Review Spam Detection Using Word Embeddings and Deep Neural Networks

Aliaksandr Barushka(✉) and Petr Hajek(✉)

Institute of System Engineering and Informatics,
Faculty of Economics and Administration, University of Pardubice,
Studentska 84, 532 10 Pardubice, Czech Republic
aliaksandr.barushka@student.upce.cz,
petr.hajek@upce.cz

Abstract. Review spam (fake review) detection is increasingly important taking into consideration the rapid growth of internet purchases. Therefore, sophisticated spam filters must be designed to tackle the problem. Traditional machine learning algorithms use review content and other features to detect review spam. However, as demonstrated in related studies, the linguistic context of words may be of particular importance for text categorization. In order to enhance the performance of review spam detection, we propose a novel content-based approach that considers both bag-of-words and word context. More precisely, our approach utilizes *n*-grams and the skip-gram word embedding method to build a vector model. As a result, high-dimensional feature representation is generated. To handle the representation and classify the review spam accurately, a deep feed-forward neural network is used in the second step. To verify our approach, we use two hotel review datasets, including positive and negative reviews. We show that the proposed detection system outperforms other popular algorithms for review spam detection in terms of accuracy and area under ROC. Importantly, the system provides balanced performance on both classes, legitimate and spam, irrespective of review polarity.

Keywords: Review spam · Skip-gram · Word2vec · Word embedding · Neural network

1 Introduction

Review spam (fake review) can be defined as unwanted and misleading messages that can be published on multiple platforms, such as forums or online shops. User base of those online platforms is steadily growing over the years. For instance, TripAdvisor is the world largest travel site with over 455 million average monthly unique visitors and 600 million reviews and opinions covering 7.5 million accommodations, airlines, attractions, and restaurants [1]. Many travelers rely on reviews before choosing a hotel to stay. Online reviews have become a concern of the industry, since review spam may mislead purchasers and eventually lead to a lawsuit against the seller. According to recent statistics, every third review is spam on TripAdvisor [2]. By writing positive or negative review spam, the seller may gain a competitive advantage or disadvantage,

Published by Springer Nature Switzerland AG 2019
J. MacIntyre et al. (Eds.): AIAI 2019, IFIP AICT 559, pp. 340–350, 2019.
https://doi.org/10.1007/978-3-030-19823-7_28

respectively. To guarantee fair competition, it is therefore crucial for shopping portals to identify and block review spam and ban fraud users.

Review spam can be detected either manually or automatically. Compared to automatic detection, manual review detection is slow, expensive and relatively inaccurate [3]. Therefore, over the past decade, researches have explored ways to improve automatic review spam detection. Machine learning approaches, such as neural networks, support vector machines or Naïve Bayes, have a reputation of effective methods in detecting review spam [4]. Such methods utilize content-based and other features of reviews to filter review spam accurately. Review spam detection is usually considered as a binary classification problem, in which each review is classified either as legitimate (trustworthy) or spam (fake). Besides overall high classification accuracy, low false positive rate is of particular importance because, otherwise, users of shopping portals would not be able to see legitimate reviews and trustworthy users would get offended and may lose motivation to submit reviews on the particular portal. The main idea behind content-based machine learning models is to build a word (phrase) list and assign a weight to each word or phrase (bag-of-words) or word category (part-of-speech tagging or psycholinguistic) [5]. However, such features suffer from sparsity, which makes it difficult to capture semantic representation of reviews. To address this issue, Ren and Ji [6] proposed a gated recurrent neural network model to detect opinion spam. This approach utilized word embeddings obtained by using the CBOW (continuous bag-of-words) model [7, 8] so that words are mapped to vectors based on their context. Thus, global semantic information can be obtained, and, to certain degree, the problem of scarce data is overcome. This approach was reportedly more effective than traditional bag-of-words or part-of-speech tagging.

Inspired by these recent findings, here we utilize the word embeddings to obtain the semantic representation of online reviews. Word2vec [7, 8] is a popular method to produce word embeddings (vector space model) from a corpus of text data. The word representation can be obtained by two alternative model architectures, namely CBOW or skip-gram. Unlike earlier literature, here we use a skip-gram model for this task, which exploits word context more effectively and thus generates a more generalizable context when compared with the CBOW model [7]. To train the skip-gram model, we use the hierarchical softmax algorithm, a computationally effective version of the softmax algorithm. To further enhance the detection performance, we combine the generated word embeddings with bag-of-words in the second stage and train a deep feed-forward neural network (DNN) to classify spam/legitimate reviews. DNN is used to capture complex features hidden in high-dimensional data representations [9–11].

The rest of the paper has the following structure. In Sect. 2, the literature review of the recent advances in review spam detection is introduced. Section 3 describes the corpora of hotel reviews that are used in experiments. In Sect. 4, our model for review spam detection is introduced. Section 5 presents the results of the experiments and the final section summarizes our findings and suggests future research directions.

2 Review Spam Detection – A Literature Review

Review spam has been increasingly subject to scrutiny owing to the outstanding importance of product reviews that are used for purchase and business decisions. To influence the decisions and thus make profit, spam reviews are produced to promote (positive reviews) or demote (negative reviews) some products [6]. To detect those deceptive reviews (opinions), machine learning methods have been used because, as shown in earlier literature [3, 4], human readers have limited capacity to detect spam reviews. This task is typically handled as a classification problem, with reviews categorized as spam or legitimate class. This annotation (class label) is provided by people and the aim of the machine learning-based detection system is to automatically classify reviews into these classes.

One of the first published effort to detect review spam utilized the fact that spammers duplicate their reviews, either on the same or different product [12]. Similarly, spam scoring proposed in [13] was based on the cosine similarity between reviews. Furthermore, Wang et al. [14] developed a review graph to capture the interactions among reviews, reviewers and stores. Thus, the honesty of reviews could be calculated. Interestingly, this approach did not use any review text information. In contrast, the approach proposed in [15] was based on text features only. Li et al. [16] examined the effect of several feature categories on review spam identification, including content, sentiment, product or profile features. Review metadata were integrated with relational features in SpEagle, a unified framework to rank reviews [17]. Unusual temporal patterns of correlated review ratings were also used to detect spam attacks [18]. These patterns make real-time detection of abnormal events possible [19, 20]. Spatial patterns (user IPs) were utilized together with temporal patterns in [21].

Most existing approaches focus on extracting informative features from the texts of reviews to enhance detection performance. Traditional features include bag-of-words, part-of-speech tagging or psycholinguistic word lists [5, 22, 23]. For example, Ott et al. [24] identified *n*-gram features in the texts and then employed support vector machines (SVMs) to perform the classification of reviews. As noted above, the semantic representation of reviews can be captured by word embeddings. In addition to their use in [6], the CBOW model was also combined with network features in a semi-supervised approach developed by [25]. A sentence weighted convolutional neural network was used to obtain the document representations for review spam detection [26].

3 Dataset

In this study, we used two datasets from Cornell University[1], namely positive hotel review spam [22] and negative hotel review spam [24]. Since the details on these datasets can be found in [22, 24], we provide only their brief description in this study.

The positive hotel review spam dataset contained 400 legitimate and 400 spam positive reviews from TripAdvisor (20 legitimate and 20 spam reviews for each of the

[1] http://myleott.com/op-spam.html.

20 selected hotels). The spam reviews were gathered using Amazon Mechanical Turk. Only a single review per Turker was allowed, and unreasonably short or plagiarized reviews were rejected. For the positive dataset, only 5-star reviews were included.

A similar procedure was used to collect the negative hotel review spam dataset. Again, Turkers were employed to provide spam reviews on 20 popular hotels, such as such as Affinia Chicago or Ambassador East Hotel, and corresponding legitimate reviews were obtained from several online review communities, such as Expedia, TripAdvisor or Hotels.com. For the negative dataset, only 1- or 2-star reviews were used. The average review length for both datasets was 116 words. The datasets included the following types of information: message content, spam label, hotel information, polarity of the message, and travel agency aggregator name.

To pre-process the content of the reviews in the datasets, several tools were applied. First, tokenization was performed using standard delimiters: ".,;:'"()?!". Second, the stopwords were removed using the Rainbow stopwords list to reduce the noise in the data. Third, all numbers, punctuation and some special symbols were stripped off and all tokens were transformed to lowercase letters.

For the bag-of-words representation, *tf.idf* weighting scheme was used, in which the weight a term is obtained as follows:

$$v_{ij} = \left(1 + \log\left(tf_{ij}\right)\right) \times \log(N/df_i), \tag{1}$$

where v_{ij} is the weight of the i-the term in the j-th review, tf_{ij} is term frequency, df_i is document frequency, and N is the number of reviews. This weighting scheme was selected because it takes into account review lengths and term rareness.

4 Methods

The proposed architecture for spam review detection is depicted in Fig. 1. To avoid overfitting, the datasets were divided into training and testing data using 10 times repeated stratified 10-fold cross-validation.

In the n-gram model, we used the bag-of-words representation as defined in Eq. (1). In this model, text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity. In bag-of-words, string attributes are converted into a set of numeric attributes representing word occurrence information from the text contained in the strings. Note that only most relevant terms (attributes) were selected according to their weights v_{ij}. In agreement with previous studies [10, 11], top 2000 terms were retained, including bigrams and trigrams as suggested in [26]. An example of features with the highest information gain is presented in Table 1.

To obtain word embeddings, the skip-gram model was employed. This is a language modelling and feature learning technique that maps words or phrases from the vocabulary to vectors of numerical values. Word embeddings are unsupervisedly

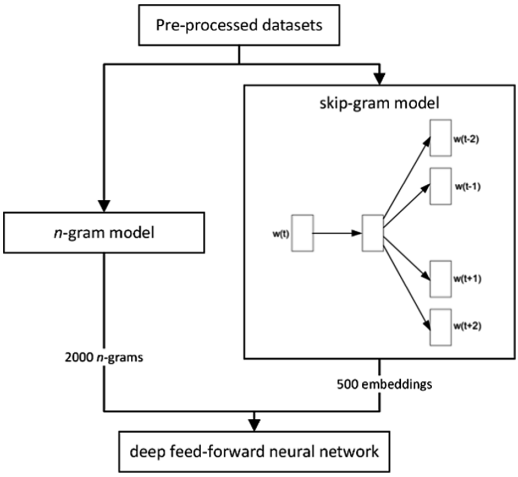

Fig. 1. The proposed architecture for spam review detection.

Table 1. Top 10 features from the *n*-gram model in terms of information gain (IG).

Negative dataset		Positive dataset	
Feature	IG	Feature	IG
"chicago"	0.123	"chicago"	0.090
"at the"	0.046	"location"	0.062
"luxury"	0.044	"floor"	0.052
"location"	0.043	"bathroom"	0.044
"-"	0.038	"on the"	0.041
"when i"	0.036	"small"	0.037
"chicago hotel"	0.034	"reviews"	0.037
"smell"	0.033	"luxury"	0.037
"my room"	0.033	"2"	0.037
"recently"	0.030	"priceline"	0.035

learned word representation vectors whose relative similarities correlate with semantic similarity. The skip-gram model, one of the word2vec methods, includes the following steps [7, 8]:

- obtain a training dataset (sequences of words) $w_1, w_2, \ldots, w_T$;
- train the classifier and embedding function parameters;
- process each word w_t in the vocabulary by applying embedding function to generate digital representation for every word in the vocabulary in high-dimensional space;
- map every word in the vocabulary to digital representation of the word.

The skip-gram model aims to find word representations that can be used to predict the context words in a sentence. The objective function of the skip-gram model is defined as follows:

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c}\log p\left(w_{t+j}|w_t\right), \tag{2}$$

where $w_1, w_2, \ldots, w_T$ is a sequence of training words, c is the size of context, and $p(w_{t+1}|w_t)$ is defined using the hierarchical softmax (a binary tree representation of the output layer) as follows [7]:

$$p(w|w_I) = \prod_{j=1}^{L(w)-1}\sigma\left(\left[\!\left[n(w,j+1) = \mathrm{ch}(n(w,j))\right]\!\right]v'^{T}_{n(w,j)}v_{w_I}\right), \tag{3}$$

where w_I are input words, v_w and v'_w are the input and output vector representations of word w, respectively, $n(w, j)$ is the j-th node in the tree, $L(w)$ is the length of the path from root node to word w, ch(n) is a child node of n chosen arbitrarily, $[\![x]\!] = 1$ if x is true, otherwise $[\![x]\!] = -1$, and $\sigma(x)$ is a sigmoidal function. Given the vocabulary size V, the computational complexity per training example per context word is $O(\log(V))$, which is a substantial improvement over the original softmax ($O(V)$).

The size of the word vectors (embeddings) was set to 500 and context size c = 5 [7] to generate a complex representation. The average values of the vector were used to represent each review. Thus, the input attributes (features) for the subsequent supervised learning included 2000 n-grams and 500 embeddings.

Deep feed-forward neural network (DNN) was used to classify reviews into spam/legitimate categories. DNN enables processing the complex sparse representations of documents just like online reviews [10]. To reduce the risk of overfitting and avoid poor local error minima, we used dropout regularization and rectified linear units, respectively, as suggested in [9]. To train the DNN, we used the mini-batch gradient descent algorithm that ensures stable convergence. This algorithm updates the synapse weights as follows:

$$s_{t+1} = s_t - \eta\nabla_\theta J\left(w_t; x^{(i:i+n)}; y^{(i:i+n)}\right), \tag{4}$$

where s is synapse weight, t is the iteration index, η is learning rate, J denotes an objective function, x^i is the input, y^i is the output of the i-the training example, and n is size of the mini-batch. The DNN was trained using different numbers of neurons in hidden layers = {10, 20, 50, 100} and different numbers of hidden layers = {1, 2, 3}. A grid search procedure was used to find the optimal DNN structure. Dropout rate for the input and hidden layers were set to 0.2 and 0.5, respectively, and the number of iterations was 1000, η = 0.1 and n = 100.

5 Experimental Results

While evaluating the experimental results, we took into consideration four evaluation measures: accuracy, area under ROC (receiver operating characteristic) curve (AUC), FN (false negative) rate and FP (false positive) rate. Accuracy represents the percentage of reviews correctly classified, FP rate stands for the percentage of spam reviews incorrectly classified as legitimate, while FP rate is the percentage of legitimate reviews misclassified as spam. Hereinafter, we present the averages and standard deviations of 100 experiments (10 times repeated stratified 10-fold cross-validation) performed on the two datasets. The *n*-gram and skip-gram models, as well as the experiments with the DNN method were performed in Deeplearning4j program environment.

To demonstrate the effectiveness of the proposed detection system, we also compared the results with several state-of-the-art approaches:

- Convolutional neural network (CNN) was used in [6, 26] as a basic CNN with unigrams and bigrams as inputs. Here we trained the CNN model using the mini-batch gradient descent algorithm with patch size 5×5 and max pool size 2×2, the remaining parameters were the same as for the DNN model;
- Naïve Bayes represents a baseline classifier used in earlier research [16];
- Support vector machine (SVM) represents another popular method used in previous literature on review spam detection [3, 22, 24]. It was also used as a baseline method in recent studies [20]. In this study, SVM was trained using the SMO algorithm with varying complexity $C = \{2^0, 2^1, \ldots, 2^6\}$ and polynomial kernel function;
- Random Forest (RF) is another well-performing benchmark method used in several comparative studies [10, 11]. Here we used it with 100 random trees.

Note that all the baselines are suitable for review spam detection due to their qualities in handling sparse and high-dimensional data [10]. All the experiments with the comparative methods were run in Weka 3.8.2 program environment. The results of the conducted experiments are provided in Tables 2, 3, 4 and 5. To show the improvement in performance, we also compared the results with those obtained for the traditional *n*-gram model.

The results in Table 2 show that the *n*-gram + skip-gram approach performs well in terms of accuracy rate. All machine learning algorithms benefit from this combination except RF. It is important to note that the DNN approach achieves the highest accuracy rate and that the proposed method helps improve the performance of the *n*-gram model for both positive and negative polarity by about 2%. Compared with the baseline methods, the DNN method performed best for the negative dataset but CNN and SVM performed similarly at $P < 0.05$ using the Wilcoxon signed rank test. For the positive dataset, DNN performed significantly better than the compared counterparts. What is interesting in this dataset is that the performance of DNN is 4% better than the baselines. These results corroborate those achieved in the original paper [24] where the highest accuracies were 86.0% and 89.3% for the negative and positive dataset, respectively. However, this comparison must be interpreted with caution because different cross-validation procedure was used in [24]. In summary, Table 2 shows the

superiority of DNN in review spam detection and substantial improvement achieved using the skip-gram model over the baseline *n*-gram model.

The results in Table 2 demonstrate that the *n*-gram + skip-gram outperforms the *n*-gram model in terms of FN rate except the RF method that, however, performed relatively poorly considering FP rate. In fact, FP rate is usually preferable to FN rate in the related literature [27] due to the importance of retaining legitimate reviews and trustworthy users. From this perspective, DNN and CNN are the best methods. The most remarkable result to emerge from Tables 3 and 4 is the balanced performance of DNN on both classes, legitimate and spam.

Table 2. Results of the experiments - accuracy.

	Negative dataset		Positive dataset	
	n-gram	*n*-gram + skip-gram	*n*-gram	*n*-gram + skip-gram
NB	80.36 ± 3.12	81.75 ± 3.02	82.88 ± 3.12	84.38 ± 2.96
SVM	84.00 ± 4.36	86.50 ± 2.99*	81.63 ± 5.04	84.50 ± 2.71
DNN	86.88 ± 4.09*	**88.38 ± 3.12**	87.25 ± 2.99*	**89.75 ± 3.05**
RF	85.25 ± 4.67	84.50 ± 3.59	86.50 ± 3.76	84.00 ± 4.71
CNN	82.25 ± 3.67	88.13 ± 4.57*	81.75 ± 11.70	85.75 ± 12.87

*Significantly similar performance as the best at $P < 0.05$ (Wilcoxon signed rank test)

Table 3. Results of the experiments – FN rate.

	Negative dataset		Positive dataset	
	n-gram	*n*-gram + skip-gram	*n*-gram	*n*-gram + skip-gram
NB	0.198 ± 0.061	0.208 ± 0.047	0.158 ± 0.053	0.170 ± 0.054
SVM	0.155 ± 0.028	0.138 ± 0.041*	0.170 ± 0.079	0.145 ± 0.052
DNN	0.140 ± 0.047	**0.115 ± 0.038**	0.128 ± 0.048	**0.103 ± 0.038**
RF	0.118 ± 0.044*	0.170 ± 0.048	0.110 ± 0.058*	0.135 ± 0.050
CNN	0.190 ± 0.058	0.123 ± 0.066*	0.130 ± 0.074	0.180 ± 0.290

*Significantly similar performance as the best at $P < 0.05$ (Wilcoxon signed rank test)

Table 4. Results of the experiments – FP rate.

	Negative dataset		Positive dataset	
	n-gram	*n*-gram + skip-gram	*n*-gram	*n*-gram + skip-gram
NB	0.195 ± 0.064	0.158 ± 0.056	0.185 ± 0.056	0.143 ± 0.044
SVM	0.165 ± 0.068	0.133 ± 0.049*	0.198 ± 0.066	0.165 ± 0.043
DNN	0.123 ± 0.056*	0.118 ± 0.033*	0.128 ± 0.049	**0.103 ± 0.034**
RF	0.178 ± 0.077	0.140 ± 0.036	0.160 ± 0.050	0.185 ± 0.068
CNN	0.165 ± 0.054	**0.115 ± 0.047**	0.235 ± 0.246	0.105 ± 0.055*

*Significantly similar performance as the best at $P < 0.05$ (Wilcoxon signed rank test)

As can be seen from Table 5, the DNN method performs well on both classes even when only the *n*-gram model is used. However, the additional semantic representation provided by the skip-gram model leads to further improvement. Among the compared baselines, CNN performs well on the negative dataset but only when the skip-gram model is included, suggesting that this method is more sensitive to both the polarity of the reviews and word context. In contrast, RF provides good performance on the positive dataset even with the baseline *n*-gram model. This can be explained by the fact that RF tends to apply features with many distinct values (embeddings, in this case), rather than those with few distinct values (unigrams and bigrams). As a result, the information contained in *n*-grams is not fully utilized in the *n*-gram + skip-gram model. In that sense, DNN achieves a more robust performance, with high AUC for both datasets. In addition, the DNN models integrating the *n*-gram and skip-gram features consistency outperforms those with *n*-grams only.

Table 5. Results of the experiments – AUC.

	Negative dataset		Positive dataset	
	n-gram	*n*-gram + skip-gram	*n*-gram	*n*-gram + skip-gram
NB	0.833 ± 0.044	0.880 ± 0.036	0.886 ± 0.029	0.895 ± 0.032
SVM	0.840 ± 0.044	0.865 ± 0.030	0.816 ± 0.050	0.845 ± 0.027
DNN	0.946 ± 0.020*	0.956 ± 0.013*	0.950 ± 0.023*	**0.956 ± 0.025**
RF	0.924 ± 0.033	0.925 ± 0.025	0.943 ± 0.023*	0.932 ± 0.026*
CNN	0.915 ± 0.026	**0.958 ± 0.017**	0.877 ± 0.155	0.923 ± 0.113

*Significantly similar performance as the best at $P < 0.05$ (Wilcoxon signed rank test)

6 Conclusion

In this study, we demonstrated that using word embedding methods help achieve better results in review spam detection across state-of-the-art classification methods. The results show that the proposed content-based approach based on the DNN method performed best in terms of accuracy, FN and FP rate and AUC for the review spam dataset with positive polarity. Moreover, the proposed system demonstrated decent and balanced performance also for the dataset with negative polarity. Therefore, the proposed model can be recommended for review spam, irrespective of reviews' polarity.

The results of the experiments suggest that DNN with word embeddings might be used in alternative document categorization tasks with sparse high-dimensional data, such as e-mail and social network spam detection. In further research, it would be interesting to include additional features, including the content of the other reviews for the same hotel. Moreover, it would be worth using the proposed approach in different review domains, rather than only hotel reviews. Additional important features could be extracted using attention-based recurrent neural networks that have recently been reported to be effective in social network spam filtering [28]. Different languages represent another challenge for future research.

Acknowledgments. This article was supported by the by the grant No. SGS_2019_17 of the Student Grant Competition.

References

1. TripAdvisor Homepage. http://ir.tripadvisor.com/. Accessed 21 January 2019
2. The Times. https://www.thetimes.co.uk/article/hotel-and-caf-cheats-are-caught-trying-to-buy-tripadvisor-stars-027fbcwc8. Accessed 22 January 2019
3. Harris, C.: Detecting deceptive opinion spam using human computation. In: Workshops at AAAI on Artificial Intelligence, pp. 87–93. AAAI (2012)
4. Heydari, A., ali Tavakoli, M., Salim, N., Heydari, Z.: Detection of review spam: a survey. Expert Syst. Appl. **42**(7), 3634–3642 (2015)
5. Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H.: Survey of review spam detection using machine learning techniques. J. Big Data **2**(1), 1–23 (2015)
6. Ren, Y., Ji, D.: Neural networks for deceptive opinion spam detection: an empirical study. Inf. Sci. **385**, 213–224 (2017)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, NIPS, vol. 26, pp. 3111–3119 (2013)
8. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, vol. 32, pp. 1188–1196. JMLR (2014)
9. Barushka, A., Hájek, P.: Spam filtering using regularized neural networks with rectified linear units. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) AI*IA 2016. LNCS (LNAI), vol. 10037, pp. 65–75. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49130-1_6
10. Barushka, A., Hajek, P.: Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Appl. Intell. **48**(10), 3538–3556 (2018)
11. Barushka, A., Hajek, P.: Spam filtering in social networks using regularized deep neural networks with ensemble learning. In: Iliadis, L., Maglogiannis, I., Plagianakos, V. (eds.) AIAI 2018. IAICT, vol. 519, pp. 38–49. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92007-8_4
12. Jindal, N., Liu, B.: Analyzing and detecting review spam. In: 7th IEEE International Conference on Data Mining (ICDM 2007), pp. 547–552. IEEE (2007)
13. Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W.: Detecting product review spammers using rating behaviors. In: 19th ACM International Conference on Information and Knowledge Management, pp. 939–948. ACM (2010)
14. Wang, G., Xie, S., Liu, B., Philip, S.Y.: Review graph based online store review spammer detection. In: 11th International Conference on Data mining (ICDM 2011), pp. 1242–1247. IEEE (2011)
15. Lau, R.Y., Liao, S.Y., Kwok, R.C.W., Xu, K., Xia, Y., Li, Y.: Text mining and probabilistic language modeling for online review spam detecting. ACM Trans. Manage. Inf. Syst. **2**(4), 1–30 (2011)
16. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to identify review spam. In: International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 2488–2493 (2011)
17. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994. ACM (2015)

18. Xie, S., Wang, G., Lin, S., Yu, P.S.: Review spam detection via temporal pattern discovery. In: 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 823–831. ACM (2012)
19. Ye, J., Kumar, S., Akoglu, L.: Temporal opinion spam detection by multivariate indicative signals. In: 10th International AAAI Conference on Web and Social Media (ICWSM 2016), pp. 743–746. AAAI (2016)
20. Li, H., et al.: Bimodal distribution and co-bursting in review spam detection. In: 26th International Conference on World Wide Web, pp. 1063–1072 (2017)
21. Li, H., Chen, Z., Mukherjee, A., Liu, B., Shao, J.: Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: 9th International AAAI Conference on Web and Social Media (ICWSM 2015), pp. 634–637. AAAI (2015)
22. Ott, M., Cardie, C., Hancock, J.: Estimating the prevalence of deception in online review communities. In: 21st International Conference on World Wide Web, pp. 201–210. ACM (2012)
23. Liu, Y., Pang, B.: A unified framework for detecting author spamicity by modeling review deviation. Expert Syst. Appl. **112**, 148–155 (2018)
24. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 497–501 (2013)
25. Yilmaz, C.M., Durahim, A.O.: SPR2EP: a semi-supervised spam review detection framework. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 306–313. IEEE (2018)
26. Li, L., Qin, B., Ren, W., Liu, T.: Document representation and feature combination for deceptive spam review detection. Neurocomputing **254**, 33–41 (2017)
27. Zhang, Y., Wang, S., Phillips, P., Ji, G.: Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowl.-Based Syst. **64**, 22–31 (2014)
28. Chen, T., Li, X., Yin, H., Zhang, J.: Call attention to rumors: deep attention based recurrent neural networks for early rumor detection. In: Ganji, M., Rashidi, L., Fung, Benjamin C.M., Wang, C. (eds.) PAKDD 2018. LNCS (LNAI), vol. 11154, pp. 40–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04503-6_4