Research Article

Automatic Approach of Sentiment Lexicon Generation for Mobile Shopping Reviews

Jun Feng ¹, **Cheng Gong** ¹, **Xiaodong Li** ¹ and **Raymond Y. K. Lau**²

¹College of Computer and Information, Hohai University, Nanjing 211100, China

²Department of Information Systems, City University of Hong Kong, Hong Kong

Correspondence should be addressed to Xiaodong Li; xiaodong.c.li@outlook.com

Received 30 March 2018; Revised 11 July 2018; Accepted 30 July 2018; Published 12 August 2018

Academic Editor: Zhe He

Copyright © 2018 Jun Feng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The dramatic increase in the use of smartphones has allowed people to comment on various products at any time. The analysis of the sentiment of users' product reviews largely depends on the quality of sentiment lexicons. Thus, the generation of high-quality sentiment lexicons is a critical topic. In this paper, we propose an automatic approach for constructing a domain-specific sentiment lexicon by considering the relationship between sentiment words and product features in mobile shopping reviews. The approach first selects sentiment words and product features from original reviews and mines the relationship between them using an improved pointwise mutual information algorithm. Second, sentiment words that are related to mobile shopping are clustered into categories to form sentiment dimensions. At each sentiment dimension, each sentiment word can take the value of 0 or 1, where 1 indicates that the word belongs to a particular category whereas 0 indicates that it does not belong to that category. The generated lexicon is evaluated by constructing a sentiment classification task using several product reviews written in both Chinese and English. Two popular non-domain-specific sentiment lexicons as well as state-of-the-art machine-learning and deep-learning models are chosen as benchmarks, and the experimental results show that our sentiment lexicons outperform the benchmarks with statistically significant differences, thus proving the effectiveness of the proposed approach.

1. Introduction

With the rapid development of smartphones, mobile shopping, which is already popular, is expected to grow faster. After shopping, people provide a large number of reviews about different kinds of products on the Internet. Different products could be preferred by different consumer groups. Hence, it is becoming increasingly important to learn about a customer's emotional inclinations and favorite products through online reviews. Sentiment classification can be performed using machine-learning, lexicon-based, and hybrid approaches. Sentiment lexicons are important resources for these approaches. The analysis of sentiment orientation is widely known as a domain-specific task. However, almost all the existing sentiment lexicons are general lexicons, which are not suitable for the analysis of product reviews on the Internet. Thus, automatic construction methods for sentiment lexicons have attracted increasing attention recently, especially

methods for constructing sentiment lexicons aimed at mobile shopping.

Sentiment analysis, which is also called opinion mining, review mining (appraisal extraction), or attitude analysis, is the task of detecting, extracting, and classifying opinions, sentiments, and attitudes concerning different topics [1]. In a machine-learning approach, sentiment analysis can be considered as a supervised classification task. Pang et al. [2] solved the sentiment classification problem by training the classifier. However, most machine-learning approaches rely on features that are engineered by machine-learning methods. In a lexicon-based approach, a dictionary is created to judge whether the polarity of words in the text is positive or negative. For example, Turney [3] scanned a review for phrases that matched certain patterns (adjectives and adverbs) and then added up all sentiment orientations to compute the orientation of a document. A hybrid approach combines both the above approaches and has a relative

advantage in sentiment analysis. Ortigosa et al. [4] developed a lexicon from a corpus and then chose sentiment words along with the labeled class as the input features for a machine-learning classification method. Sentiment lexicons play a key role in a majority of the above methods.

A sentiment lexicon (or an opinion lexicon) is a list of words and phrases that are commonly used to express positive or negative sentiments [5]. Researchers have proposed many approaches to compile these sentiment words. Technically, the existing automatic lexicon construction methods for both English and Chinese languages are mainly divided into corpus-based and knowledge-based methods. Turney [3] developed a corpus-based method in which the sentiment orientation of a word was judged by using pointwise mutual information (PMI) to describe the closeness of the word and seed words. Knowledge-based methods require a relatively complete knowledge base. Hu and Liu [6] constructed a sentiment lexicon by searching for the synonyms and antonyms of a word in WordNet. For a specific domain, the sentiment lexicon constructed from the corresponding domain corpus is more practical. When building a sentiment lexicon for online product reviews, the product features modified by sentiment words are also very important factors [7]. However, the existing general sentiment lexicons usually include only limited common words, and these words are divided into binary or other fixed categories according to the sentiment orientation.

In this paper, we present a novel method to construct a domain-specific sentiment lexicon by mining the relationship between sentiment words and product features in a specific corpus. In our approach, first, a sentiment matrix is constructed based on the relationship between sentiment words and product features. Every row of the sentiment matrix is regarded as a vector representation of the sentiment word. The sentiment words in the matrix space are clustered based on the distance between the vectors. Second, sentiment words that are related to mobile shopping are clustered into categories to form sentiment dimensions. In the process of building the sentiment matrix, the idea of term frequency-inverse document frequency (TFIDF) is utilized to screen the product features. Furthermore, the traditional PMI algorithm is improved to obtain a new algorithm called EPMI, which is more suited to mobile shopping reviews. Extensive experiments are performed on seven different domain product reviews, which include reviews in both Chinese and English. Compared to two popular general lexicons as well as state-of-the-art machine-learning and deep-learning models, our lexicon can obtain satisfactory classification performance. The experimental results also show that the filtering of product features and the application of the EPMI algorithm can greatly improve the performance of our lexicon for mobile shopping reviews.

The rest of the paper is structured as follows. Discussions on sentiment classification and lexicon generation and a review of the most recent research are presented in Section 2. Our methods for constructing the sentiment lexicon for mobile shopping reviews and a walk-through example of our methods are presented in Section 3. The experimental setup and results are described in Section 4. The conclusions of the paper are summarized in Section 5.

2. Related Work

This section is structured as follows. In the first part of this section, we review previous works on sentiment classification approaches. In the second part, we summarize works on approaches for sentiment lexicon creation. In addition, we briefly introduce the sentiment dimensions considered in the lexicon and product feature identification for product reviews.

2.1. Sentiment Classification. Sentiment classification aims to automatically classify the text of reviews written by customers into positive or negative opinions. Sentiment classification techniques can be roughly divided into machine-learning, lexicon-based, and hybrid approaches [8].

Machine-Learning Approaches. In such approaches, the analysis of customers' emotional inclinations is considered to be a problem of polarity classification. Pang et al. [2] applied three machine-learning methods (naive Bayes (NB), maximum entropy, and a support vector machine (SVM)) to sentiment classification as a form of traditional topic-based categorization. Zhang et al. [9] used machine learning (NB and SVM) to classify the sentiments expressed in restaurant reviews written in Cantonese. Li et al. [10] adopted extreme learning machine and deep-learning architecture to improve feature representations for text classification. Enríquez et al. [11] showed how a vector-based word representation obtained via Word2Vec can help in improving the results of a document classifier based on the bag-of-words model. However, these supervised machine-learning techniques require a large corpus of training data, and their performance is acceptable only if the match between the training and test data is good.

Lexicon-Based Approaches. These approaches adopt a lexicon to perform sentiment analysis by counting and weighting sentiment words that have been evaluated and tagged [12]. Nasukawa and Yi [13] developed a method to determine subject favorability by creating a sentiment lexicon containing 3513 sentiment terms. Qiu et al. [14] used a lexicon-based approach to identify sentiment sentences in contextual advertising. The most common lexicon resources are SentiWordNet, WordNet, and ConceptNet, and among these resources, SentiWordNet is the most widely used [15].

Hybrid Approaches. Nowadays, researchers are also using combined approaches, in which two or more approaches are combined to achieve better accuracy. Sindhwani and Melville [16] presented a unified framework in which lexical background information, unlabeled data, and labeled training examples can be effectively combined. Li et al. [17] set up a system to analyze the market impact by combining the stock price and news sentiment. Ortigosa et al. [4] performed sentiment classification and sentiment change detection on Facebook comments using a hybrid approach. They combined lexicon-based and machine-learning methods by considering a lexicon as the source of features and using a classification model to evaluate the lexicon; this approach is similar to the one used in our experiments in this study.

2.2. Lexicon Creation. A sentiment lexicon is an important tool for identifying the sentiment polarity of reviews provided by mobile users [18]. Two methods are commonly used to generate sentiment lexicons: knowledge-based and corpus-based methods.

Knowledge-Based Methods. These methods exploit available lexicographical resources such as WordNet or HowNet. Hu and Liu [6] developed a lexicon by searching for the synonyms and antonyms of a word in WordNet. Kamps [19] inferred that the greater the closeness of two words, the smaller the number of iterations required to determine the synonymous relationship between the words. Both these studies used the relationship between words in a knowledge base. The main strategy in these methods is to first manually collect an initial seed set of sentiment words and their orientations and then search for their synonyms and antonyms in a knowledge base to expand this set [12]. However, very few complete and robust knowledge bases are available for the Chinese language.

Corpus-Based Methods. These methods depend on syntactic patterns or patterns that occur together along with a seed list of opinion words to find other opinion words in a large corpus [20]. Hatzivassiloglou and McKeown [21] found that, with a change in the emotional polarity in the text, the turning point appears but concatenation does not. Based on the idea that the emotional polarity of a word tends to be consistent with the emotional polarity of its neighboring words, Turney and Littman [22] constructed a dictionary from a large corpus. Both these works [21, 22] are based on a corpus rather than a knowledge base. The corpus-based approach has a major advantage in that it can find domain-specific words and their orientations if a domain-specific corpus is used in the discovery process. Therefore, our work also focuses on a corpus-based approach. In addition, PMI is commonly used in this approach to exploit the syntactic patterns of cooccurrence patterns. Turney and Littman [22] used PMI and latent semantic analysis to measure the correlation between two words, and this method, which uses PMI to calculate the correlation between a word and seed word, is called semantic-orientation PMI (SO-PMI). Yang et al. [23] introduced a method based on SO-PMI to construct a sentiment lexicon and improved the SO-PMI model based on user behavior. In the process of our lexicon construction, we improve the traditional PMI to make it more suitable for mobile shopping reviews.

In the process of lexicon construction, we focus on two issues: the sentiment dimensions of the lexicon, and feature or topic identification in product review domains.

Sentiment Dimensions. Ekman [24] found that humans have six basic emotional categories: happiness, sadness, fear, surprise, anger, and jealousy. Ekman's theory, which is accepted by numerous psychologists and linguists, is widely used in the field of sentiment analysis. Rubin et al. [25] presented an empirically verified model on the basis of the idea [26] that an emotion can be divided into eight categories with two major bipolar dimensions: positive and negative effects.

TABLE 1: Mathematical symbols and their meanings.

Symbol	Meaning
A, B, F	Sentiment matrix
C	Matrix of relationship between features
$E = \{e_1, e_2, \dots, e_n\}$	Set of sentiment words
$M = \{m_1, m_2, \dots, m_p\}$	Set of primitive product features
$M' = \{m'_1, m'_2, \dots, m'_t\}$	Set of product features after filtering
$D = \{d_1, d_2, \dots, d_n\}$	Set of reviews

Although early approaches simply focused on this binary classification [27], we not only consider the two polarities but also anticipate that sentiment words can be reasonably clustered into finer-gained categorizations.

Feature Identification. Considering that many words in different fields may have different sentiment polarities, it is necessary to explicitly extract the sentiment words and topics or product features, especially in the mobile review domain. Fast et al. [28] found out that using experts or crowdsourcing to construct domain-specific sentiment lexicons is very difficult. Zhang et al. [29] proposed a hybrid method that combined Apriori and PMI to extract product features. Mishne [30] chose the part of speech (POS) and word counts as features in a text classification task. In our research, the primitive product feature extraction also uses the POS as a selection criterion.

3. Methods

In this section, we present our proposed framework to generate domain-specific sentiment lexicons for mobile shopping. Figure 1 shows the framework of our method. The domain-specific lexicon is based on the relationship between sentiment words and product features modified by the sentiment words. A sentiment matrix is adopted to represent the relationship between the sentiment words and product features. First, we use PMI to express the relationship between sentiment words and product features. Second, we use TFIDF to filter product features so as to reduce the matrix dimension. Finally, we improve the traditional PMI to develop a new algorithm called EPMI, which is used to build a new sentiment matrix. Each row in the sentiment matrix is a vector representation of the sentiment word. After obtaining the sentiment matrix, we cluster the sentiment words into several categories based on the distance between their vector representations. The mathematical symbols used in the process of construction are listed in Table 1.

3.1. Building of Primitive Sentiment Matrix. To perform the key step of mining the relationship between sentiment words and product features, we need to determine the sentiment words and product features in the corpus. To choose the terms from the corpus as candidate words, we use the POS.

Sentiment words are commonly used to express positive or negative sentiments. Sentiment lexicons usually contain such words, which can indicate the sentiment polarity (e.g.,

FIGURE 1: Framework of method used for constructing domain-specific lexicon for mobile shopping.

“good” and “wonderful” indicate positive opinions, whereas “rubbish,” “cheap,” and “terrible” indicate negative opinions). In mobile shopping reviews, a number of verbs can also indicate the sentiment polarity (e.g., “like” and “love” indicate positive opinions, whereas “dislike” and “refund” indicate negative opinions). In some previous studies [31, 32], the words whose POS is an adjective or adverb are considered as sentiment words. The sentiment lexicons developed or used in some other studies [6, 33] are also mainly concerned with adjectives and adverbs. In addition, product features in the product review domain are usually nouns or noun phrases found in review sentences [6]. Therefore, we choose adjectives, adverbs, and verbs as sentiment words and choose nouns as primitive product features. For instance, in the hotel review “The food in the dining room is really good, the breakfast tastes good,” the product features are “dinning,” “breakfast,” and “food,” and the sentiment words are “good” and “tastes.”

If a sentiment word A modifies a product feature B , we consider that there is a relationship between them. In mobile shopping reviews, this relationship can be shown as a phenomenon of cooccurrence. We use PMI to quantify this type of cooccurrence relationship. PMI is defined as

$$\text{PMI}(\text{word}_1, \text{word}_2) = \log_2 \left(\frac{p(\text{word}_1, \text{word}_2)}{p(\text{word}_1)p(\text{word}_2)} \right) \quad (1)$$

Here, $p(\text{word}_1, \text{word}_2)$ is the cooccurrence probability of word_1 and word_2 in the local window and is expressed as

$$p(\text{word}_1, \text{word}_2) = \frac{\text{count}(\text{word}_1, \text{word}_2)}{N} \quad (2)$$

where N is the total number of words contained in the corpus. $\text{count}(\text{word}_1, \text{word}_2)$ represents the number of occurrences of the two words in the local window. Similarly, the frequency of each word can be obtained as

$$p(\text{word}) = \frac{\text{count}(\text{word})}{N} \quad (3)$$

In (1), $p(\text{word}_1)p(\text{word}_2)$ gives the probability of cooccurrence if these two words are statistically independent. The ratio of $p(\text{word}_1, \text{word}_2)$ to $p(\text{word}_1)p(\text{word}_2)$ is thus a measure of the degree of statistical dependence between the words.

The PMI value between the sentiment words and product features can reflect the relationship between them. By calculating the PMI value between all the sentiment words and product features, we can obtain a sentiment matrix that contains the relationship between the sentiment words and product features. Let us denote $E = \{e_1, e_2, \dots, e_n\}$ as the set of sentiment words and $M = \{m_1, m_2, \dots, m_p\}$ as the set of product features. Matrix A , as shown below, consists of n rows and p columns.

Definition 1. Sentiment matrix A : the rows represent the sentiment words, whereas the columns represent the product features. The value of each cell w_{ij} is given by $\text{PMI}(e_i, m_j)$.

$$A = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1p} \\ w_{21} & w_{22} & \cdots & w_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{np} \end{bmatrix} \quad (4)$$

In the above matrix, each sentiment word e_i can be represented as a vector $\vec{e}_i = [w_{i1}, w_{i2}, \dots, w_{ip}]$. Sentiment matrix A is the primitive sentiment matrix, and this matrix is optimized, as described in the next subsection.

3.2. Filtering of Product Features. So far, we have obtained the primitive sentiment matrix A , and each sentiment word e in the matrix can be represented as a vector. According to our approach, these vectors should be clustered into several categories. However, we found that the number of product features is very large because we consider all nouns as product features. Consequently, the word vector will face the dimension disaster problem. The clustering of high-dimensional data is still a challenging problem because of the curse of dimensionality [34]. In addition, the use of high dimensions will result in low computational efficiency, especially in mobile computing. In Hu and Liu's study [6], only those product features regarding which many people have expressed their opinions are reserved. Similarly, we also select key product features from the primitive nouns. Next, we will describe our feature selection method in detail.

The high-dimension problem stems from the large number of nouns in the corpus. The number is large because we choose all nouns as product features. For instance, consider the product review "This hotel is great, I can recommend my mom to live next time." The word "mom" and "time" will be treated as product features, but these words do not represent any features of the hotel. In addition, this type of nouns can be found everywhere in mobile shopping reviews. Therefore, it is necessary to filter out the key product features rather than choose all nouns as the product features. Product features should be nouns that frequently appear in a particular category of product reviews and rarely appear in other categories. Therefore, we use the idea of TFIDF to select real product features. TFIDF is defined as

$$\text{TFIDF}(\text{word}) = \text{TF}(\text{word}) \times \text{IDF}(\text{word}) \quad (5)$$

Here, $\text{TF}(\text{word})$ means the term frequency of the word in the document. $\text{IDF}(\text{word})$ means the inverse document frequency, that is, whether the word is common or rare across all documents. It is important to note that the TFIDF value of the same word may be different in different documents. However, TFIDF is usually used for documents rather than pieces of reviews. There may be thousands of comments about a single product. We just need to merge the same kind of comments together to form the corresponding document.

From (5), we can obtain the TFIDF value of words in different documents. Unlike the analysis described in the

previous subsection, here, we choose the nouns whose TFIDF values are relatively high in the document as product features of the product. Unexpectedly, we find that the nouns whose TFIDF values are relatively high happen to be words that are closely related to the reviewed product. For example, if there are numerous reviews about a hotel, we can retrieve words such as "bathroom" and "air-conditioning" from the corresponding document. When we are commenting on a hotel, we often refer to the "bathroom" or "air-conditioning" in the hotel. However, these two words rarely appear in the reviews of products from other domains such as the electronics domain. We can certainly define a threshold α that the TFIDF value of real product features must reach. Let us denote $M' = \{m'_1, m'_2, \dots, m'_t\}$ where $M' \subseteq M$ as the set of the remaining product features after filtering by TFIDF. Accordingly, we can obtain another sentiment matrix that is similar to sentiment matrix A . This matrix (sentiment matrix B) consists of n rows and t columns.

Definition 2. Sentiment matrix B : this matrix can be considered to be part of sentiment matrix A . The rows represent the sentiment words, whereas the columns represent the product features after filtering by TFIDF. The value of each cell w_{ij} , which is the same as that in sentiment matrix A , is given by $\text{PMI}(e_i, m'_j)$.

In matrix B , each sentiment word e_i can be represented as a vector $\vec{e}_i = [w_{i1}, w_{i2}, \dots, w_{it}]$. Here, t can be considerably less than p when the threshold α is set appropriately. Compared to sentiment matrix A , sentiment matrix B can effectively solve the high-dimension problem in word embedding. However, there are still some defects in the sentiment matrix, which will be elaborated in the next subsection.

3.3. Optimization of Sentiment Matrix by EPMI. Here, we introduce an example from hotel reviews to further explain the defect in sentiment matrices A and B . We focus on two sentiment words ($e_1 = \text{"rich"}$ and $e_2 = \text{"hearty"}$) and two product features ($m_1 = \text{"food"}$ and $m_2 = \text{"breakfast"}$). Both these sentiment words can be used to express opinions about a wide variety of foods. The meanings of these two sentiment words are very similar, and these words are commonly used in the hotel review domain. If we just consider the two features m_1 and m_2 , e_1 and e_2 can be represented as $\vec{e}_1 = [w_{11}, w_{12}]$ and $\vec{e}_2 = [w_{21}, w_{22}]$ in the sentiment matrix. w_{ij} is given by $\text{PMI}(e_i, m_j)$.

As is well known, the distance or angle between word vectors can be considered to be the similarities between words. The greater the similarity between two words, the shorter the distance between them. However, in a hotel review, the two sentiment words (e_1, e_2) and two product features (m_1, m_2) can be matched with each other flexibly. Although some customers may usually modify m_1 with e_1 and m_2 with e_2 , they may rarely modify m_1 with e_2 and m_2 with e_1 . This means that the PMI value of (e_1, m_1) and (e_2, m_2) is relatively high, but the PMI value of (e_1, m_2) and (e_2, m_1) is very low. Therefore, an illusion is created that e_1 and e_2 are irrelevant in the two dimensions of m_1 and m_2 . This irrational result stems from the flexibility of product reviews and the

Input: $M = \{m_1, m_2, \dots, m_p\}, M' = \{m'_1, m'_2, \dots, m'_t\}, D = \{d_1, d_2, \dots, d_n\}$

Output: Matrix $C[t][p]$

```

(1)  $i = 1, C[t][p] = \text{zero matrix}$ 
(2) while  $i \leq t$  do
(3)    $j = 1$ 
(4)   while  $j \leq p$  do
(5)     for each  $d \in D$  do
(6)       if  $(m'_i, m_j)$  in  $d$  &&  $m'_i \neq m_j$  then
(7)          $C[i][j] = C[i][j] + 1$ 
(8)        $j = j + 1$ 
(9)      $i = i + 1$ 
(10) For each row  $row \in C[t][p]$  do
(11)    $row = \text{normal}(row)$ 
(12) Return  $C$ 

```

ALGORITHM 1: Mining relationship between product features.

diversity of vocabulary in mobile shopping reviews. Although e_1 rarely modifies m_2 , it cannot be simply considered to be irrelevant.

When we consider the relationship between a sentiment word and product feature, it is not sufficient to just calculate the PMI value of these two words directly. We still need to consider the relationship between the sentiment word and other product features that are related to the initial product feature. In the mobile shopping reviews about a hotel, there are many features related to m_2 (*breakfast*), such as “food” and “dinning.” Therefore, when we calculate the PMI value of e_1 and m_2 , we consider the cooccurrence of not only e_1 and m_2 but also e_1 and m_1 or other product features related to m_2 . We use $u_{ij} \in [0, 1]$ to reflect the degree of correlation between the two product features m_i and m_j . The larger the value of u_{ij} , the more relevant m_j to m_i . $u_{ij} = 0$ indicates that the two features m_i and m_j are irrelevant. In particular, if the features m_i and m_j represent the same feature, the value of u_{ij} between them is zero. Considering all the p product features contained in the corpus, we define EPMI as

$$\text{EPMI}(e_i, m_j) = \text{PMI}(e_i, m_j) + \sum_{k=1}^p u_{jk} * \text{PMI}(e_i, m_k) \quad (6)$$

Once we know the value of u_{ij} between any two features m_i and m_j , we can obtain the EPMI value on the basis of the PMI value. Considering that we can screen the features according to the method described in the previous subsection, we focus on the correlation between the remaining product features after filtering and all the primitive features. We assume that the more frequently two features appear in the same review, the higher the correlation between them is. The pseudocode for mining the relationship between them is presented in Algorithm 1.

Following the earlier definitions, M is still the set of all product features for a given kind of production, and M' is the set of product features obtained by carrying out filtering using the approach described in the previous subsection. D is the set of reviews related to the product. (m'_i, m_j) in d means

that features m'_i and m_j appear together in review d . The function $\text{normal}()$ is a simple normalization function that is used to ensure that every element in the vector belongs to $[0, 1]$. This algorithm is an effective algorithm in the sense that it can find the features that are the most relevant to a specific feature. We can obtain matrix C using the above algorithm. After obtaining matrix C , we can use EPMI to build a new sentiment matrix.

Definition 3. Sentiment matrix F can be determined by (7). The only difference between sentiment matrices F and B is that matrix F is obtained using our approach (EPMI) rather than the traditional PMI. In other words, each cell of sentiment matrix F represents the EPMI value between the sentiment words and product features rather than the PMI value between them.

$$F[n][t] = B[n][t] + A[n][p] * C^T[t][p] \quad (7)$$

So far, we have obtained three sentiment matrices A , B , and F . The sentiment words can be represented by the vectors in each of the sentiment matrices.

In mobile shopping reviews, customers often use different sentiment words to modify different product features. In addition, customers may have a good feeling about some features of a product, but they may be dissatisfied with some other features at the same time. Therefore, different product features also reflect different feelings. We assume that sentiment words can be divided into different categories according to the relationship between them and the product features.

Therefore, we cluster the sentiment words into several categories rather than into binary or other fixed categories. In other words, the sentiment dimension of a word in our domain-specific lexicon is flexible rather than having only limited emotional polarity. For each sentiment dimension, each sentiment word can take the value of 0 or 1, where 1 indicates that it belongs to a particular category whereas 0 indicates that it does not belong to that category. If we cluster the sentiment words into five categories, the representation

TABLE 2: Sample product reviews.

1	早餐 很 丰盛, 餐厅 非常 大。 (Breakfast is very hearty, the dining room is very large)
2	食物 种类 丰富, 那里 有 免费 的 早餐。 (The food is plentiful and there is a free breakfast)
3	餐厅 免费 供应 丰盛 的 早餐。 (The dinning serves a hearty breakfast for free)
4	餐厅 里 的 食物 挺 丰富 (The food in the dinning is rich)
5	餐厅 食物 真 不错, 早餐 尝 起来 很 好吃 (The food in the dinning is really good, the breakfast tastes good)

TABLE 3: Number of instances of cooccurrence of features.

	m_1	m_2	m_3	m_4
m'_1	0	2	2	1
m'_2	2	0	3	1

$\vec{e} = [0, 0, 1, 0, 0]$ means that the word e belongs to the third category. The flexibility of sentiment dimensions is a main characteristic of the domain-specific lexicon built using our approach.

3.4. Walk-Through Example. Here, we will elaborate on the differences between EPMI and PMI using an example. Suppose that we want to determine the semantic correlations between the sentiment words $e_1 = \text{“丰富”}$ (rich) and $e_2 = \text{“丰盛”}$ (hearty), and the five sentences listed in Table 2 are our corpus. This small corpus is a part of Chinese mobile shopping reviews about hotels.

Here, N is 33 because these five comments contain 33 Chinese words. In this example, there are four nouns (or primitive product features), i.e., $M = \{m_1 \sim m_4 \mid \text{“食物” (food), “早餐” (breakfast), “餐厅” (dining room), “种类” (variety)}\}$. To keep our example simple and understandable, we focus on only two features, $m_1 = \text{“食物” (food)}$ and $m_2 = \text{“早餐” (breakfast)}$. Therefore, M' can be $\{m'_1, m'_2\}$. $\text{count}(e_1)$ is 2 because the word e_1 appears only twice in this small corpus. Similarly, $\text{count}(e_2)$ is 2, $\text{count}(m_1)$ is 3, and $\text{count}(m_2)$ is 4. The size of the local window of cooccurrence is set as 3. $\text{count}(e_1, m_1)$ is 2 because e_1 and m_1 cooccur twice in the second and fourth comments within the window. Similarly, $\text{count}(e_2, m_2)$ is 2, $\text{count}(e_1, m_2)$ is 0, and $\text{count}(e_2, m_1)$ is 0. Using (1), we can obtain $\text{PMI}(e_1, m_1) = \log_2(p(e_1, m_1)/p(e_1)p(m_1)) = (2/33)/((2/33)(3/33)) = 3.46$ and $\text{PMI}(e_1, m_2) = 0$. Similarly, $\text{PMI}(e_2, m_1) = 0$ and $\text{PMI}(e_2, m_2) = (2/33)/((2/33)(4/33)) = 3.04$. Therefore, the two sentiment words e_1 and e_2 can be represented as $\vec{e}_1 = [3.46, 0]$ and $\vec{e}_2 = [0, 3.04]$, respectively, in sentiment matrices A and B .

We calculate u_{ij} using Algorithm 1. First, by iterating through these five comments, we obtain the number of instances of cooccurrence of m'_i and m_j (Table 3). In this table, each cell shows the number of times that two features appear

together in the same comment. The values in this table are similar to those in matrix C obtained in Algorithm 1.

Next, we normalize this table or matrix. We choose the min-max normalization function as the normal() function in the algorithm. Finally, we obtain $u_{12} = 1$, $u_{13} = 1$, $u_{14} = 1/2$, $u_{21} = 2/3$, $u_{23} = 1$, and $u_{24} = 1/3$. Note that u_{11} and u_{22} are 0.

Using (6), we obtain $\text{EPMI}(e_1, m_2) = \text{PMI}(e_1, m_2) + u_{21}\text{PMI}(e_1, m_1) + u_{22}\text{PMI}(e_1, m_2) + u_{23}\text{PMI}(e_1, m_3) + u_{24}\text{PMI}(e_1, m_4) = 3.65$ and $\text{EPMI}(e_1, m_1) = 5.48$. Similarly, $\text{EPMI}(e_2, m_1) = 6.08$ and $\text{EPMI}(e_2, m_2) = 6.08$. The two sentiment words e_1 and e_2 can be represented as $\vec{e}_1' = [5.48, 3.65]$ and $\vec{e}_2' = [6.08, 6.08]$ in sentiment matrix F .

It is obvious that the distance between $\vec{e}_1'$ and $\vec{e}_2'$ is considerably closer than that between $\vec{e}_1$ and $\vec{e}_2$, irrespective of the Euclidean or cosine distance. This result reflects the difference between our EPMI algorithm and PMI. These two sentiment words are very similar and are commonly used in hotel reviews. The greater the similarity between the two words, the shorter the distance between them. In the clustering model, the vectors that are located at shorter distances are easier to be clustered into the same category.

4. Experiments

To evaluate the domain-specific lexicon developed using our approach, we design an experimental setup using which we compare the proposed domain-specific lexicon with two popular general lexicons and with state-of-the-art machine-learning and deep-learning approaches that do not use a lexicon. We mainly evaluate different lexicons and approaches using document-level classification tasks in the domain of online product reviews. For hybrid sentiment classification methods, we consider the features of the document vector representation as the lexicon. We use the F1-measure as our main evaluation index and choose NB and SVM as the classifiers. In the following subsections, the details of the experiments and their results are described.

4.1. Dataset. The dataset includes both Chinese and English shopping reviews. These reviews are for seven types of products. The detailed statistics of this dataset are listed in Table 4.

The Chinese product reviews include three domains: hotel, cloth, and fruit. The hotel reviews are provided by Dr. Tan (<http://download.csdn.net/download/lssc4205/9903298>), and the cloth and fruit reviews are crawled from a mobile shopping application JD (<https://www.jd.com/>). The English reviews are obtained from the famous Amazon product review dataset collected by Blizter et al. [35]. It is widely used as a benchmark dataset for cross-domain sentiment classification. Four domains—book, DVD, electronics, and kitchen—are included in this dataset. For each domain, 1000 positive and 1000 negative reviews are included.

4.2. Experimental Design. We use the open-source software jieba (<https://pypi.python.org/pypi/jieba/>) to carry out pre-processing tasks on the Chinese product reviews, including Chinese word segmentation and POS tagging. For the

TABLE 4: Statistics of dataset.

Polarity	Chinese				English		
	Hotel	Cloth	Fruit	Books	Kitchen	DVD	Electronics
Positive	5321	5000	5000	1000	1000	1000	1000
Negative	2444	5000	5000	1000	1000	1000	1000

sentiment classification approaches that do not use a lexicon, we compare our method with the classical bag-of-words and deep-learning model Word2Vec [36]. Furthermore, we compare the domain-specific sentiment lexicon with two popular general sentiment lexicons. We use the scikit-learn [37] python library implementation of the classifier. The detailed differences between the three test groups are described below.

(a) No Lexicon

(BOW) The classical method to express the document involves the use of the bag-of-words model [2]. Each document d is represented by a feature-presence vector $\vec{d} = [0, 1, \dots, 0]$.

(W2V) In addition to the bag-of-words classical representation, we use the encoding of words provided by Word2Vec, which is a deep-learning tool released by Google in 2013. This tool adopts two main model architectures—the continuous bag-of-words model and continuous skip-gram model—to learn the vector representations of words [38]. To use Word2Vec for document-level tasks, a method is required that can unify all word vectors and generate a single vector representing the entire document [11]. Thus, the final representation is obtained according to the number of words contained in the document as follows:

$$\vec{v}_d = \frac{\sum_{i=0}^n \vec{v}_i}{n} \quad (8)$$

We use the genism (<https://radimrehurek.com/gensim/models/word2vec.html>) python library implementation of Word2Vec. We use the default values for almost all the parameters and use vectors with 200 dimensions.

(b) General Lexicon

For this test group, we use the hybrid sentiment classification approaches. That is, we consider the words in the lexicon, the sentiment dimensions of the lexicon, and a combination of the words and dimensions as the features for the machine-learning classifier. First, we choose a general sentiment lexicon DUTIR [39] for Chinese reviews. The DUTIR lexicon contains 27446 common Chinese words. The sentiment polarity of these words is labeled as positive, negative, or neutral.

(DUTIR) We consider only the words contained in the DUTIR lexicon as features, as in the case of the bag-of-words model. Therefore,

the review d can be represented as $\vec{d}_0 = [0, 1, \dots, 1]$.

(Only 3) We consider the three polarities of the sentiment words in the DUTIR lexicon. We represent the product review d by a three-dimensional vector $\vec{d}_1 = [k_0, k_1, k_2]$, where k_0 , k_1 , and k_2 are the number of words with the three types of polarities in the review.

(DUTIR+3) Here, we combine the above two representations. The product review d can be represented as $\vec{d} = \vec{d}_0 + \vec{d}_1 = [0, 1, \dots, 1, k_0, k_1, k_2]$.

For the English reviews, we choose a general sentiment lexicon SentiWordNet (<http://sentiwordnet.isti.cnr.it/>). SentiWordNet assigns three sentiment scores to each synset of WordNet: positivity, negativity, and objectivity. In other words, both the sentiment dimensions of the words for these two general sentiment lexicons in Chinese and English are 3.

(SentiWordNet) As in the case of the DUTIR lexicon, we focus on the words contained in the SentiWordNet lexicon. The review d can be represented as $\vec{d}_0 = [0, 1, \dots, 1]$.

(Only 3) We represent the product review d by a three-dimensional vector $\vec{d}_1 = [k_0, k_1, k_2]$. Here, k_0 is the sum of the positivity scores of the words in the review d . Similarly, k_1 and k_2 represent the sum of the negativity and objectivity scores, respectively.

(SentiWordNet+3) As in the case of the DUTIR lexicon, the product review d can be represented as $\vec{d} = \vec{d}_0 + \vec{d}_1 = [0, 1, \dots, 1, k_0, k_1, k_2]$.

(c) Domain-Specific Lexicon

We use hybrid approaches to evaluate the domain-specific lexicon developed using our method. We set the window size as 3 and α as 0.01 (as mentioned in Section 3.2). We use k -means (<http://scikit-learn.org/stable/modules/clustering.html#k-means>) to cluster the sentiment words into k categories based on the distance in the matrix space. Unlike the general lexicons, the sentiment dimension of words in our domain-specific lexicon is k . Note that we select k through a fivefold cross-validation on the training set. The details of the selection of k are explained in the next subsection. In our experiment, the number

of clusters is not more than 30. In the following discussion, the domain-specific lexicon built using our method is denoted as DS.

(DS) We consider the sentiment words contained in the domain-specific lexicon as features, as in the case of the bag-of-words model. Accordingly, the review d can be represented as $\vec{d}_0 = [0, 1, \dots, 1]$.

(Only k) We cluster the sentiment words into k categories using k -means. We represent the product review d by a k ($k \geq 2$)-dimensional vector $\vec{d}_1 = [m_0, m_1, \dots, m_{k-1}]$. Obviously, m_t represents the number of sentiment words that belong to the $(t + 1)$ th category in the review.

(DS+ k) Here, we combine the above two representations. The product review d can be represented as $\vec{d} = \vec{d}_0 + \vec{d}_1 = [0, 1, \dots, 1, m_0, m_1, \dots, m_{k-1}]$.

Three different sentiment matrices are considered in our lexicon construction process. The sentiment word representations in different matrices are very different. Therefore, the clustering result would also be different. We use $k(\text{PMI})$, $k(\text{TFIDF-PMI})$, and $k(\text{EPMI})$ to represent the clustering results of sentiment matrices A , B , and F , respectively. We discuss the different results of the three matrices in the next subsection.

In addition, we evaluate the lexicon in terms of the coverage and usage. We assume that the test set contains N unique words and that these N words include T sentiment words, which are contained in the sentiment lexicon. We also assume that the size of the lexicon that is used to train the classification model is S . Therefore, the coverage of the lexicon is T/N , and the usage of lexicon is T/S . If the coverage of the lexicon is low, the classification performance will be unsatisfactory. If the usage of the lexicon is low, computing resources will be wasted, which should be avoided especially for mobile devices. Considering these two evaluation indexes, we propose an average evaluation index such as the F1-measure. Let V , U , and T represent the coverage, usage, and average of the lexicon, respectively. Then, T is defined as

$$T = \frac{2 * V * U}{V + U} * 100\% \quad (9)$$

4.3. Results and Discussion

Overall Results. Table 5 lists the overall classification results. All the tasks are balanced two-class problems. The best result for each domain review is marked in bold font, and the second-best result is underlined.

First, for the domain-specific and general lexicons, DS+ k achieves the best results for all the seven domain reviews whereas DS achieves the second-best results for four of the reviews. DS outperforms the general lexicons DUTIR

FIGURE 2: Time required for clustering in sentiment matrix built using three methods. The three different lines represent three different sentiment matrices.

and SentiWordNet. These results indicate that the domain-specific lexicon, which is constructed from the corresponding corpus, shows better performance for sentiment classification tasks on shopping reviews.

Second, for no lexicon approaches, the classical bag-of-words model obviously performs better than the deep-learning model Word2Vec in terms of sentiment classification tasks. BOW achieves the second-best results for three of the reviews, whereas W2V shows nearly the worst performance for both Chinese and English reviews. The poor performance is unexpected, and a large corpus of training data is perhaps required for training Word2Vec [40].

Third, for sentiment dimensions, the performances of Only 3 and Only k are relatively worse for both Chinese and English reviews. That is, it is not sufficient to just consider the sentiment dimensions when we use the lexicon as the source of the features to express the reviews. However, the performance of DS+ k is better than those of DS and Only k for both Chinese and English reviews. This result indicates the effectiveness of combining the words and sentiment dimensions of the lexicon.

Note that, in Table 5, k represents $k(\text{EPMI})$. Considering that (DS+ k) provides the best performance, we analyze the differences among the results of DS+ $k(\text{PMI})$, DS+ $k(\text{TFIDF-PMI})$, and DS+ $k(\text{EPMI})$ in detail.

EPMI versus PMI and TFIDF-PMI. Table 6 lists the classification results of (DS+ k) with the three different methods mentioned in Section 3. First, we find that the classification performance of DS+ $k(\text{EPMI})$ is better than those of DS+ $k(\text{PMI})$ and DS+ $k(\text{TFIDF-PMI})$. In particular, the performance of DS+ $k(\text{PMI})$ is relatively poor. According to a t -test, the differences among the results of the three methods are significant ($p < 0.05$). This result reflects the advantages of EPMI over traditional PMI in sentiment classification.

Second, we discuss the differences among the three methods in terms of time efficiency. Figure 2 shows the average clustering times required by the three different sentiment

TABLE 5: F1-measure classification results for shopping domain reviews (NB).

Method	Chinese			English			
	Hotel	Cloth	Fruit	Books	Kitchen	DVD	Electronics
No lexicon							
BOW	<u>81.9</u>	<u>86.4</u>	<u>88.8</u>	74.0	79.3	77.6	77.0
W2V	61.2	68.2	69.8	61.5	67.1	63.8	61.7
General lexicon							
DUTIR	71.5	75.0	73.8	-	-	-	-
SentiWordNet	-	-	-	72.3	79.0	76.8	76.5
Only 3	68.8	70.7	70.2	65.1	62.3	68.4	61.9
DUTIR+3	72.6	74.8	74.2	-	-	-	-
SentiWordNet+3	-	-	-	74.9	79.8	78.5	77.9
Domain-specific lexicon							
DS	80.3	84.9	86.8	<u>77.7</u>	<u>81.1</u>	<u>79.5</u>	<u>79.0</u>
Only k	70.3	71.2	71.5	61.6	64.9	62.7	63.2
DS+ k	83.6	87.6	89.5	78.9	82.5	80.6	80.5

TABLE 6: Classification results of DS+ k with three different methods.

DS + k	Chinese			English			
	Hotel	Cloth	Fruit	Books	Kitchen	DVD	Electronics
k (PMI)	81.0	85.2	87.3	77.3	81.5	79.0	79.2
k (TFIDF-PMI)	82.8	87.1	89.0	78.4	82.1	80.1	79.9
k (EPMI)	83.6	87.6	89.5	78.9	82.5	80.6	80.5

matrices. The time consumed by sentiment matrices B and F is considerably less than that consumed by sentiment matrix A . This is because in matrices B and F , the dimension of the vector in the matrix space is reduced by using the method described in Section 3.2. The dimension reduction leads to a substantial increase in the efficiency and accuracy of classification. Therefore, sentiment matrix F constructed using EPMI shows the best performance.

Selection of K . The sentiment dimensions of the domain-specific lexicon is k . Now, we analyze the influence of different k values on the classification performance. Figure 3 shows the performance of Only k (EPMI) with the change in k for the English product reviews. When k is 2, Only k (EPMI) shows the best performance for the books and DVD domains. For the kitchen and electronics domains, a larger k improves the classification performance of Only k (EPMI). The appropriate value of k for the domain-specific lexicon is different for different fields. We select the value of k through a fivefold cross-validation on the training set in our experiments.

However, we find that the performance of Only k (EPMI) is worse than that of Only 3 in the books and DVD domains (Table 5). We believe that this is because $k = 2$ is not a good choice for Only k (EPMI) for performing sentiment classification tasks. To prove this point, we look at the performance results of Only k ($k = 2$) for all English product reviews (Table 7).

Table 7 indicates that the performance of Only k (EPMI) is not good when k is fixed at 2. In our domain-specific lexicon, low sentiment dimensions such as $k = 2$ is not good for DS. k has a substantial influence on our domain-specific lexicon

FIGURE 3: Performance of Only k (EPMI) with change in k .TABLE 7: Performance results of Only k (EPMI). k is selected using two different methods.

	Selection of k	Books	Kitchen	DVD	Electronics
Only k	$k = 2$	61.1	60.3	62.1	53.6
	Cross-validation	61.6	64.9	62.7	63.2

for sentiment classification tasks. Therefore, it is necessary to select the k value by cross-validation.

TABLE 8: F1-measure classification results for shopping domain reviews (SVM).

Methods	Chinese			English			
	Hotel	Cloth	Fruit	Books	Kitchen	DVD	Electronics
No lexicon							
BOW	73.5↓	76.1↓	82.1↓	75.8↑	79.4	78.0↑	77.2
General lexicon							
DUTIR+3	79.5↑	82.1↑	82.5↑	-	-	-	-
SentiWordNet+3	-	-	-	74.1↓	76.2↓	74.9↓	76.3↓
Domain-specific lexicon							
DS+k	82.3↓	84.2↓	86.8↓	79.1	80.9↓	79.2↓	78.3↓

TABLE 9: Coverage, usage, and average of lexicons for Chinese reviews.

	BOW			DUTIR			DS		
	V	U	T	V	U	T	V	U	T
Hotel	71.0	34.1	46.1	7.7	2.5	3.8	24.5	39.6	30.2
Cloth	73.4	36.9	79.1	8.2	2.7	4.1	29.0	40.2	33.7
Fruit	75.0	37.5	50.0	8.0	3.3	4.0	28.9	39.9	31.9

TABLE 10: Coverage, usage, and average of lexicons for English reviews.

	BOW			SentiWordNet			DS		
	V	U	T	V	U	T	V	U	T
Books	62.3	24.1	34.7	51.6	5.2	9.4	37.9	32.5	34.9
Kitchen	67.1	28.2	39.7	54.8	2.7	5.1	42.8	39.5	41.1
DVD	60.3	24.2	34.5	46.6	5.0	9.0	34.2	33.2	33.6
Electronics	63.3	25.9	36.7	49.6	2.8	5.2	38.5	35.1	36.7

NB versus SVM. For obtaining all the above results, we have chosen NB as the classifier. However, classification algorithms influence the classification performance. Therefore, we choose another popular classification algorithm SVM as the classifier for the method with the best performance among each type of methods. The results are listed in Table 8, where ↑ indicates an improvement in performance compared to that when NB is used and ↓ indicates a deterioration in performance. SVM performs better than NB when using DUTIR+3 for Chinese reviews and when using BOW for the books and DVD domains. In contrast, NB yields better performance when using the other approaches. Sentiment classification is perhaps one of the domains that have clear feature dependence, and hence, NB often performs unexpectedly well [41]. Although the domain-specific lexicon performs better with both NB and SVM, different types of models of text classification are probably required for documents with different properties. Hence, further empirical and theoretical study is required to understand the relationship between sentiment classification tasks and classification models.

Lexicon Coverage. Finally, we discuss the classification performance in terms of the coverage (V), usage (U), and average (T). The results for the test set are listed in Tables 9 and 10. In both the Chinese and English domains, the average of BOW is relatively high. For Chinese product reviews, both the coverage and usage of DUTIR are the worst because DUTIR is a general lexicon that contains only few words that often

appear in shopping reviews. The coverage of SentiWordNet is considerably higher than that of DUTIR. This partly explains why the performance of SentiWordNet is better than that of DUTIR for sentiment classification tasks. The better performance is also probably because SentiWordNet contains more words related to mobile shopping reviews than DUTIR. The coverage of SentiWordNet is higher than that of the domain-specific lexicon for English product reviews, whereas the usage of SentiWordNet is considerably low than that of DS. The very low usage of lexicons may impact their performance and waste the computing resources of mobile devices. The average of DS is considerably higher than that of the general lexicon for both Chinese and English product reviews. This result reflects the advantage of our domain-specific lexicon for mobile shopping reviews in another way.

5. Conclusions

The analysis of the sentiment of users' product reviews largely depends on the quality of sentiment lexicons. This paper presents a sentiment lexicon construction approach for mobile shopping. In this approach, a sentiment matrix that considers the relationship between sentiment words and product features is built. The sentiment words are clustered based on the distance between them in the matrix space. One characteristic of our lexicon is that the sentiment words are clustered into several categories rather than into binary or other fixed categories. In other words, the sentiment

dimension of the words in our lexicon is flexible. In addition, the product features are filtered based on the idea of TFIDF. Moreover, the EPMI algorithm is proposed, which is more appropriate for the mobile review domain. The experimental results show that our sentiment lexicons outperform the benchmarks with statistically significant differences in terms of sentiment classification tasks, thus proving the effectiveness of the proposed approach.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work described in this paper was partially supported by the National Key R&D Program of China (Grant no. 2018YFC0407901), partially supported by the National Natural Science Foundation of China under Grants no. 61370091 and no. 61602149, partially supported by the Fundamental Research Funds for the Central Universities under Grant no. 2016B01714, and partially supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions. Lau's work is supported by grants from the Research Grants Council of the Hong Kong SAR (Projects CityU 11502115 and CityU 11525716) and the NSFC Basic Research Program (Project 71671155).

References

- [1] A. Montoyo, P. Martínez-Barco, and A. Balahur, "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments," *Decision Support Systems*, vol. 53, no. 4, pp. 675–679, 2012.
- [2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing—Volume 10 (EMNLP '02)*, pp. 79–86, Association for Computational Linguistics, Stroudsburg, Pa, USA, July 2002.
- [3] P. D. Turney, "Thumbs up or thumbs down?" in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417–424, Philadelphia, Pennsylvania, July 2002.
- [4] A. Ortigosa, J. M. Martín, and R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning," *Computers in Human Behavior*, vol. 31, no. 1, pp. 527–541, 2014.
- [5] B. Liu, "Sentiment analysis and opinion mining," *Morgan & Claypool*, 2012.
- [6] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pp. 168–177, August 2004.
- [7] Y.-J. Tai and H.-Y. Kao, "Automatic domain-specific sentiment lexicon generation with label propagation," in *Proceedings of the International Conference on Information Integration and Web-Based Applications Services*, pp. 53–62, 2013.
- [8] D. Maynard and A. Funk, "Automatic detection of political opinions in tweets," in *Proceedings of the 1st Workshop on Making Sense of Microposts 2011: Big Things Come in Small Packages, MSM 2011 - Co-located with the 8th Extended Semantic Web Conference, ESWC 2011*, pp. 81–92, Greece, May 2011.
- [9] Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, "Sentiment classification of internet restaurant reviews written in cantonese," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7674–7682, 2011.
- [10] X. Li, J. Cao, and Z. Pan, "Market impact analysis via deep learned architectures," *Neural Computing and Applications*, pp. 1–12, 2018.
- [11] F. Enríquez, J. A. Troyano, and T. López-Solaz, "An approach to the use of word embeddings in an opinion classification task," *Expert Systems with Applications*, vol. 66, pp. 1–6, 2016.
- [12] Z. Hailong, G. Wenyan, and J. Bo, "Machine learning and lexicon based methods for sentiment classification: A survey," in *Proceedings of the 11th Web Information System and Application Conference, WISA 2014*, pp. 262–265, China, September 2014.
- [13] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd International Conference on Knowledge Capture, K-CAP 2003*, pp. 70–77, USA, October 2003.
- [14] G. Qiu, X. He, F. Zhang, Y. Shi, J. Bu, and C. Chen, "DASA: Dissatisfaction-oriented Advertising based on Sentiment Analysis," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6182–6191, 2010.
- [15] S. Jadav, B. Tanawal, and H. Guadani, "Sentiment analysis: a review," *International Journal of Advance Engineering and Research Development*, vol. 4, pp. 957–962, 2017.
- [16] V. Sindhwani and P. Melville, "Document-word co-regularization for semi-supervised sentiment analysis," in *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM 2008*, pp. 1025–1030, Italy, December 2008.
- [17] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, "News impact on stock price return via sentiment analysis," *Knowledge-Based Systems*, vol. 69, no. 1, pp. 14–23, 2014.
- [18] G. Badaro, R. Baly, R. Akel et al., "A Light Lexicon-based Mobile Application for Sentiment Mining of Arabic Tweets," in *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pp. 18–25, Beijing, China, July 2015.
- [19] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using WordNet to measure semantic orientations of adjectives," in *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*, pp. 1115–1118, Portugal, May 2004.
- [20] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: a survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.
- [21] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL '97)*, pp. 174–181, 1997.
- [22] P. D. Turney and M. L. Littman, "Measuring praise and criticism: inference of semantic orientation from association," *ACM Transactions on Information and System Security*, vol. 21, no. 4, pp. 315–346, 2003.
- [23] A. Yang, J. Lin, Y. Zhou, and J. Chen, "Research on building a Chinese sentiment lexicon based on SO-PMI," *Applied Mechanics and Materials*, vol. 263–266, no. 1, pp. 1688–1693, 2013.

- [24] P. Ekman, "An Argument for Basic Emotions," *Cognition & Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
- [25] V. L. Rubin, J. M. Stanton, and E. D. Liddy, *Discerning Emotions in Texts*, Stanford University, 2004.
- [26] A. Tellegen, D. Watson, and L. A. Clark, "On the dimensional and hierarchical structure of affect," *Psychological Science*, vol. 10, no. 4, pp. 297–303, 1999.
- [27] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, "Sentiment Analysis Is a Big Suitcase," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 74–80, 2017.
- [28] E. Fast, B. Chen, and M. S. Bernstein, "Empath: Understanding topic signals in large-scale text," in *Proceedings of the 34th Annual Conference on Human Factors in Computing Systems, CHI 2016*, pp. 4647–4657, USA, May 2016.
- [29] H. Zhang, Z. Yu, M. Xu, and Y. Shi, "Feature-level sentiment analysis for Chinese product reviews," in *Proceedings of the 3rd International Conference on Computer Research and Development (ICCRD)*, pp. 135–140, Shanghai, China, March 2011.
- [30] M. Gilad, "Experiments with mood classification in blog posts," *ACM Transactions on Multimedia Computing Communications and Applications*, 2005.
- [31] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V. S. Subrahmanian, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone," in *Proceedings of the 2007 International Conference on Weblogs and Social Media, ICWSM 2007*, USA, March 2007.
- [32] Y. Lu, X. Kong, X. Quan, W. Liu, and Y. Xu, "Exploring the Sentiment Strength of User Reviews," in *Web-Age Information Management*, vol. 6184 of *Lecture Notes in Computer Science*, pp. 471–482, Springer, Berlin, Germany, 2010.
- [33] T. Wilson, P. Hoffmann, S. Somasundaran et al., "Opinion-Finder," in *Proceedings of the HLT/EMNLP*, pp. 34–35, Vancouver, British Columbia, Canada, October 2005.
- [34] C. Hou, F. Nie, D. Yi, and D. Tao, "Discriminative embedded clustering: a framework for grouping high-dimensional data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1287–1299, 2015.
- [35] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 440–447, June 2007.
- [36] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS '13)*, pp. 3111–3119, December 2013.
- [37] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [38] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1857–1863, 2015.
- [39] H. Lin, L. Xu, H. Ren et al., "Constructing the affective lexicon ontology," *Journal of The China Society for Science and Technical Information*, vol. 27, pp. 180–185, 2008.
- [40] A. Edgar, R. Sidarta, S. Mariano, and F. S. Diego, "Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database," in *ASAI Simposio Argentino de Inteligencia Artificial*, 2016.
- [41] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple bayesian classifier," *Machine Learning*, vol. 29, pp. 103–130, 1996.