentiment polarity of the comment will be the same with the two results; if two algorithms return different sentiment labels, then the comment will be put into the rejection set for further decision, which means we are not assigning a sentiment label to the comment at the moment. This allows us to put aside comments with which the classifiers are not confident enough and thus can reach a better accuracy on the comments who are given an exact sentiment label.

Our last aim in this work is to deal with the comments in the rejection set in order to complete the whole three-way decision concept. The idea is shown below in Fig. 3. We use the supervised learning method to classify the unlabelled comments in the rejection set, and the training data in supervised learning is made up by the comments that are previously labelled by the two lexicon-based algorithms. As [12, 21] have mentioned, the two main weaknesses of supervised learning are that it's difficult to find abundant labelled data for training and the classifier's performance may drop harshly when applied into a new domain of topic. However, our work extracts training data from the unlabelled dataset itself (see Fig. 3), which can solve the two problems at the same time.

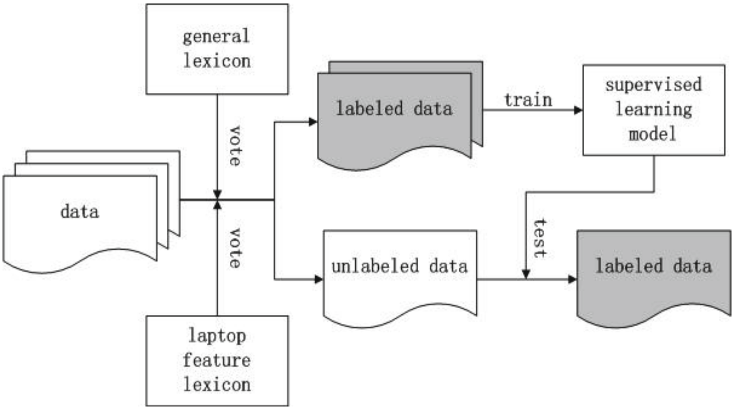


**Fig. 3.** A hybrid algorithm for sentiment analysis

Obviously, the idea has its drawbacks as part of the training data may be wrongly classified by lexicon-based algorithms and thus have an incorrect sentiment label, which may reduce the accuracy of supervised learning. But it has been indicated in [21] that the supervised method can still reach considerable accuracy provided that a large amount of training data have their labels assigned correctly. For this reason, we can expect our supervised learning algorithm to provide a good performance, as [21] have shown in their work.

# 4 Experimental Results

## 4.1 Dataset

Our test dataset is called ChnSentiCorp-nb-4000, which is collected by S.Tan[1]. The dataset consists of 3993 different comments on laptops, 1996 of which is positive and others negative. The average length of comments is around 60 Chinese characters. We will run the sentiment classification scheme introduced in Sect. 3 on the dataset, predicting the sentiment polarity of each comment.

[1] `http://www.searchforum.org.cn/tansongbo/senti_corpus.jsp`

### 4.2 Performance Measure

We will generally evaluate the performance of our method by its classification accuracy. When the general sentiment lexicon and the laptop feature lexicon is independently applied to the dataset, the calculation of Accuracy is presented below, where TP means true positives, TN means true negatives and ALL means the number of comments in the dataset.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{ALL}} \tag{4}$$

After two lexicon-based methods are combined together, the calculation of Accuracy is shown in (5), where TIE means number of comments that are rejected by the classifier because the vote results in a tie. Besides, we will introduce a new measure called Reject (or tie rate) to represent the percentage of comments that are rejected.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{ALL} - \text{TIE}} \tag{5}$$

$$\text{Reject} = \frac{\text{TIE}}{\text{ALL}} \tag{6}$$

When the three-way decision is applied, the calculation of Accuracy is shown in (7). TP2 and TN2 mean the number of true positives and true negatives that are obtained by the tie-breaker (in our work we use Naive Bayesian Classifier as the tie-breaker). 3rdWayAccuracy represents the accuracy our tie-breaker reached on the unassigned comments. For comparison, we set a a baseline where we randomly "guess" a sentiment label for each rejected comment. It is obvious that theoretically the 3rdWayAccuracy for random-choice strategy will be 50%, so (9) represents the baseline accuracy, which our method must be superior to.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN} + \text{TP2} + \text{TN2}}{\text{ALL}} \tag{7}$$

$$\text{3rdWayAccuracy} = \frac{\text{TP2} + \text{TN2}}{\text{TIE}} \tag{8}$$

$$\text{Baseline} = \frac{\text{TP} + \text{TN} + 0.5 * \text{TIE}}{\text{ALL}} \tag{9}$$

### 4.3 Results and Analysis

Before presenting our experimental results, we will firstly introduce a previous work which is applied on the same dataset. The previous work is done by Yu et al.[22], in which paralle1ized sentiment classification algorithms are ran with different weighting methods, feature selection methods and supervised classifiers. The results in [22] is shown below and the average accuracy is around 80.2%.

**Table 4.** Experimental results by Yu et al.

| feature selection method | weighting method | classifier | accuracy |
|---|---|---|---|
| Bigram | Boolean | Knn | 84% |
| Bigram | Boolean | Rocchio | 82.9% |
| Bigram | Tf-idf | Knn | 78.4% |
| Bigram | Tf-idf | Rocchio | 87.7% |
| Sentimen Lexicon | Boolean | Knn | 78.9% |
| Sentimen Lexicon | Boolean | Rocchio | 78.4% |
| Sentimen | Tf-idf Lexicon | Knn | 74.1% |
| Sentimen | Tf-idf Lexicon | Rocchio | 81.6% |
| Substring | Boolean | Knn | 79.5% |
| Substring | Boolean | Rocchio | 68.8% |
| Substring | Tf-idf | Knn | 82.9% |
| Substring | Tf-idf | Rocchio | 85.3% |
| | average | | 80.2% |

Then our experimental results is shown below in Table 5. The first two rows shows the classification accuracy of general lexicon HowNet and the laptop feature lexicon constructed in our work. Results indicate that the laptop feature lexicon can do almost as well as the general sentiment lexicon which is publicly available on the Internet. This suggests that our method of extracting patterns describing product features and estimate their sentiment scores can actually make contributions to sentiment analysis.

Furthermore, when the two lexicons are combined together to make a vote system, the result is shown in the third row. 27.35% of all the reviews are rejected with rule (2) but those not rejected can reach the accuracy of 85.66%. Taking rejection as a third way of decision will provide a more subtle view in classification problems, which can reflect the true state of nature better[20].

According to [20], additional information is needed to deal with the undecided samples. So we apply the algorithm in Fig. 3, use supervised learning as a tie-breaker to classify the unlabelled data. The comparison result is shown in the forth and fifth row in Table 5. Our proposed scheme reached accuracy of 84.90%, which outperforms the baseline of random guess strategy and either of the two lexicons. Also, when comparing Table 4 with Table 5 it is easy to see our proposed scheme is better than [22] in most cases and is superior to its average accuracy as well. Unlike [22], our proposed scheme is unsupervised (as shown in Fig. 3, the "training data" in the supervised process is part of the testing data itself), which again proves its effectiveness. Those results suggest that our method can return satisfying classification results while maintaining its unsupervised feature.

## 5 Conclusion and Future Work

In this work, an unsupervised sentiment analysis scheme is ran on a dataset made up by customers' comments on laptops. A new laptop feature lexicon

**Table 5.** Experimental results

| algorithm | accuracy | reject rate |
|---|---|---|
| General Lexicon | 77.54% | - |
| Feature Lexicon | 74.28% | - |
| GL + FL | 85.66% | 27.35% |
| 3-way: baseline | 75.91% | - |
| 3-way: proposed | 84.90% | - |

which is generated from the dataset itself is introduced to provide an extra view in sentiment categorization. Also, three-way decision methods are used as well in order to get better classification accuracy. Experiment results show that the laptop feature lexicon can do almost as well as a general lexicon in classification accuracy. Besides, when the two lexicons are combined together with the three-way decision method, the classifier can reach a great improvement in its performance.

In the future, we aim to apply our feature lexicon construction methods to other domains to test its validity. Besides, the three-way decision model used in the proposed scheme is simple and intuitive. In the future, we hope to build a three-way classifier which is more theoretically precise in order to make the work more convincing.

## References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
2. Cambria, E., Schuller, B., Xia, Y., et al.: New avenues in opinion mining and sentiment analysis (2013)
3. Yin, C., Peng, Q.: Sentiment Analysis for Product Features in Chinese Reviews Based on Semantic Association. In: International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009, vol. 3, pp. 81–85. IEEE (2009)
4. Zhang, H., Yu, Z., Xu, M., et al.: Feature-level sentiment analysis for Chinese product reviews. In: 2011 3rd International Conference on Computer Research and Development (ICCRD), vol. 2, pp. 135–140. IEEE (2011)
5. Feldman, R.: Techniques and applications for sentiment analysis. Communications of the ACM 56(4), 82–89 (2013)
6. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. Association for Computational Linguistics (2002)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
8. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 271. Association for Computational Linguistics (2004)

9. Hu, X., Tang, L., Tang, J., et al.: 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. ACM (2013)
10. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics (2002)
11. Li, G., Liu, F.: A clustering-based approach on sentiment analysis. In: 2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 331–337. IEEE (2010)
12. Taboada, M., Brooke, J., Tofiloski, M., et al.: Lexicon-based methods for sentiment analysis. Computational Linguistics 37(2), 267–307 (2011)
13. Hogenboom, A., Bal, D., Frasincar, F., et al.: Exploiting emoticons in sentiment analysis. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 703–710. ACM (2013)
14. Hu, X., Tang, J., Gao, H., et al.: Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp. 607–618 (2013)
15. Zhang, L., Liu, B.: Identifying noun product features that imply opinions. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, vol. 2, pp. 575–580. Association for Computational Linguistics (2011)
16. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 25–32. Association for Computational Linguistics (2003)
17. Dong, Z., Dong, Q.: HowNet (2000)
18. Yao, Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180(3), 341–353 (2010)
19. Yao, Y.: Three-way decision: An interpretation of rules in rough set theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 642–649. Springer, Heidelberg (2009)
20. Zhou, B., Yao, Y., Luo, J.: A three-way decision approach to email spam filtering. In: Farzindar, A., Kešelj, V. (eds.) Canadian AI 2010. LNCS, vol. 6085, pp. 28–39. Springer, Heidelberg (2010)
21. Tan, S., Wang, Y., Cheng, X.: Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 743–744. ACM (2008)
22. Yu, Y., Xiang, X., Shang, L.: Research on Parallelized Sentiment Classification Algorithms. Computer Science 40(6), 206–210 (2013)