Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

12-2016

Lexicon knowledge extraction with sentiment polarity computation

Zhaoxia WANG
Singapore Management University, zxwang@smu.edu.sg

Vincent Joo Chuan TONG

Pingcheng RUAN

Fang LI

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Artificial Intelligence and Robotics Commons, and the Numerical Analysis and Scientific Computing Commons

Citation

1

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Published in 16th IEEE International Conference on Data Mining Workshops: 12-15 December, Barcelona, Spain: Proceedings, 978-983.
DOI: 10.1109/ICDMW.2016.0142

Lexicon Knowledge Extraction with Sentiment Polarity Computation

Zhaoxia WANG, Victor Joo Chuan TONG
Social and Cognitive Computing (SCC)
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (ASTAR)
Singapore
wangz@ihpc.a-star.edu.sg; tongjc@ihpc.a-star.edu.sg

Pingcheng RUAN, Fang LI
School of Computer Science and Engineering
Nanyang Technological University,
Singapore
ruanpingcheng@gmail.com; ASFLi@ntu.edu.sg

Abstract—Sentiment analysis is one of the most popular natural language processing techniques. It aims to identify the sentiment polarity (positive, negative, neutral or mixed) within a given text. The proper lexicon knowledge is very important for the lexicon-based sentiment analysis methods since they hinge on using the polarity of the lexical item to determine a text's sentiment polarity. However, it is quite common that some lexical items appear positive in the text of one domain but appear negative in another. In this paper, we propose an innovative knowledge building algorithm to extract sentiment lexicon knowledge through computing their polarity value based on their polarity distribution in text dataset, such as in a set of domain specific reviews. The proposed algorithm was tested by a set of domain microblogs. The results demonstrate the effectiveness of the proposed method. The proposed lexicon knowledge extraction method can enhance the performance of knowledge based sentiment analysis.

Keywords—Sentiment analysis; Lexicon knowledge extraction; Natural Language Processing; Domain knowledge building; Chinese microblog; Weibo

I. INTRODUCTION

With the advent of the Internet and social media, massive volumes of reviews, recommendations, feedback and critiques are generated and shared throughout the Internet [1] [2]. If rigorously analyzed and understood, such social media texts can provide important information which can be used by companies to identify market trends, brand awareness and consumer preferences. They can also be used to provide suggestions for customers for their purchasing decisions.

Hence, not surprisingly, such social media text data has played a critical role for addressing both theoretical issues (e.g., testing theories of internet behaviors) and practical problems (e.g., predicting consumer preferences) [3]. Social media data reflect users' emotions and attitudes on almost every topic and sentiment analysis of such social media data has attracted research interest from both the academia and industry.

Sentiment analysis is a subject that falls within the purview of computational linguistics and targets to determine the attitude of a writer towards a specific topic [1] [2] [3]. On the coarse-grained level, these attitudes can be positive, negative, neutral and mixed [4]. On the fine-grained level, the sentiment can further be categorized into specific emotions, such as happiness, anger, anxiety, among others [5].

Sentiment analysis can be broadly categorized into two main classes: machine-learning based algorithms [6] [7] [8] and lexicon-based approaches [9] [10] [11]. Some researchers classify the methods into three classes and the third approach is the hybrid method, which combines lexicon-based and machine-learning based methods [12].

Machine-learning methods require a large training database to be effective. However, training database is not always available, especially for real time data collected from the Web. Although there are reports of hybrid methods [12] [13], they suffer from the same limitations as machine learning methods, which depend on the training data to be effective.

Lexicon-based methods are commonly used techniques, but the performances of such systems are limited by semantic ambiguity [11], and same words may have different meanings in different domains [14] [15] [16] [17] [18]. It is quite common that some lexicon items appear positive in the text of one domain but appear negative in another [4] [5] [19] [20]. The greater the data volume, the greater the challenge will be for filtering out the noise, understanding the sentiment and identifying useful information from different domain contents.

One paradigm of sentiment analysis is to analyze individual lexical items from the source and then combine their sentiment polarities together to predict the overall sentiment for the text [21]. Typically, they first utilize a general sentiment dictionary to determine the context-free polarity score of individual lexical items. Then they parse and analyze the syntactic structure of the detected lexical items. Lastly, they calculate a total sentiment score for the text. For example, Taboaca et al. compiled their sentiment dictionaries from Epinion, Polartiy Dataset and General Inquirer and had each compiled word assigned with a polarity score by a native English speaker [22]. To determine the sentiment of a sentence, they shifted the detected word's polarity based on its nearby Intensifier (*very*), Negator (*not*) and Diminisher (*less*) [23] [24].

The work is supported by A*STAR Joint Council Office Development Programme "Social Technologies+ Programme".

Basically, this kind of approach mimics the way humans comprehend the text, but the biggest factor that contributes to its effectiveness is the assignment of polarity score on individual words. Firstly, different people with their own cultures and backgrounds may associate different types and levels of subjective interpretations to the same word. On the other hand, the same word may also convey different sentiments under different contexts. A word may express a positive meaning in one domain but express a negative meaning instead in another. Consider the following two examples containing the word *'fast'*, one in the domain of train service and the other in that of the mobile phone.

- *I enjoy the fast train service.*
- *Sucks! The battery level of my mobile phone runs out fast!*

Even with the same word *fast,* the first review utilizes it to show the author's positive opinion on the train service. In contrast, the second review utilizes it to express the author's negative sentiment on the battery level. This phenomenon is referred to as domain-dependent polarity shifting. This is quite a common phenomenon associated with general descriptive words.

In order to address the above issue, we propose an innovative approach to extract sentiment lexicons based on their distribution in a set of domain specific text data. The method is capable of automatically computing the polarity score for each candidate word for the selected domain. The generated domain-specific dictionary can be applied to predict the sentiment of the text data in the same domain.

The whole paper is organized as follows. Section 1 gives a general introduction to the topic, the problem and its significance. Section 2 summarizes the recent related work on word polarity calculation and domain-specific polarity update. Section 3 discusses our algorithm in detail. Section 4 illustrates the algorithm's performance on a set of microblogs and delves into the novelty of the proposed method. Lastly, the whole paper is concluded in Section 5, which will also highlight some potential future work.

II. Related Work

In this section, we examine the most recent methods for calculating the context-free lexicon polarity and the techniques for updating the polarity for specific domains. We will analyze their respective strengths and weaknesses.

In 2002, Turney introduced a simple yet effective unsupervised learning algorithm, Pointwise Mutual Information and Information Retrieval (PMI-IR), to measure the semantic similarity between two words [25]. He assumed that two words with similar meanings tend to appear together. In order to measure the closeness of a specific word to a reference word, he proposed to use the ratio between both words' co-occurrences and the selected word's occurrence in the document set to evaluate their similarity.

In 2006, Ye et al. adapted the PMI-IR algorithm to measure the sentiment polarity of Chinese words [26]. They selected *Excellent* as the positive reference word and *Poor* as the negative reference word. These two words would make up a reference word pair (RWP). The sentiment polarity of the word was computed based on the frequency of its co-occurrence with each reference word in the Google search result.

Similarly, Zhu et al. also selected their own reference words for each sentiment category [27]. However, Zhu et al. utilized a general dictionary HowNet to measure the word's similarity to the reference word. In HowNet, a word may consist of many connotations. Each connotation may consist of multiple sememes, the smallest unit to convey a specific meaning. HowNet organizes the words containing the same sememe together into a set and then organizes these sememe sets into a tree of hyponyms. Zhu et al. proposed to measure a word's shortest path to each of the sentiment reference word in HowNet and compared the path length to calculate the word's polarity.

The above approaches can be summarized into the following steps. Firstly, a reference word for each sentiment pole is determined. Then, the external corpus source is selected. Lastly, the polarity of any word is evaluated based on the closeness relationship with these reference words in the selected corpus. These techniques are effective in deriving a polarity score for any word, but they are elastic to the changes of text domain. As discussed in the introduction, some general descriptive words may be positive in one domain but turn negative in another. Considering this, several researchers have proposed the ideas to update a word's polarity for specific domains, as described in the following paragraph.

Vishnu et al. introduced their method to compile the Domain Independent Dictionary (DID) and Domain Specific Dictionary (DSD) from the SentiWordNet and domain specific corpus [17]. Firstly, they selected a list of candidate words. Then, they calculated the domain independent polarity value (*sw)* for each word from SentiWordNet. The positive value represented positive sentiment and negative value represented the opposite. Afterwards, they measured the difference in the frequency proportionality (*dfp)* of a word in positive and negative text data, which gave its domain peculiar polarity. For each candidate word, if its *dfp* and *sw* had the same sign, it implied that there was no polarity shifting in this domain. This word would be included in DID and *sw* would be its polarity. If *sw* and *dfp* had different signs, the word would be included in DSD and *dfp* would be its polarity value. A similar approach could also be applied to multiple domains. Each text dataset would produce its own DIDs and DSDs. The intersecting part of all DIDs would give a list of general domain independent words and their corresponding polarity values. DSD would generate the list of domain specific words.

Similarly, Demiroz et al. also measured the domain independent polarity from SentiWordNet and term frequency difference from the text corpus [28]. If the signs for a particular word disagreed with each other, they chose to update the word's polarity based on one of the following four strategies:

- **Flip:** Replace the original polarity with its opposite

- **Objective Flip:** Switch the original objective words to either positive or negative. Or, switch the original subjective word to objective
- **Shift:** Shift the polarity of that word towards the other pole
- **DeltaScore:** Compute the new polarity based on the difference in term frequency

Moreover, they also provided different criteria to determine the words for polarity update. For example, update the top k% of words that show a disagreement, update when disagreement exceeds a threshold or iteratively update until no improvement can be made to the validation set.

As we can see from the above ideas, researchers basically calculate the domain independent polarity for all words as a base and then generate a slight modification on the polarity of those words based on the given domain specific corpus. However, the major problem mentioned above still exists: their algorithm's performance will largely depend on the effectiveness of general sentiment dictionary. Any discrepancy from the dictionary may cause great errors in generating the domain dependent polarity. Considering this, we propose an algorithm that does not rely on the general sentiment dictionary. Instead, it relies purely on the word's distribution in the text dataset.

III. The Proposed Algorithm

In this section, we detail our proposed algorithm of extracting sentiment lexicon by calculating the polarity score for a lexical item based on its polarity distribution in the dataset of different sentiment categories. Text preprocessing, part of speech (POS) tagging, polarity computation and word selection are the main processes of the algorithm, which will be detailed in this section.

A. Text Preprocessing

The text preprocessing step basically aims to filter out the noise from the input text. For example, during this step, we remove the tags, urls and other unrecognized entities from the source to minimize the impact for the later analysis. Moreover, for languages like English, in which a word may appear in different forms, we reduce it to its original form so that the analysis load can be decreased. For languages like Chinese, we will employ external tools to segment a sentence into a list of words.

B. POS Tagging

POS tagging is a necessary step to perform sentiment analysis, as the part of speech has a great impact on a word's sentiment polarity. In our algorithm, different parts of speech are assigned different sentiment weights. For example, we assume that adjectives convey the stronger sentiment information than verbs and nouns. So we assign larger sentiment weights to them. Verbs and nouns may also convey sentiment information from time to time. For example, the verb *love* and the noun *congratulations* are often associated with positive sentiment. However, to express the sentiment, we believe adjectives play a much more dominant role than verbs and nouns. In light of the above, we will assign smaller sentiment weights to verbs and nouns. The sentiment weight will limit the maximum polarity score, either positive or negative. We will explain the definition of the sentiment weight in the following subsection.

C. Polarity Computation

In this subsection, we propose an innovative formula to compute the polarity score for each word occurring in the text. The computed score will range from -1 to1. A larger negative value represents a more negative sentiment and a larger positive value represents a more positive sentiment.

In the proposed algorithm, the following notations or parameters are introduced for explanation:

- **w:** A word that appears in the text
- **p(w):** The part of speech of the word, **w**
- **sw(p):** The sentiment weight for the part of speech, p
- **ocr(w,+):** The number of positive text data that contain **w**
- **ocr(w,-):** The number of negative text data that contain **w**

The polarity score **ps** of word **w** is calculated by the proposed formula:

$$ps(w) = \frac{sw(p(w))^{ocr(w,-)+1} - sw(p(w))^{ocr(w,+)+1}}{1 - sw(p(w))} \quad (1)$$

The proposed formula has the following advantages in computing a word's polarity value and the reasons for using the exponential form have been explained:

1. The first time the word **w** is detected in positive text data, it shows strong evidence that this word may be a positive sentiment word. Hence, the algorithm will shift the polarity of **w** to the positive pole to a large scale. However, the extent of shifting will be smaller when more occurrences of **w** are detected in positive text data, as the evidence is not as strong as the first one. This also applies to the word found in negative text data.
2. The polarity (sign of the **ps**) is purely determined by the differences of the word's occurrences in positive and negative text data. When the word occurs for more times in positive text data than in negative ones, it will be assigned a positive polarity value and vice versa.
3. The actual polarity value is determined by how even the word is distributed in positive and negative text data, and the word polarity does not increase linearly with the word frequency. Therefore, we use an exponential form to capture this relationship. For example, in the following Table I, even though the differences between **ocr (w,+)** and **ocr (w,-)** for both words **w1** and **w2** are the same (which is 1), the **ps(w)** values are quite different. The formula can recognize that **w1** is more evenly distributed

and hence it assigns a smaller absolute polarity value to **w1** than to **w2**.

4. The sentiment weight **sw** controls the maximum absolute value of the polarity score. In general, the polarity value of any word will be bounded between -1 / (1 - **sw**) and +1 / (1 - **sw**). We set 0.5 to be the maximum sw and 0 to be the minimum. In this way, **ps** will always fall in the interval between -1 to +1.

Figures 1, 2 and 3 depict the polarity value **ps** for different sentiment weights **sw** when the values of **ocr(w,-)** and **ocr(w,+)** range from 0 to 20.

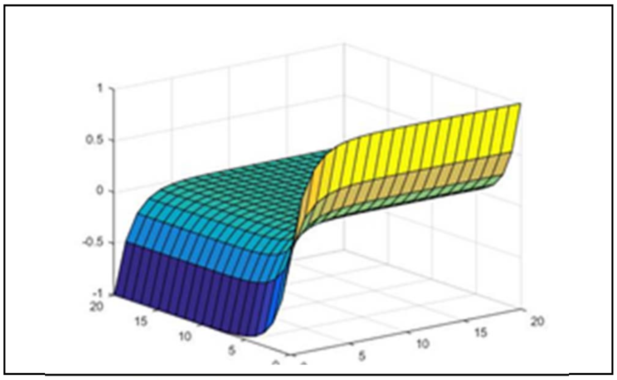

Figure 1: Polarity value for **sw = 0.5**

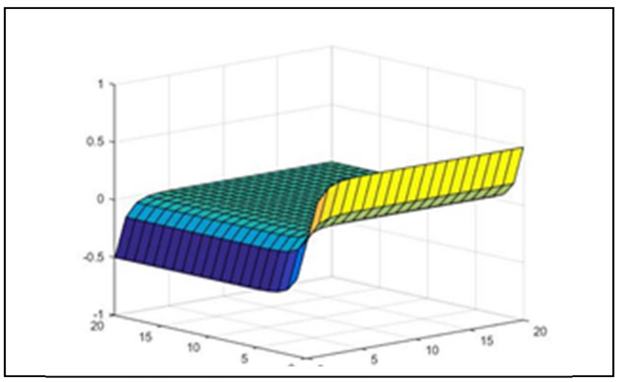

Figure 2: Polarity value for **sw = 0.333**

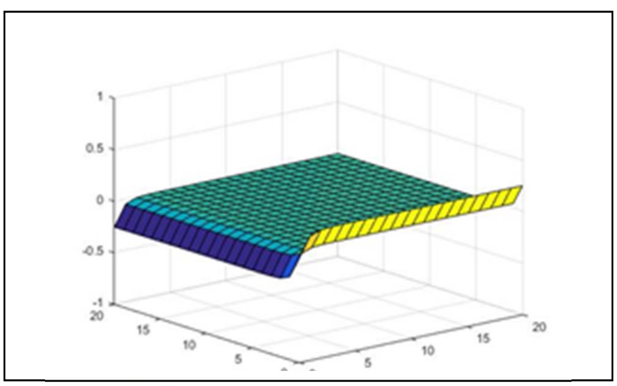

Figure 3: Polarity value for **sw = 0.2**

TABLE I. EXAMPLE OF THE POLARITY VALUES OF TWO WORDS WITH DIFFERENT DISTRIBUTIONS IN REVIEWS

sw = 0.5			
Words	**ocr(w,+)**	**ocr(w,-)**	**ps(w)**
w1	6	5	+0.0156
w2	1	0	+0.5

D. *Word Selection*

At the end of the above process, we have computed a polarity value for each word with different parts of speech that occurs in the text. Then we carry out a sorting process based on the value in the non-descending order. The first few words in the top of the list will be the most negative words for this domain and the last few words will be the most positive ones. Users may handcraft their own strategies to select the word based on their own need. For example, they can select the top and bottom ***k*** percent for their sentiment lexicon. Or, they may set a threshold: any word with its polarity value more than this threshold will be included in the sentiment dictionary for that particular domain.

E. *Pseudocode of the proposed method*

The pseudo-code in the following box summarizes our key steps to extract the specific lexicon from a given dataset.

For each data in the selected text dataset:
Preprocess the data to remove noise
Reduce the word to its original form
Segment the word if necessary
Tag the Part Of Speech
Language Detection, if Chinese, perform Segment

*For each word **w** that occurs in the text:*
Count the number of positive txt data that contain w
Count the number of negative text data that contain w
*Determine the sentiment weight **(sw)** based on w's POS*
Calculate the w's polarity score ps according to formula (1)
*Insert the word into the word list **l** with its polarity value*

*Sort the word list **l** based on the word's polarity value*
*Select the top positive and negative set of words in the list **l***

The following section will test our algorithm using a case from the real world.

IV. EXPERIMENTS

In this section, we applied our algorithm on sets of Chinese microblogs on mobile phone review, which were downloaded from Sina Weibo, a popular Chinese microblog platform. We extracted their domain-specific sentiment words by using the proposed algorithm, annotating the top 50 of them and analyzing the feasibility.

A. Data

In total, the whole text consists of three sets of microblogs. Each set comments on a mobile phone of a particular brand: AA, SS and HH respectively (for confidentiality, the original brand names of the mobile phones we analyzed are represented by these stand-in names). We had each microblog annotated by five independent annotators for the sentiment expressed within the content. The result of their sentiment annotation ranges from positive, negative, neutral and mixed. Then we assigned to the microblog the sentiment, which was agreed by at least 60% of the annotators. At the end of the above step, we have collected in total 178 positive microblogs and 121 negative microblogs. The following Table II gives one example from each category.

TABLE II. EXAMPLES OF THE MICROBLOG DATA

Sentiment	Raw text	English translation
Positive	*换手机了，好高兴，HH p6，[哈哈] [嘻嘻] [嘻嘻]。*	*Had changed mobile phone, so happy, HH p6, [ha ha] [hee hee] [hee hee].*
Negative	*AA 是就是矫情，冬天充电还要把手机捂热了才能充上电，. . . ，真是矫情。*	*AA is hypocritical. It must be warm first before it can be charged in winter, ..., it is really hypocritical.*

B. Algorithm Parameters and Configuration

With the microblogs collected and labeled, we went on to perform the word segmentation and POS tagging. Here we employed FudanNLP, a Chinese-based natural language processing suite, to accomplish this task [29].

Meanwhile, we assigned different sentiment weights **sw** to different parts of speech. For the proposed algorithm, the range of **sw** is [0, 0.5] as explained in the previous subsection. 0.5 is the maximum value and 0 is the minimum. As discussed, the adjectives, for most of the time, convey the strongest sentiment, followed by the verbs and the nouns, as shown in Table III. Hence, we assigned their sentiment weights in descending order, as illustrated in the table.

TABLE III. ALGORITHM PARAMETERS USED IN THIS PAPER

Sentiment Weight (sw) for different part of speeches	
Part of Speech	Sentiment Weight
Adjectives	0.5
Verb	0.333
Noun	0.2

C. Result and Discussion of the algorithm

With the above parameters and configuration of the algorithm, we computed the polarity value for each word that occurred in the text based on its polarity distributions in the microblogs as well as its part of speech. The polarity score **ps** of word **w** is obtained by using equation (1), which was sorted based on the polarity score in non-descending order as shown in Tables IV and V.

Tables IV and V list the top 5 positive and negative words obtained by using the proposed method.

TABLE IV. THE EXAMPLES OF POSITIVE WORDS EXTRACTED WITH HIGHER POLARITY SCORE

Sentiment	Word	English translation	Part of speech	Polarity value
Positive	便宜	Cheap	Adjective	0.996
	强大	Powerful	Adjective	0.984
	好看	Beautiful	Adjective	0.937
	强	Strong	Adjective	0.875
	高兴	Happy	Adjective	0.75

TABLE V. EXAMPLES OF NEGATIVE WORDS EXTRACTED WITH HIGHER POLARITY SCORE

Sentiment	Word	English translation	Part of speech	Polarity value
Negative	烦躁	Anxious	Adjective	-0.968
	差	Lousy	Adjective	-0.875
	难受	Aweful	Adjective	-0.75
	矫情	Hypocritical or Unreasonable	Adjective	-0.75
	丢人	Disgraceful	Adjective	-0.75

We checked their consistency with the algorithm's prediction and compiled the results which counted the number of consistent labeling in the top N positive and negative words. Among the most 10 subjective words, the accuracy of the proposed method reached 90% or even higher. When a larger N was selected, the accuracy was still maintained around 75%. We selected the top positive words with $ps(w) > 0.5$ and the top negative words with $ps(w) < -0.5$ to construct the knowledge lexicons. The detailed evaluation results are shown in Table VI.

TABLE VI. THE PERFORMANCE OF THE PROPOSED METHOD

	Positive	Negative	Mean
Precision	79.22%	75.00%	77.11%
Recsall	84.72%	67.35%	76.03%
F1	81.88%	70.97%	76.42%
Accuracy	77.69%		

It is observed that the accuracy (77.69%) of the algorithm can be compared to the previous work with a mean accuracy of 75.2% [17]. Proper knowledge is a key issue for the knowledge based methods [30]. The knowledge extracted by this algorithm have been used to enhance the knowledge base of sentiment analysis methods [23] [24] to obtain better results. Users of this algorithm may select their own criteria. For example, they could set a threshold on the polarity score or select a fixed number of words with the largest absolute polarity value. The accuracy of the proposed method can be 100% if only words that have very large absolute polarity values are selected.

V. CONCLUSION

In this paper, we discussed the importance of having a proper lexicon knowledge base and reviewed the recent popular approaches to accomplish this task from the domain-specific text data. After reviewing the pros and cons of previous methods, we proposed our own algorithm to extract the lexical items and compute their polarity score. We tested our algorithm on sets of Chinese microblogs in the domain of mobile phones. The experimental result showed an acceptable performance, though there is still room for improvement:

- In our current algorithm design, the words that occur in the text are all considered as candidates for the sentiment lexicon items. In fact, most of them are not and shall be eliminated from consideration. Also, we have been using relatively small datasets for testing but we plan to use larger datasets for further study and testing.
- The current algorithm is based on a statistical method without leveraging on syntactic structures and semantics. We may incorporate these factors in the future for better performance.
- The assignment of sentiment weight to different parts of speeches is heuristic in this paper and we are currently conducting further research to derive these weights in a principled manner.
- The current experiment only considers words (unigram). The experiment considering both words and phrases (bigram, trigram and n-gram) is in the process of being carried out and the results will be reported in the future.

ACKNOWLEDGMENT

The team of the project "SentiMo - Advanced Social Media Analytics" provided great support for this research work. The authors would like to thank Dr. Kenneth Kwok, Dr Ho Seng Beng and their team for their discussion and help in conducting this research.

REFERENCES

[1] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, "SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives," in *the 26th International Conference on Computational Linguistics (COLING), Osaka*, 2016.

[2] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, 2016.

[3] Z. Wang, V. J. C. Tong, X. Xin, and H. C. Chin, "Anomaly setection through enhanced sentiment analysis on social media data," *2014 IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, pp. 917–922, Dec. 2014.

[4] Y. Bae and H. Lee, "Sentiment analysis of Twitter audiences: Measuring the positive or negative influence of popular twitterers," *J. Am. Soc. Inf. Sci. Technol.*, vol. 63, no. 12, pp. 2521–2535, 2012.

[5] Z. Wang, V. J. C. Tong, and D. Chan, "Issues of social data analytics with a new method for sentiment analysis of social media data," *2014 IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, pp. 899–904, Dec. 2014.

[6] A. Balahur and M. Turchi, "Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 56–75, Jan. 2014.

[7] P. Chikersal, S. Poria, E. Cambria, A. Gelbukh, and C. E. Siong, "Modelling public sentiment in Twitter: using linguistic patterns to enhance supervised learning," *Comput. Linguist. Intell. Text Process.*, pp. 49–65, 2015.

[8] Z. Wang, V. J. C. Tong, and H. C. Chin, "Enhancing machine-learning methods for sentiment classification of web data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8870, pp. 394–405, 2014.

[9] Y. Rao, J. Lei, L. Wenyin, Q. Li, and M. Chen, "Building emotional dictionary for sentiment analysis of online news," *World Wide Web*, vol. 17, pp. 723–742, Jun. 2014.

[10] A. Hogenboom, B. Heerschop, F. Frasincar, U. Kaymak, and F. De Jong, "Multi-lingual support for lexicon-based sentiment analysis guided by semantics," *Decis. Support Syst.*, vol. 62, pp. 43–53, 2014.

[11] A. Balahur, R. Mihalcea, and A. Montoyo, "Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 1–6, Jan. 2014.

[12] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexicon-based and learning-based methods for Twitter sentiment analysis," *Int. J. Electron. Commun. Soft Comput. Sci. Eng.*, vol. 89, pp. 1–8, 2015.

[13] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," *J. Informetr.*, vol. 3, no. 2, pp. 143–157, Apr. 2009.

[14] Y. Choi, Y. Kim, and S.-H. Myaeng, "Domain-specific sentiment analysis using contextual feature generation," *Sci. Technol.*, pp. 37–44, 2009.

[15] S. Park, W. Lee, and I. C. Moon, "Efficient extraction of domain specific sentiment lexicon with active learning," *Pattern Recognit. Lett.*, vol. 56, pp. 38–44, 2015.

[16] J. Bross and H. Ehrig, "Automatic construction of domain and aspect specific sentiment lexicons for customer review mining," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, 2013, pp. 1077–1086.

[17] K. S. Vishnu, T. Apoorva, and D. Gupta, "Learning domain-specific and domain-independent opinion oriented lexicons using multiple domain knowledge," *Contemp. Comput. (IC3), 2014 Seventh Int. Conf.*, pp. 318–323, 2014.

[18] E. Cambria, A. Hussain, C. Havasi, and C. Eckl, "Sentic computing: exploitation of common sense for the development of emotion-sensitive systems," *LNCS*, vol. 5967, pp. 148–156, 2010.

[19] E. Cambria, J. Fu, F. Bisio, and S. Poria, "AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis," *AAAI*, pp. 508–514, 2015.

[20] E. Cambria, T. Mazzocco, A. Hussain, and C. Eckl, "Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space," *LNCS*, vol. 6677, pp. 601–610, 2011.

[21] G. Vinodhini and R. Chandrasekaran, "Sentiment analysis and opinion mining: A survey," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, pp. 282–292, 2012.

[22] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguist.*, vol. 37, no. 2, pp. 267–307, 2011.

[23] Z. Wang, R. S. M. Goh, and Y. Yang, "SentiMo-A Method and system for fine-grained classification of social media sentiment and emotion patterns," Singapore Patent Application10201407766R, 2014.

[24] Z. Wang and V. J. C. Tong, "ChiEFS-A method and system for Chinese hybrid multilingual emotion fine-grained sensing of text data," Singapore Patent Application No. 10201601413Q, 2015.

[25] P. D. Turney, "Mining the web for synonyms: PMI-IR versus LSA on TOEFL," *Proc. 12th Eur. Conf. Mach. Learn. (ECML-2001), Freiburg, Ger.*, pp. 491–502, 2001.

[26] Q. Ye, B. Lin, and Y. Li, "Sentiment classification for Chinese reviews: A comparison between SVM and semantic approaches," *Mach. Learn. Cybern. 2005. Proc. 2005 Int. Conf.*, vol. 4, no. August, pp. 2341–2346, 2005.

[27] Y. Zhu, Z. Wen, P. Wang, and Z. Peng, "A method of building Chinese basic semantic lexicon based on word similarity," *Pattern Recognition, 2009. CCPR 2009. Chinese Conf.*, pp. 1–4, Nov. 2009.

[28] G. Demiroz, B. Yanikoglu, D. Tapucu, and Y. Saygin, "Learning domain-specific polarity lexicons," *Data Min. Work. (ICDMW), 2012 IEEE 12th Int. Conf.*, pp. 674–679, Dec. 2012.

[29] X. Qiu, Q. Zhang, and X. Huang, "FudanNLP : A toolkit for Chinese natural language processing," *ACL (Conference Syst. Demonstr.*, pp. 49–54, 2013.

[30] T. Yang, H. Pen, Z. Wang, and C. S. Chang, "Feature knowledge based fault detection of induction motors through the analysis of stator current data," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 549–558, Mar. 2016.