J SYST SCI SYST ENG
Vol. 28, No. 6, December 2019, pp. 731–746
DOI: https://doi.org/10.1007/s11518-019-5438-4
ISSN: 1004-3756 (paper), 1861-9576 (online)
CN 11-2983/N

DCWord: A Novel Deep Learning Approach to Deceptive Review Identification by Word Vectors

Wen Zhang,[a] Qiang Wang,[a] Xiangjun Li,[b] Taketoshi Yoshida,[c] Jian Li[a]

[a]Research Base of Beijing Modern Manufacturing Development, College of Economics and Management, Beijing University of Technology, Beijing 100124, China
zhangwen@bjut.edu.cn (✉), wangqiang@emails.bjut.edu.cn, lianjiansem@bjut.edu.cn
[b]School of Information Engineering, Xi'an University, Xi'an 710065, China
leelindass@yahoo.com
[c]School of Knowledge Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan
yoshida@jaist.ac.jp

Abstract. Due to the anonymous and free-for-all characteristics of online forums, it is very hard for human beings to differentiate deceptive reviews from truthful reviews. This paper proposes a deep learning approach for text representation called DCWord (Deep Context representation by Word vectors) to deceptive review identification. The basic idea is that since deceptive reviews and truthful reviews are composed by writers without and with real experience on using the online purchased goods or services, there should be different contextual information of words between them. Unlike state-of-the-art techniques in seeking best linguistic features for representation, we use word vectors to characterize contextual information of words in deceptive and truthful reviews automatically. The average-pooling strategy (called DCWord-A) and max-pooling strategy (called DCWord-M) are used to produce review vectors from word vectors. Experimental results on the Spam dataset and the Deception dataset demonstrate that the DCWord-M representation with LR (Logistic Regression) produces the best performances and outperforms state-of-the-art techniques on deceptive review identification. Moreover, the DCWord-M strategy outperforms the DCWord-A strategy in review representation for deceptive review identification. The outcome of this study provides potential implications for online review management and business intelligence of deceptive review identification.

Keywords: Online business intelligence, skip-gram model, DCWord representation, deceptive review identification, deep learning

1. Introduction

With the development of Web 2.0 and E-commerce, business stakeholders are accepting that online opinions are valuable and indispensable for its success in winning good reputation in market (Mudambi and Schuff 2010, Chatterjeei 2001). Online reviews refer to users' opinions on products or services posted in the Internet forums such as Dianping[1] , Koubei[2] and TripAdvisor[3] . These reviews are often used by customers in purchasing decision making or E-commerce merchants in online promotion campaign. Due to word-of-mouth effect (Kietzmann and Canhoto 2013), positive online reviews are profitable in earning good reputation for the online retailers while negative online reviews will hurt their reputations. For this reason, deceptive reviews are pervasive in Internet to mislead online shoppers (Chen and Wang 2013, Marrese-Taylor et al. 2013).

On the one hand, some merchants strive to produce and collect positive online reviews for themselves and defame their competitors with negative online reviews, even by hiring "water army" to post online reviews (Liu 2012). On the other hand, it is impossible for human beings to differentiate deceptive reviews from truthful reviews to a satisfactory extent (Ott et al. 2011). To be worse, anyone can post online re-

views anonymously in the Internet with a little cost and these reviews might bring about great commercial winning for themselves or great loss for their competitors. This fact causes massive spam, misleading and even fraudulent online reviews on products and merchants.

However, for the sake of time and expense saving, more and more customers are turning to online shopping (Lim et al. 2016) and most of them are making use of online reviews in their purchasing decision making, especially for those novices who have no experience on the products in consideration. As the widespread of deceptive online reviews, there are an increasing number of customers who are anxiously worrying about being cheated even trapped by these reviews in online shopping. This outcome leads deceptive review identification becoming a crucial concern in research field (Ott et al. 2011, Jindal and Liu 2008, Gokhman et al. 2012, Li et al. 2014, Feng et al. 2012, Feng and Hirst 2013, Zhou et al. 2008, Li et al. 2011).

To automatically identify spam reviews, Ott et al. (2008) use LIWC2007 (Linguistic Inquiry and Word Count) software (Li et al. 2014) to produce psychological features from reviews and combine these features with N-grams to automatically categorize deceptive reviews using support vector machine (SVM). Their experiments on the Spam dataset report that the combination of psychological features and N-grams can produce accuracy as 90% in spam categorization and is much better than human judges with accuracy as 60%. They observe that truthful opinions tend to include more sensorial and concrete language than deceptive opinions and are more specific about spatial configurations. Feng et al. (2012) combine part-of-speech (POS) features and Probabilistic Context Free Grammar (PCFG) features to improve deceptive review identification. Their experiments on four datasets as the Spam dataset, the Yelp dataset, the Heuristic TripAdvisor dataset and the Essays dataset show that the PCFG feature can significantly improve performances of deceptive opinion detection compared with words and POS (part-of-speech) features.

Feng and Hirst (2013) propose the idea of profile compatibility to identify deceptive reviews. First, the general aspects and distinct aspects of the reviewed target objective as well as their descriptive words are extracted from the reviews. Second, for an incoming new review, it is aligned with the constructed profiles of the target objective to decide its compatibility. Third, they combine the compatibility features, N-grams and PCFG rules together and use them as the input of SVM classifier for deceptive review identification. The experiments on the Spam dataset show that the compatibility feature can improve the performances of deceptive review identification significantly. Similar work in automatic deceptive review identification can be found in Zhou et al. (2008), Li et al. (2011), Ren and Zhang (2016) and Li et al. (2011)

Although the above mentioned studies have already conducted extensive investigations on deceptive review identification, it still needs more research. First, all the features used for deceptive review identification are derived by traditional natural language processing (NLP) such as N-grams, POS features and PCFG features. That is to say, the performances of deceptive review identification is highly dependent on the outcome of those NLP tools such Stanford parser (Klein and Manning 2003) and Q-tagger[4] . However, those NLP tools are trained mostly on news corpora which are entirely different from the online reviews. This fact makes those NLP tools not necessarily suitable to extract lexical features from online reviews. Second, the contextual information of words in reviews is ignored in existing studies. Although N-grams are usually used to capture contextual information of words in traditional

NLP, the lengths of N-grams are often set as less than 3 and the order of words in an N-gram is fixed and that will make N-gram incapable of capturing word contexts in a flexible manner. For instance, "hotel Hilton" and "Hilton hotel" have the same meaning but N-grams treat them differently because the two words in them are of different orders.

Motivated by the above observation, this paper proposes an approach called DCWord to represent user reviews as numerical vectors by deep learning. The basic idea is that there should be different contextual information for words in deceptive reviews and truthful reviews. Thus, by learning words with its deceptive and truthful contexts, we embed each word into two numeric vectors as deceptive vector and truthful vector. Further, for an incoming new review, we represent it by using the derived deceptive vectors and truthful vectors in contrast based on word occurrences in its textual contents. We propose two strategies as the average pooling strategy (DCWord-A) and the maximum pooling strategy (DCWord-M) to transfer vectors of words in a review into a review vector. Finally, the DCWord vectors of reviews are used to train the machine learning classifiers as SVM, LR and BPNN (back propagation neural network) to identify the deceptive reviews. To the best of our knowledge, this paper is the first to introduce word vectors in deep learning to deceptive review identification.

The remaining of the paper is organized as follows. Section 2 presents the word vector representation. Section 3 proposes the DCWord representation for reviews. Section 4 conducts the experiments and Section 5 concludes the paper.

2. Word Vector Representation

Using numerical vectors to represent words in corpus is becoming an attractive theme in machine learning and NLP areas (Collobert and Weston 2008). Different from traditional methods such as TF-IDF and LSI, where each word can be regarded as with a representation vector of a fixed length of the documents in the corpus, word vectors are produced by learning from its contextual words using neural network architecture and its lengths are variable according to specific application. More importantly, unlike traditional methods that more often than not suffer from the curse of dimensionality, the size of a word vector is much smaller than the number of documents in the dataset. In this way, the words in texts are embedded in dense numerical vectors and similar words are represented with similar contexts mathematically.

In this paper, we regard the surrounding words of a word as its context for deceptive review identification. We are motivated by that a word is characterized by the company it keeps (Firth 1957) and, not only the individual word but also the contextual information of the individual word is useful for further information processing (Zhang et al. 2009). Thus, we use word vector to capture the context of a word w_t . Usually, the word vectors can be learned by using either continuous Bag-of-Words (CBOW) model or continuous Skip-gram model (Mikolov et al. 2013). For simplicity and computation efficiency, the Skip-gram model is the state-of-the-art model adopted by many tasks in natural language processing. For brevity, the Skip-gram model can be depicted in **Figure 1**. Here, we use the word $v(w_t)$ to predict its 5 maximum distance words $(v(w_{t-2}), v(w_{t-1}), v(w_{t+1}), v(w_{t+2}))$. Note that $v(w_{t-2}), \cdots, v(w_{t+2})$ are all continuous numerical vectors and projection can be regarded as a neural work with one layer of hidden units.

The goal of the model training is to tune the word vectors that can be used to predict the surrounding words in a sentence. That is, given a sequence of training words $w_1, w_2, \cdots, w_t$, the Skip-gram needs to maxi-

mize the average log probability as described in Equation (1).

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c}\log p(w_{t+j}|w_t) \quad (1)$$

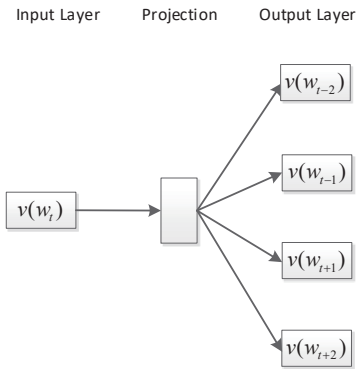


Figure 1 The Skip-Gram Model

Here, c is the maximum distance from the word w_t. The larger is c, the more training examples are involved in the model leading to a higher accuracy of the model but with more computation complexity. The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function as shown in Equation (2).

$$p(w_{t+j}|w_t)=\frac{\exp(v'(w_{t+j})^{\intercal}v(w_t))}{\sum_{i=1}^{V}\exp(v'(w_i)^{\intercal}v(w_t))} \quad (2)$$

$v'(w_i)$ is the output word vector for the word w_i and V is the size of the vocabulary. In traditional Skip-gram model, there are two word vectors as input word vector $v(w_i)$ and output word vector $v'(w_i)$ for the word w_i . In practice, it is very difficult to maximize Equation 1 if V is huge because, at each run of gradient descent, it will involve updating $O(|V|)$ parameters. Thus, negative sampling (Mikolov et al. 2013) is employed to solve this problem to compute $p(w_{t+j}|w_t)$.

3. The Proposed Method-DCWord Representation

The framework of using word vectors to identify deceptive reviews automatically is shown in **Figure 2**. We can see that the proposed method can be divided into two phases as training phase and test phase. In the training phase, the deceptive word vectors and truthful word vectors are produced with the Skip-gram model by using deceptive reviews and truthful reviews, respectively. Then, each training review is represented by these deceptive and truthful word vectors (see Section 3.1 and Section 3.2) as a DCWord vector to train machine learning classifier. In the test phase, each test review is represented by the learned deceptive and truthful word vectors in the training phase as a DCWord vector to test the learned model. Note that the training reviews are used for two purposes in the proposed method. The one is to produce the deceptive word vectors and truthful word vectors by the Skip-gram model and the other is to train machine learning classifier. The test reviews are used merely to examine the learned classifiers.

3.1 Word Representation

Because the input of Skip-gram model is sequences of words, we partition each review into sentences using the sentence boundary determination algorithm proposed by Nitin et al. (2005). Moreover, we adopt the stop word list obtained from USPTO (United States Patent and Trademark Office) patent fulltext and image database[5] in training the Skip-gram model. For a word in the vocabulary, we use the words in the set $window(w_t, c)$, i.e. the neighboring words of the word within the maximum distance as window size c , as the input for the Skip-gram model to produce its neural word embedding $vec(w_t)$. Using this method, we train the Skip-gram model for words in deceptive reviews and truthful reviews respectively. As a result, for each word w_t, we derive its deceptive word vector $vec^{\mathrm{dec}}(w_t)$ and truthful word vectors $vec^{\mathrm{tru}}(w_t)$ under deceptive context and truthful context respectively.

Actually, we can divide words in reviews as two types as general words and specific words.

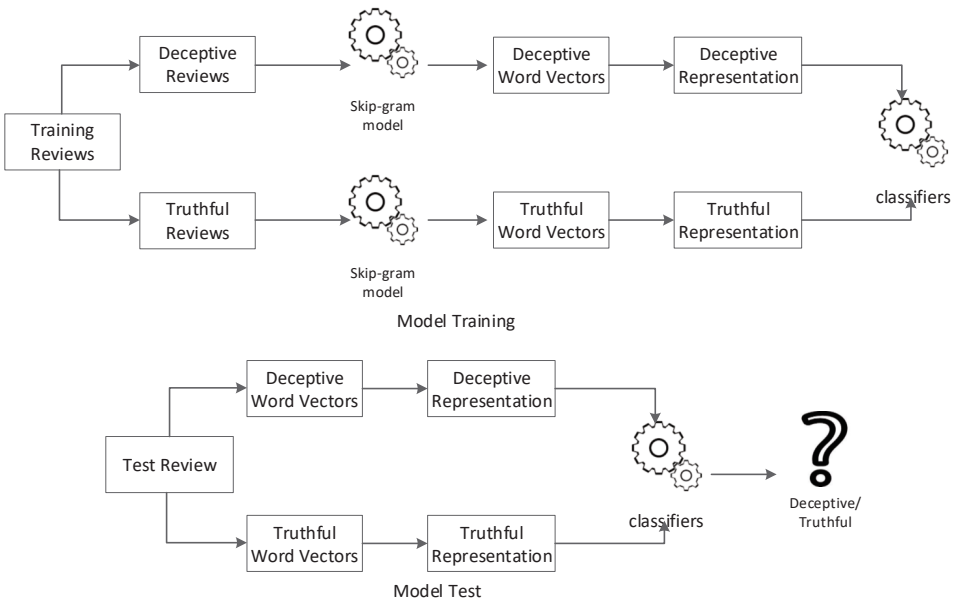


Figure 2 The Framework of Using DCWord for Deceptive Review Identification

Algorithm 1 The procedure to extract general words from reviews.

Input: Dwv-deceptive word vectors;
Twv-truthful word vectors;
N-predefined number of investigated neighbors given a target word;
M -the number of general words;
Output: G- the set of general words in reviews;
Procedure:

1: $CG = |Dwv \cap Twv|$;
2: $gList = \{\}$;
3: **for** each word w_t w_i in CG **do**
4: Retrieve the top N neighbors of w from Dwv to compose$dwList$;
5: Retrieve the top N neighbors of w from Twv to compose$twList$;
6: $ol(dwList, twList) = \frac{|dwList \cap twList|}{N}$
7: Add $(w_t, ol(dwList, twList))$ to $gList$;
8: End for
9: Sort $gList$ in descending order by $ol(dwList, twList)$ and retrieve the top–M words to G;

General words are about general concepts such as "room", "price" and "service" with respect to target objective "hotel". Specific words are of specific concepts such as "furniture", "lobby" and "restaurant" with respect to target objective "hotel". The general words are of less usefulness in identifying deceptive reviews because they have similar contextual information in both deceptive and truthful reviews. However, the specific words are more powerful than general words in distinguishing deceptive reviews from truthful reviews because the context information of specific words is dependent on reviewers' real experiences.

Algorithm 1 shows the procedure to extract general words from the reviews. The basic idea here is that for a given target word w, the larger overlapping is its top-N similar words in deceptive reviews and truthful reviews, the higher priority it would be regarded as a general word. Line 1 is used to retrieve the common words from the vocabulary of deceptive reviews and the vocabulary of truthful reviews. Lines 3 and 4 are used to extract its top- deceptive neighbors $dwList$ and truthful neighbors $twList$, respectively, when given a target word w. Line 6 is used to compute the overlapping ratio $ol(dwList, twList)$ of the top$N$ neighbors from deceptive reviews and truthful reviews. Line 7 is used to store the word and the overlapping ratio of its neighboring words in the deceptive reviews and truthful

reviews in to $gList$. Line 9 is used to retrieve the top-M words with largest overlapping ratios from $gList$.

3.2 Review Representation

After deriving the deceptive word vectors $vec^{\text{dec}(w_t)}$ and the truthful word vectors $vec^{\text{tru}}(w_t)$ as well as the general word set G, each review r_i is represented as the concatenation $vec(r_i) = (vec^{\text{dec}(r_i)}, vec^{\text{tru}}(r_i))$ where $vec^{\text{dec}(r_i)}$ is the deceptive representation for the review r_i and $vec^{\text{tru}}(r_i)$ is the truthful representation for the review r_i. In order to represent a review using the vectors of words in it, we propose two strategies as the average-pooling strategy (i.e. DCWord-A) and max-pooling strategy (i.e. DCWord-M).

Under the average-pooling strategy, the deceptive representation $vec^{\text{dec}}(r_i)$ is computed as $vec^{\text{dec}}(r_i) = \frac{1}{|W(r_i)-G|} \sum_{w \in W(r_i)-G} vec^{\text{dec}}(w_t)$ where $W(r_i)$ is the words contained in the review r_i. By analogy, we compute the truthful representation as $vec^{\text{tru}}(r_i) = \frac{1}{|W(r_i)-G|} \sum_{w \in W(r_i)-G} vec^{\text{tru}}(w_t)$. Note that if a word only appearing in deceptive re-views, then its truthful vector is simply regulated as a zero vector. In a word, when using DCWord-A representation, each review is represented as $vec(r_i) = (\frac{1}{|W(r_i)-G|} \sum_{w \in W(r_i)-G} vec^{\text{dec}}(w_t), \frac{1}{|W(r_i)-G|} \sum_{w \in W(r_i)-G} vec^{\text{tru}}(w_t)))$. **Algorithm 2** shows the details of the procedure to represent reviews by the DCWord-A method.

Under the max-pooling strategy, the deceptive representation $vec^{\text{dec}}(r_i)$ is computed as $= vec^{\text{dec}}(r_i) = \max_{w_t \in W(r_i)-G_j} vec^{\text{dec}}(w_t)$ where $W(r_i)$ is the words contained in the review r_i. The manipulation is an element-wise function to produce maximum elements on dimensions. That is, for all vectors of words in the set $W(r_i) - G$, the $\max_j$ will produce a vector from them by concatenating the maximum element of each dimension among them. In the same way, we compute the truthful representation as $vec^{\text{dec}}(r_i)$ is computed as $vec^{\text{tru}}(r_i) = \max_{w_t \in W(r_i)-G_j} vec^{\text{tru}}(w_t)$. If a word only appearing in deceptive (truthful) reviews, then its truthful (deceptive) vector is simply regulated as a zero vector. When using DCWord-M representation, each review r_i is represented as $vec(r_i) = (\max_{w_t \in W(r_i)-G_j} vec^{\text{dec}}(w_t), \max_{w_t \in W(r_i)-G_j} vec^{\text{tru}}(w_t))$. **Algorithm 3** shows the details of the procedure to represent reviews by the DCWord-M method.

Algorithm 2 The procedure for review representation by word vectors using the DCWord-A strategy.

Input: Dwv-deceptive word vectors;
Twv-truthful word vectors;
$W(r_i)$-the word list of review r_i ;
G- the set of general words in reviews;
Output: $vec(r_i)$-DCWord-A representation for review r_i ;
Procedure:

1: Initialize $vec(r_i)$ as a zero vector;
2: **for** each word w in $W(r_i) - G$ **do**
3: Retrieve the word vector $vec^{\text{dec}}(w_t)$ in Dwv ;
4: Retrieve the word vector $vec^{\text{tru}}(w_t)$ in Twv ;
5: $vec(r_i) = vec(r_i) + (vec^{\text{dec}}(w_t), vec^{\text{tru}}(w_t))$
6: End for
7: $vec(r_i) = \frac{1}{|W(r_i)-G|}$

3.3 Deceptive Review Identification

In constructing the learning models, we denote the label of deceptive reviews as 1 and the label of truth labels as –1. With SVM as the classifier, we use $vec(r_i)$ as the input vector and the output is the label given by the decision function $f(x) = \text{sgn}((w, x)+b) \in \{-1, 1\}$. The linear kernel as $(u * v)^1$ is used to train the classification model because it is proved superior to non-linear kernels in text categorization (Zhang et al. 2008). With LR as the classifier, we use $vec(r_i)$ as the input vector and the output is the probability $P(y_i = 1|vec(r_i))$, i.e., the prob-

Algorithm 3 The procedure for review representation by word vectors using the DCWord-M strategy.

Input: Dwv-deceptive word vectors;
Twv-truthful word vectors;
$W(r_i)$-the word list of review r_i ;
G- the set of general words in reviews;
Output: $vec(r_i)$-DCWord-M representation for review r_i ;
Procedure:

Initialize $vec(r_i)$ as a zero vector;
for each word w in $W(r_i) - G$ **do**
  Retrieve the word vector $vec^{\text{dec}}(w_t)$ in Dwv ;
  Retrieve the word vector $vec^{\text{tru}}(w_t)$ in Twv ;
  for For each dimension j of **do**
    if $vec(r_i)_j < (vec^{\text{dec}}(w), vec^{\text{tru}}(w))_j$ **then**
      $vec(r_i)_j = (vec^{\text{dec}}(w), vec^{\text{tru}}(w))_j$
    End If
  End for
End for

ability of the review r_i belonging to the "deceptive" category. If $P(y_i|vec(r_i))$ is greater than 0.5, then the review r_i is regarded as "deceptive". Otherwise, the review r_i is regarded as "truthful". With the BPNN as learning model, the size of the input layer is set as equal to the size of the DCWord vectors of reviews. The size of hidden nodes is set as 100 by trial and error. The size of the output layer is set as 2, i.e. a two-element vector to indicate the possibility of being "deceptive" or "truthful". When using the deceptive reviews for training, its output vector is set as (1, –1) and for truthful reviews, its output vector is set as (–1, 1). For a test review r_i , if we find the value of the first element is larger than the second, it will be labeled as "deceptive". Otherwise, the test review will be labeled as "truthful".

4. Theoretical analysis

Any text-based system requires some representation of texts, and the appropriate representation depends on the kind of task to be performed (Zhang et al. 2007). This makes text representation is crucial for the success of text mining task. There are two kinds of work involved in text representation as indexing and term weighting. Indexing is the job of assigning indexing terms (also called features in machine learning) to the texts. In this aspect, we should consider the semantic quality and statistical quality of the index terms (Guo et al. 2018). That is to say, to how much extent the index terms are representative of the meaning of the text and to how much extent the index terms can discriminate different texts (Cao and Tang 2014). Term weighting is the job of assigning weights to terms (or features), to measure the importance of terms in documents (Chen et al. 2018). That is to say, what is the importance of the indexing terms with respect to the texts?

With the above theory of text representation, this paper proposes word vectors by deep learning on truthful and deceptive reviews to represent reviews for the task of deceptive review identification. Then, the deceptive word vector $vec^{\text{dec}}(w_t)$ and the truthful word vector $vec^{\text{tru}}(w_t)$ are concatenated to represent the word w_t in the reviews. Actually, $vec^{\text{dec}}(w_t)$ contains the contextual information of the word w_t in deceptive reviews and $vec^{\text{tru}}(w_t)$ contains the contextual information of the word in truthful reviews. Thus, the review r_i is represented by composing the vectors $vec^{\text{dec}}(w_t)$ and $vec^{\text{tru}}(w_t)$ of words in it by either the average-pooling strategy or the max-pooling strategy.

In the point of view of text representation, the DCWord representation improves the state-of-the-art representation for deceptive review identification in at least two aspects. The first aspect is that it saves the indexing process in text presentation by automatic feature embedding. Unlike Ott et al. (2008) and Feng et al. (2012), they make use of N-gram and PCFG rules for explicit indexing terms. However, it is not necessary for the DCWord rep-

Table 1 The Basic Information of Reviews in the Spam Dataset

Polarity		# of Hotels	# of Reviews	# of Sentences
Positive	Deceptive_from_MTurk	20	400	3043
	Truthful_from_Web	20	400	3480
Negative	Deceptive_from_MTurk	20	400	4149
	Truthful_from_Web	20	400	4483

Table 2 The Basic Information of Reviews in the Deception Dataset

Subject	Category	# of Reviews	# of Sentences
doctor	deceptive_MTurk	356	2369
	truthful	200	1151
restaurant	deceptive_MTurk	201	1827
	truthful	200	1892

resentation to select indexing terms explicitly but only needs to set the number of features as the length of word vectors (i.e., L.V. in Section 4.3). The features are learned automatically from the truthful and deceptive reviews by automatic feature engineering of deep learning (Mikolov et al. 2013).

The second aspect is that saves the computation of term weighting in text presentation by using the elements of word vectors and pooling strategy for feature weighting. Most existing techniques make use of a subordinate procedure to compute the weighting of terms in texts such as TF-IDF (Term Frequency Inverse Document Frequency) and RIDF (Residual Inverse Document Frequency). However, it is not necessary for the DCWord representation to conduct term weighting because the feature weights are also learned by the Skip-gram model in producing the word vectors. In addition, considering that the identification of deceptive reviews is in fact a text classification task, we concatenate the deceptive representation and truthful representation of reviews to sharpen the difference of the features as well as their weighting with the goal of augmenting the discriminative power of the DCWord representation.

5. Experiments

5.1 The Dataset

The dataset used in the experiments is the Spam dataset from Ott et al. (2011). For each review, we conduct part-of speech analysis, stop-word elimination, and stemming and PCFG analysis. The part of speech of English word is determined by QTAG which is a probabilistic parts-of-speech tagger and can be downloaded freely online[6] . We use the same stop-words as used in training the Skip-gram model. The porter stemming algorithm is used to produce individual word stem[7] . For each review, we extract all of its sentences using the sentence boundary determination method described in (Nitin et al. 2005). The spam dataset contains 7,677 lexical terms (words) and 43,519 PCFG rules including 3,955 type 1 rules, 15,489 type 2 rules, 6,830 type 3 rules and 17,245 type 4 rules (Feng et al., 2012). The deception datasets contains 5,160 terms (words) and 43,511 PCFG rules, including 3,953 type 1 rules, 15,483 type 2 rules, 6,830 type 3 rules and 17,245 type 4 rules (Feng et al. 2012). **Table 1** and **Table 2** show the basic information of reviews in the spam dataset and the basic information of reviews in the deception dataset, respectively.

5.2 Experiment Setup

We divide each of the mentioned datasets into 10 folds uniformly using random sampling and

set the retaining level from 0.1 to 0.9 with interval as 0.1 for performance comparison. In the experiment setting, we only use 1 fold of reviews for test by random sampling when varying different number of folds for training. That is, if we set the retaining level as 0.6, then we partition the whole dataset as 10 folds and use 6 folds for training and 1 fold for test by random sampling. Then, under the average-pooling and max-pooling strategy, respectively, we repeat the experiments 10 times to average the performances. Note that for the method of compatibility profile (Feng and Hirst 2013), for each hotel, we extract 20 aspects for them with their modifiers as adjective words. That is to say, the baseline method of compatibility profile has extra 40 features by bi-alignment compared with traditional N-gram and PCFG features (Feng et al. 2012).

As mentioned in Section 3.1, there are three parameters needed to be set in training the Skip-gram model. The first one is the maximum distance from the word ω_t , i.e. c in Equation 1. Considering that the average length of the sentences in the reviews of both datasets is 15 after stop word elimination, we set the parameter c as 8 because, a smaller distance will make the word vector $w(t)$ incapable of capturing the contextual information of the word ω_t and a larger distance will introduce much irrelevance to the contextual information of the word ω_t.

The second parameter is the length of the word vector $w(t)$ derived from the Skip-gram model. Here, we set three different numbers as the layer sizes of the Skip-gram model as 100, 200, 300 and 400 for the Spam dataset and as 50, 100, 200 and 300 for the Deception dataset to produce different lengths of word vectors. For instance, if the layer size is set as 50, each word in reviews will be denoted as a numeric vector with length as 100 in DCWord representation (i.e., 50 from deceptive reviews and 50 from truthful reviews).

The third parameter is the number of iterations in the training process of the Skip-gram model. We tune this parameter and observe the updating of word similarities (cosine similarity) measured by the vectors at different settings. We find that for both datasets, when the training process runs 500 iterations, there is little updating of word similarities. For this reason, we set the number of iterations as 500 in training the Skip-gram model.

As for the general word extraction in Section 3.1, we set the top-N as 10 (N=10) words and the number of general words for each topic as 10 (M =10) by trial and error. That is to say, for a word w_t , if its number of common neighboring words is among the top-10 positions of all the words in $gList$ under the topic, then we regard the word w_t as a general word. We also ask human beings to check the appropriateness of the setting of top-N and M to make sure that all the general words in their mind are extracted from the reviews by the method. We admit that this method is a little arbitrary but to the best of our knowledge, there is no better way to set the parameters Top-N and M.

With the above method, we extract 10 words for each category of reviews as shown in **Table 3**. We can also extract more general words under each topic. However, on the one hand, it is of little help to improve the performance of deceptive review identification. On the other hand, other words than the mentioned 10 words in **Table 3** seem ambiguous between general and specific words such as the words "bath" and "floor" under the topic "hotel", the words "family" and "advice" under the topic "doctor" and the words "desert" and "salad" under the topic "restaurant". For this reason, we only use 10 general words under each topic.

Table 3 The Top-10 General Words Extracted from the Reviews by Algorithm 1

Topic	Hotel	Doctor	Restaurant
1	Hotel	patient	restaurant
2	Day	doctor	place
3	Night	surgery	food
4	Price	treatment	dinner
5	Service	health	service
6	Money	experience	staff
7	Room	staff	waiter
8	Place	clinic	experience
9	Stay	medicine	order
10	Desk	service	reservation

5.3 Results

Figure 3 shows the experimental results of deceptive review identification on the Spam dataset. We set different lengths of word vectors (L.V.) as 100, 200, 300 and 400, respectively, and different retaining levels (L.V.) from 0.1 to 0.9. We also adopt three different classifiers as SVM (Support Vector Machine), LR (Logistic Regression) and BPNN (Back Propagation Neural Network) to combine with the proposed DCWord approach to deceptive review identification.

We can see from **Figure 3** that first, when the L.V. parameter increases from 100 to 400, the performances of all methods are increasing. However, the performance improvements are narrowed when we compare the L.V. parameter increases between from 100 to 200 and from 300 to 400. This outcome can be explained that the when using more features to represent reviews, the performances of deceptive review identification will be also be improved. Nevertheless, when the reviews are represented with "enough" number of dimensions by the DCWord approach, the performances would not be further improved by giving more dimensions to word vectors.

When the L.V. parameter is small, all words are embedded in a space with a small number of dimensions. In this case, it seems that all words in reviews share similar contexts. This leads that it is hard to differentiate the deceptive reviews from truthful reviews by using word vectors. That is, there is very limited knowledge (or patterns) to be learned by the classifiers for deceptive review identification. Even if we add more training data by increasing the retaining level, those vectors are mixed with each other and it is of limited effectiveness to improve the performances of the classifiers. Nevertheless, when we increase the L.V. parameter, the words are embedded in a large space. In this case, words with similar contexts are projected to neighborhood coordinates while words with dissimilar contexts are projected to distant coordinates. Under this condition, it is not very hard to identify deceptive reviews and truthful reviews.

Second, we see that when the retaining level is increasing from 0.1 to 0.9, the performances of all methods are boosted up to its maxima and then gradually decrease after a critical threshold as 0.6. This outcome informs us that it is a good choice to set the retaining level as 0.6 when using DCWord for deceptive review identification. That is to say, it is better to use 60 percent of reviews to train the Skip-gram model to produce the word vectors for review representation. By manual checking, we find that on the one hand, when the retraining level is smaller than 0.4, there is much sparsity (i.e. zeros) in the produced word vectors by the Skip-gram model. However, when the retraining level is larger than 0.8, the produced word vectors are of little difference from each other and, the differences of elements of each word vector are reduced resulting in features losing discriminative capacity.

On the other hand, the review sentences are very similar to each other and there are many repetitive sentences in the reviews although the reviews are different from each other on the whole. Thus, it is of little help to improve word vectors for classification when using more reviews to produce word vectors at the sentence

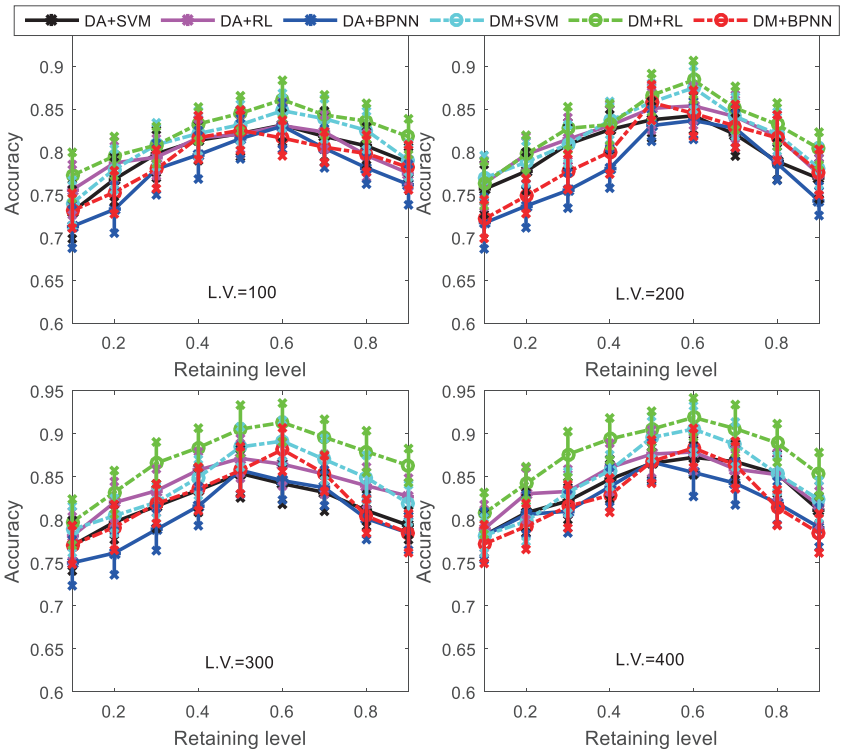


Figure 3 The Performances of the Proposed DCWord Approach in Deceptive Review Identification under Different Settings of Classifiers, Lengths of Vectors and Retaining Levels in the Spam Dataset. DA Abbreviates for the DCWord-A Strategy and DM Abbreviates for the DCWord-M Strategy

level. That is to say, 60 percentages of all reviews are enough to produce word vectors to capture the contextual information of words. We conjecture that more reviews than 60 percentages would introduce noise for word vectors to capture the contextual information of words.

Third, we see that the classifiers under the DCWord-M strategy perform better than the classifiers under the DCWord-A strategy. This outcome is consistent with Socher et al. (2011) and Collobert et al. (2011) where they claim that the max-pooling strategy is more effective than the averaging strategy in task of text classification. Moreover, we see that the LR classifier outperforms the SVM classifier and the BPNN classifier significantly under both strategies (Wilcoxon sign-rank test, $P<0.05$). This outcome is consistent with Vincent et al. (2010) and Hinton and Salakhutdinov (2006) where they claim that LR performs well in prediction task when combined with deep learning models such as denoising autoencoders and restricted Boltzman machine. Thus, it is not surprising that the LR classifier under the DCWord-M strategy (DM+LR) performs the best in the task of deceptive review identification among all the investigated methods.

It is unexpected that the SVM classifier performs only a slightly poorer than the LR classifier but better than the BPNN classifier. We explain that when using deep learning techniques to produce input vectors for classifier, LR is capable of this kind of task as a simple classifier. However, this does not necessarily mean that SVM, as a more complicated classifier than LR, is not good at the task in dealing with this task. We conjecture that the inability of BPNN in deceptive review identification as it has similar mechanism with the Skip-gram model by using neural network learning. Thus, the repeated adoption of word embedding by

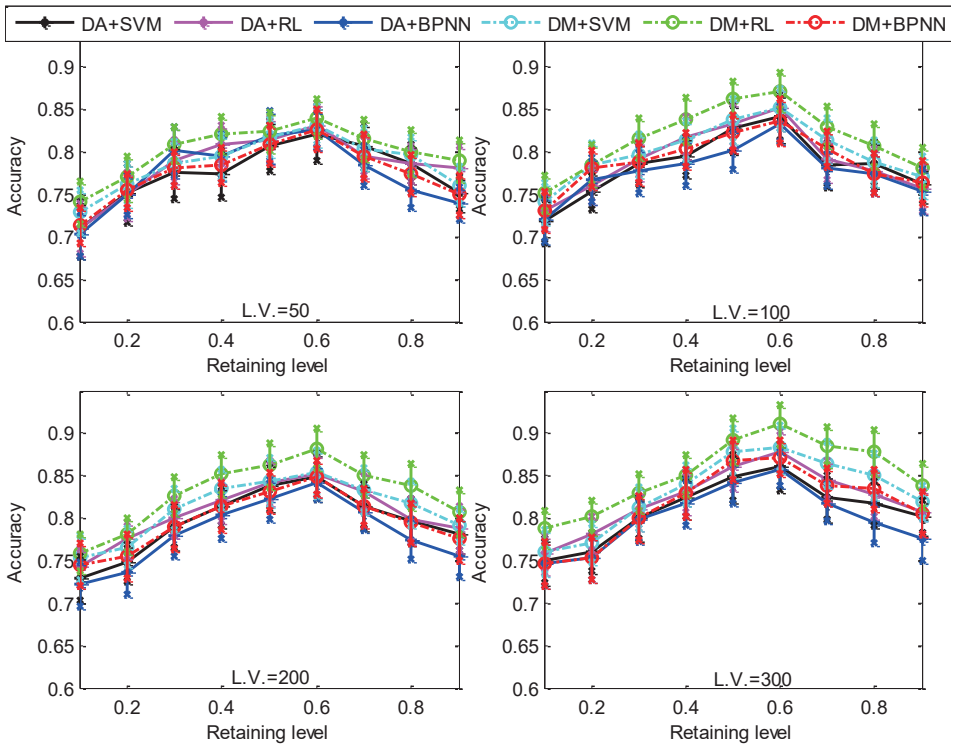


Figure 4 The Performances of the Proposed DCWord Approach in Deceptive Review Identification under Different Settings of Classifiers, Lengths of Vectors and Retaining Levels in the Deception Dataset. DA Abbreviates for the DCWord-A Strategy and DM Abbreviates for the DCWord-M Strategy

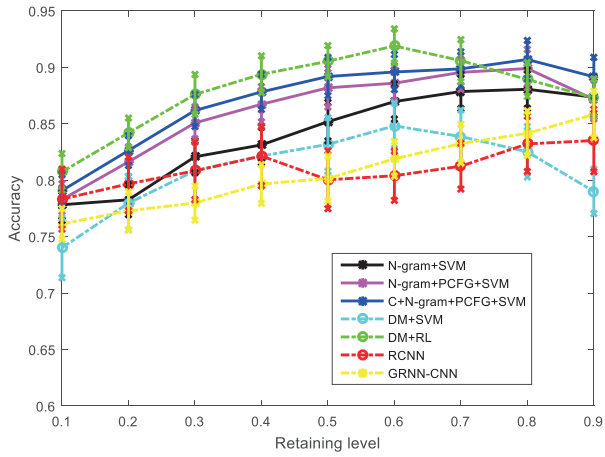
neurons deteriorates the classification performance.

Figure 4 shows the experimental results of deceptive review identification on the deception dataset. We set different lengths of word vectors (L.V.) as 50, 100, 200 and 300, respectively, because the number of unique words and the numbers of sentences of each review are smaller than that of the spam dataset as can be seen in Tables 1 and 2. We also attempt to augment its L.V. to 400 but, it cannot make further performance difference compared with that of L.V. as 300.

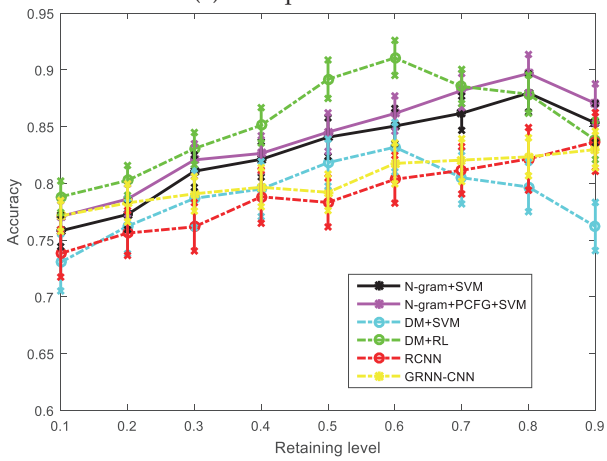
The similar outcome seen from **Figure 3** can also be seen from **Figure 4**. We can see that first, when the L.V. parameter is increasing from 50 to 300, the performances of all methods are improved significantly (Wilcoxon sign-rank test, $P<0.05$) and these improvements can be attributed to the enlargement of feature space of DCWord vectors. Second, the performances of all methods are increased to its maxima when the retaining level is set as 0.6. This outcome means that the context information of words can be learned by using 60 percentages of reviews to train the Skip-gram model via an optimal way. Third, the DCWord-M strategy performs better than the DCWord-A strategy with all the classifiers. This further validates that the max-pooling method is better than the average-pooling method. The LR classifier also performs the best among all the classifiers and this outcome further prove the feasibility of combing feature engineering by deep learning with LR for classification task (Pannakkong et al. 2018).

5.4 Baseline Comparison

Figure 5 shows the performances of the proposed DCWord approach compared with state-of-the-art techniques in deceptive review identification, including N-gram feature



(a) The Spam Dataset



(b) The Deception Dataset

Figure 5 The Performances of the Proposed DCWord Approach Compared with that of State-of-the-art Techniques in Deceptive Review Identification on the Spam Dataset (up) and the Deception Dataset (down), Respectively

set with SVM (N-gram+SVM) (Ott et al. 2011), N-gram and PCFG feature set with SVM (N-gram+PCFG+SVM) (Feng et al. 2012, Feng and Hirst 2013), recurrent convolutional neural network (RCNN) (Lai et al. 2015) and gated recurrent neural network and convolutional neural network (GRNN-CNN) (Ren and Ji 2017). All parameters in these state-of-the-art techniques are set as suggested in the references with best performances. We use the method from Feng and Hirst (2013) to extract the pro profile compatibility feature from hotels of the Spam dataset. However, this method is not suitable to extract features of doctors and restaurants from the Deception dataset for the reason that reviews in the Deception dataset are not aggregated by target objects and it is impossible for the method of compatibility profile to deal with them. Thus, the method of compatibility profile is not compared with the proposed DCWord method in the Deception dataset.

We can see from **Figure 5** that when the retaining level increases not larger than 0.8, the performances of all state-of-the-art methods are stably improved by using more training reviews. However, this is not the case for the proposed DCWord approach as it produces the best performance at the retaining level as 0.6. Moreover, in the Spam dataset, the best per-

formance of the proposed DCWord approach (DM+RL) is with accuracy as 0.9189 and in the Deception dataset, the best performance of the proposed DCWord approach (DM+RL) is with accuracy as 0.9106. However, for state-of-the-art techniques, the best performance derived by the method of compatibility profile is with accuracy as 0.9069 in the Spam dataset and, the best performance derived by the N-gram and PCFG feature set is with accuracy as 0.8968 in the Deception dataset.

This outcome can be explained as that, the proposed DCWord representation makes use of more contextual information from words by word embedding than other state-of-the-art methods. The performances of state-of-the-art methods in deceptive review identification are improved by the paradigm of adding more lexical and syntactic features of reviews to train the SVM model. For instance, the PCFG features are added into the feature set and a performance improvement is derived in comparison with that of the baseline N-gram features.

Further, the profile compatibility features of target objectives are added into the feature set and the performance of deceptive review identification is further improved. Thus, we can deduce that when using traditional NLP methods for deceptive review identification, the more features are used in training the classifier and the better performance the classifier will be derived. This outcome is also observed with the RCNN and GRNN-CNN methods. Thus, the problem of obtaining good performance can be transferred to how find more NLP features from reviews. However, this is not case for the DCWord approach, where it can learn contextual information of words automatically. With limited number of words derived from review, the proposed DCWord approach can also produce better performance than state-of-the-art techniques.

6. Concluding Remarks

This paper proposes a novel approach for representation called DCWord for deceptive review identification by deep learning word contextual information. With the Skip-gram model, we embed each word in reviews using two numeric vectors: one is from deceptive contextual information and the other one is from truthful contextual information. With these two types of word vectors, we further propose two strategies as the DCWord-A strategy and the DCWord-M strategy to represent reviews by numeric vectors. Three classifiers as SVM, LR and BPNN are used as classifiers with review vectors as input to identify deceptive reviews.

The experiments on the Spam dataset and the Deception dataset demonstrate that the proposed DCWord approach can produce better performances than state-of-the-art techniques in deceptive review identification. The proposed DCWord approach needs much smaller feature space and smaller number of training reviews to derive its better performances than state-of-the-art techniques. Moreover, the DCWord-M strategy has produced better performances than the DCWord-A strategy in both datasets. Among the three mentioned classifiers, the LR classifier has produced the best performance and this outcome is consistent with existing research.

Although the proposed DCWord approach has shown some promising aspects deceptive review identification, we admit that it still needs more improvement. In the future, we will consider combine convolutional neural network (Ciresan et al. 2011), recurrent neural network (Lai et al. 2015) and word vectors to learn the review vectors for representation automatically rather than adopting average-pooling strategy and max-pooling strategy.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China under Grant Nos. 71932002, 61379046, 91318302 and 61432001; the Innovation Fund Project of Xi'an Science and Technology Program (Special Series for Xi'an University under Grant No. 2016CXWL21). Also, the authors sincerely thank the referees for their much practical help to improve the quality of this paper.

Endnotes

[1] Dianping: http://www.dianping.com

[2] Koubei: http://www.koubei.com

[3] TripAdvisor: http://www.tripadviser.com

[4] Qtag for English part-of-speech, online: http://www.english.bham.ac.uk/staff/omason/software/qtag.html

[5] USPTO stop words, online: http://ftp.uspto.gov/patft/help/stopword.htm

[6] QTag for English part-of-speech, online: http://www.english.bham.ac.uk/staff/omason/software/qtag.html

[7] Porter stemming algorithm, online: http://tartarus.org/martin/PorterStemmer/

References

Cao L, Tang X, (2014). Topics and trends of the online public concerns based on Tianya forum. *Journal of Systems Science and Systems Engineering* 23(2):212-230.

Chatterjeei P (2001). Online reviews. Do consumers use them? *Proceedings of Conference on Association for Consumer Research*: 129-134.

Chen J, Zhou X, Tang X (2018). An empirical feasibility study of societal risk classification toward BBS posts. *Journal of Systems Science and Systems Engineering* 27(6):709-726.

Chen L, Wang F (2013). Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowledge-Based Systems* 50(3):44-59.

Ciresan D C, Meier U, Masci J, Gambardella L M, Schmidhuber (2011). Flexible, high performance convolutional neural networks for image classification. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*: 1237-1242.

Collobert R, Weston J (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. *Journal of Parallel & Distributed Computing*: 160-167.

Collobert R, Weston J, Bottou L, Karlen M., Kavukcuoglu K, Kuksa P (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(1):2493-2537.

Feng S, Banerjee R, Choi Y (2012). Syntactic stylometry for deception detection. *ACL*: 8-14.

Feng V W, Hirst G. (2013). Detecting deceptive opinions with profile compatibility. *International Joint Conference on Natural Language Processing*: 14-18.

Firth J R (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis. Philological Society* 40(2):305-321.

Gokhman S, Hancock J, Prabhu P, Ott M, Cardie C (2012). In search of a gold standard in studies of deception. *Proceedings of the EACL 2012 Workshop on Computational Approaches to Deception Detection*: 23-27.

Guo C, Du Z, Kou X (2018). Products ranking through aspect-based sentiment analysis of online heterogeneous reviews. *Journal of Systems Science and Systems Engineering* 27(5):542-558.

Hinton G E, Salakhutdinov R R (2006). Reducing the dimensionality of data with neural networks. *Science* 313(5786):504-507.

Jindal N, Liu B (2008). Opinion spam and analysis. *International Conference on Web Search and Data Mining*, ACM.

Kietzmann J, Canhoto A (2013). Bittersweet! Understanding and managing electronic word of mouth. *Journal of Public Affairs* 13(2):146-159.

Klein D, Manning C D (2003). Accurate unlexicalized parsing. *Meeting on Association for Computational Linguistics*: 423-430.

Lai S, Xu L, Liu K, Zhao J (2015). Recurrent convolutional neural network for text classification. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*: 2267-2273.

Li F, Huang M, Yang Y, Zhu X (2011). Learning to identify review spam. *International Joint Conference on Artificial Intelligence*: 2488-2493.

Li J, Ott M, Cardie C, Hovy E (2014). Towards a general rule for identifying deceptive opinion spam. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*: 1566-1576.

Lim Y J, Osman A, Salahuddin S N, Romle A R, Abdullah S(2016). Factors influencing online shopping behavior: The mediating role of purchase intention. *Procedia Economics and Finance* 35:401-410.

Liu B (2012). Opinion spam detection: Detecting fake reviews and reviewers. https://www.cs.uic.edu/liub/FBS/fake-reviews.html.

Liu Q, Gao Z, Liu B, Zhang Y (2013). A logic programming approach to aspect extraction in opinion mining. *Ieee/wic/acm International Joint Conferences on Web Intelligence* 1:276-283.

Marrese-Taylor E, Velásquez J D, Bravo-Marquez F, Matsuo Y (2013). Identifying customer preferences about tourism products using an aspect-based opinion mining approach. *Procedia Computer Science* 22:182-191.

Mikolov T, Chen K, Corrado G, Dean J (2013). Efficient estimation of word representations in vector space. *Computer Science*: 1301.

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* 26:3111-3119.

Mudambi S M, Schuff D (2010). What makes a helpful online review? A study of customer reviews on Amazon.com. *MIS Quarterly* 34(1):185-200.

Nitin I, Fred J D, Zhang T (2005). Text mining: Predictive methods for analyzing unstructured information. *Springer Science and Business Media*: 15-37.

Ott M, Choi Y, Cardie C, Hancock J T (2011). Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*: 19-24.

Pannakkong W, Sriboonchitta S, Huynh V (2018). An ensemble model of arima and ann with restricted boltzmann machine based on decomposition of discrete wavelet transform for time series forecasting. *Journal of Systems Science and Systems Engineering* 27(5):690-708.

Ren Y, Ji D (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences* 385:213-224.

Ren Y, Zhang Y (2016). Deceptive opinion spam detection using neural network. *Proceedings of the 26th International Conference on Computational Linguistics*:140-150.

Socher R, Lin C Y, Ng A Y, Manning C D (2011). Parsing natural scenes and natural language with recursive neural networks. *International Conference on Machine Learning*: 129-136.

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P A (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11(12):3371-3408.

Zhang W, Yoshida T, Tang X (2007). Text classification toward a scientific forum. *Journal of Systems Science and Systems Engineering* 16(3):356-379.

Zhang W, Yoshida T, Tang X (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems* 21(8):879-886.

Zhang W, Yoshida T, Tang X, Ho T (2009). Improving effectiveness of mutual information substantival multiword expression extraction. *Expert Systems with Application* 36(8):10919-10930.

Zhou L, Shi Y, Zhang D (2008). A statistical language modeling approach to online deception detection. *IEEE Transactions on Knowledge & Data Engineering* 20(8):1077-1081.

Wen Zhang is a professor of College of Economics and Management at Beijing University of Technology (BJUT). He received his PhD degree in knowledge science from the Japan Advanced Institute of Science and Technology in 2009. His recent research interests include machine learning, data mining, and information systems.

Qiang Wang is a PhD candidate of College of Economics and Management at Beijing University of Technology (BJUT). He received his BS degree in marketing from Qufu Normal University in 2016. His research interest includes E-commerce big data analysis, data mining, and machine learning.

Xiangjun Li is a professor with School of Information Engineering, Xi'an University. She received her PhD from Xidian University in 2013. Her current research interest includes data mining, knowledge discovery, and machine learning.

Taketoshi Yoshida is a professor with School of Knowledge Science, Japan Advanced Institute of Science and Technology. He received his PhD degree in systems engineering from Case Western Reserve University in 1984. His current research interest includes knowledge management, knowledge discovery, and information systems.

Jian Li is a professor of College of Economics and Management at Beijing University of Technology (BJUT). He received his PhD degree from Chinese Academy of Sciences in 2007. His recent research interests include supply chain finance, blockchain technology, and emergency management.