

Opinion Mining of Consumer Reviews Using Deep Neural Networks with Word-Sentiment Associations

Petr Hajek¹[0000-0001-5579-1215], Aliaksandr Barushka¹[0000-0001-6732-0277], Michal Munk^{1,2}[0000-0002-9913-3596]

¹Institute of System Engineering and Informatics, Faculty of Economics and Administration,
University of Pardubice, Studentska 84,
532 10 Pardubice, Czech Republic
petr.hajek@upce.cz, aliaksandr.barushka@student.upce.cz
²Department of Computer Science, Constantine the Philosopher University in Nitra, 949 74 Ni-
tra, Slovakia,
mmunk@ukf.sk

Abstract. Automated opinion mining of consumer reviews is becoming increasingly important due to the rising influence of reviews on online retail shopping. Existing approaches to automated opinion classification rely either on sentiment lexicons or supervised machine learning. Deep neural networks perform this classification task particularly well by utilizing dense document representation in terms of word embeddings. However, this representation model does not consider the sentiment polarity or sentiment intensity of the words. To overcome this problem, we propose a novel model of deep neural network with word-sentiment associations. This model produces richer document representation that incorporates both word context and word sentiment. Specifically, our model utilizes pre-trained word embeddings and lexicon-based sentiment indicators to provide inputs to a deep feed-forward neural network. To verify the effectiveness of the proposed model, a benchmark dataset of Amazon reviews is used. Our results strongly support integrated document representation, which shows that the proposed model outperforms other existing machine learning approaches to opinion mining of consumer reviews.

Keywords: Opinion Mining, Consumer Review, Word Embedding, Lexicon, Sentiment, Deep Neural Network.

1 Introduction

Opinion mining (sentiment analysis) of consumer reviews studies consumers' opinions on products and services [1]. The increasing number of users on online platforms produces a huge number of online product reviews. In the last two decades, opinion mining has become one of the most important text classification tasks because consumers' opinions affect the purchase decisions of other consumers. In addition, consumers' opinions in online reviews provide invaluable insights into consumer behavior and are

thus central to companies. The large number of consumer reviews available across diverse online sources has led to the necessity of employing automated opinion mining systems. Numerous machine learning methods have been used for this task, including methods with supervised learning and methods exploiting sentiment lexicons [1]. Recently, deep neural networks have emerged as an effective tool. Multiple layers enable learning complex representations of features [2]. Many deep neural networks in this domain use word embeddings as input features. Thus, words are transformed from a high-dimensional sparse space to lower-dimensional dense vectors, representing latent features and word context.

Opinion mining has been investigated at three levels of granularity, namely the document, sentence and aspect levels. For example, product reviews can be represented as documents classified into positive or negative opinion categories. Note that in this task, it is assumed that the review concerns a single product entity. In sentence-level categorization, only opinionated sentences must be first selected. Aspect-level opinion mining requires the identification of a product's aspect (target). In other words, this approach comprises several subtasks such as aspect extraction and aspect opinion classification.

Concerning the features used for opinion mining of consumer reviews, the bag-of-words model represents a traditional document representation in which word frequencies are calculated for each word (phrase) in the vocabulary [3]. However, this approach results in high-dimensional sparse document representation. Moreover, this representation ignores word order. In the case of using n -grams instead of single words, a short context is considered. To overcome these problems, word embeddings were introduced to produce low-dimensional dense word representation [4–7]. Compared with bag-of-words, word embeddings are also more effective in modeling word context and word meaning. After the appropriate document representation is generated, various neural network models can be employed for opinion classification. Alternatively, neural networks can be used to produce word embeddings; then other machine learning methods, such as support vector machines, can be used for the classification task [8].

The core problem of word embedding representations in existing studies is that the sentiment polarity and intensity of the words are ignored. As a result, a word embedding may comprise words with opposite sentiment polarity. This study aims to overcome this problem by developing a deep neural network model integrating word embeddings with their sentiment associations obtained from a wide range of lexicons. To further improve the performance of the opinion classifier in domain-specific context of reviews on different products, bag-of-words features are incorporated into the model.

The rest of this paper is structured as follows. Section 2 briefly reviews the recent advances in deep learning for opinion mining of consumer reviews. Section 3 outlines the proposed model. In Section 4, the benchmark dataset is introduced. Section 5 presents the results of the experiments in comparison with existing approaches. Section 6 presents future research directions and concludes the paper.

2 Deep Learning for Opinion Mining of Consumer Reviews – A Literature Review

This section reviews existing deep neural network (DNN)-based approaches to opinion mining of consumer reviews. As demonstrated in earlier studies, NNs outperform other traditional machine learning methods such as support vector machine (SVM) and Naïve Bayes (NB) in this task, irrespective of the context of balanced/unbalanced datasets [9]. However, the initial efforts in this domain relied on a traditional bag-of-words model that produced high-dimensional and sparse datasets. It should be noted that shallow NNs are not effective in handling sparse datasets [10]. By contrast, DNNs have the capacity to overcome this problem by capturing more complex features from the data. A DNN unsupervised learning approach was developed in [11] to show that word representation can be effectively learned by a stacked denoising autoencoder and that this representation can also be easily adapted to different review domains. To address the problem of scalability with the high-dimensional bag-of-words representation of the traditional autoencoders, a semisupervised autoencoder was developed for sentiment analysis in [12]. Supervision is introduced into the model via the loss function obtained from a linear classifier. Convolutional NNs (CNNs) were also employed to use the bag-of-words representation [3], which was also one of the first attempts to effectively use word order for opinion classification.

To further improve the performance of DNNs in opinion classification, vector representation models such as Word2Vec [13, 14] and Glove [15] were used to generate dense documents by reconstructing the linguistic context of the words. As a result, words that share a common context are located close to each other in the vector space, and the dimensionality of the space is reduced to several hundred word embeddings. CNNs and long short-term memory (LSTM) NNs were used to learn sentiment representation from word embeddings by [4]. In the next step, document representation was learned using gated recurrent units (GRUs). Different approaches for generating word embeddings were combined in a CNN model that outperformed SVM and NB. Another proposed CNN model integrates word embeddings with the representation of user text, thus incorporating user preferences [5]. Similarly, user and product information were utilized in an LSTM model with word and sentence attention [6]. To overcome the problem of the memory unit with long texts, a cached LSTM model was developed to capture the overall semantic representation [7]. Cross-domain sentiment classification represents another challenge in related literature. To learn a document representation that can be shared across domains, an end-to-end adversarial memory network was introduced in [16].

Recently, a cross-modality consistent regression model was employed to utilize three different CNN models with attention mechanisms, namely semantic, lexicon and sentiment representations. It was shown that sentiment and lexicon representations overcome the disadvantages of semantic embeddings in Twitter sentiment analysis [17]. Indeed, word embeddings used in previous studies ignore the sentiment polarity and sentiment intensity of the words and, hence, often combine words with different sentiment polarity. This may lead to misrepresentation of the documents in the context of

sentiment analysis. Moreover, the hybrid representation models combining word embeddings with the traditional bag-of-words representation may further improve the classification performance in related tasks due to highly domain-specific context [18, 19]. Product reviews from different domains is exactly such a task. Inspired by these observations, the original contribution of this study is the proposal of a DNN model integrating word embeddings, bag-of-words and a wide range of sentiment polarity and sentiment intensity features to overcome the problems of the above approaches. Notably, word-sentiment associations enable to obtain both the meaning and sentiment intensity of the words in the review representation. Deep feed-forward neural network (DFFNN) was employed in this integrated model to effectively handle the high-dimensional sparse bag-of-words representation [10].

3 DNN Model with Word-Sentiment Associations

The architecture of the proposed DNN with word-sentiment associations (DNN-WSA) model for opinion mining of consumer reviews is presented in Fig. 1. The DFFNN with two dense hidden layers was used to process the variety in the input features, including both the word-sentiment representation and the n -gram representation.

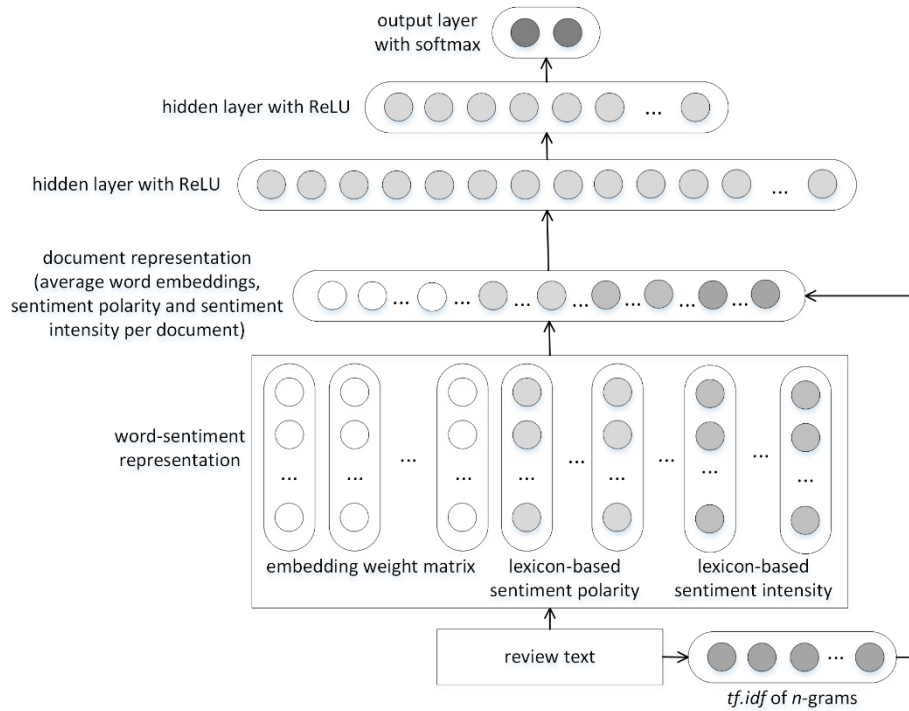


Fig. 1. The proposed DNN-WSA architecture for opinion mining of consumer reviews.

The word-sentiment representation is generated in two steps. First, word embeddings are trained using the Skip-Gram model because it is reportedly more effective than its competitors in exploiting the word context [13]. Second, the vocabulary obtained from the corpus of reviews is compared with several lexicons to append lexicon-based sentiment polarity and intensity.

To calculate the embedding weight matrix, the adapted embedding function is applied to each word w_t in the vocabulary. The embedding function is adapted for the sequence $W = \{w_1, w_2, \dots, w_t, \dots, w_T\}$ of training words so that the following objective function is maximized

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

where c represents the context window radius (how many surrounding words are considered), and $p(w_{t+1} | w_t)$ is the probability of the output word given the input words calculated using the hierarchical softmax algorithm

$$p(w_o | w_t) = \prod_{j=1}^{L(w)-1} \sigma(\llbracket n(w, j+1) = \text{ch}(n(w, j)) \rrbracket v_{n(w, j)}^T v_{w_t}), \quad (2)$$

where w_t and w_o are input and output words, respectively; v_w and v'_w denote the vector representations of the input and output words, respectively; $n(w, j)$ is the j -th node in the binary tree; $L(w)$ is the path length in the tree; $\text{ch}(n)$ represents a child node; and $\sigma(x)$ denotes a sigmoidal function, where if x is true, then $\llbracket x \rrbracket = 1$; otherwise $\llbracket x \rrbracket = -1$. To obtain the document representation for the next layer in the DNN-WSA architecture, the mean values of the vectors from the embedding weight matrix were calculated.

To complement the word-sentiment representation with the sentiment polarity and intensity, we used several predefined sentiment lexicons. To obtain a reliable sentiment assessment, it is suggested not to rely on a single lexicon [20]. Moreover, the combination of lexicon-based sentiment indicators overcomes the problem of susceptibility to indirect opinions typically present in the machine learning models. To calculate sentiment polarity, we used two handcrafted lexicons of positive and negative words: Bing Liu's opinion lexicon [21] and OpinionFinder [20]. One shortcoming of these lexicons is that equal weight is assigned to all words regardless of their sentiment intensity. To address this issue, we incorporated the sentiment intensity indicators obtained from the following lexicons with pre-trained sentiment strengths [20, 22]: S140, NRC Hashtag, AFINN and SentiWordNet. Thus, the overall positive and negative scores can be calculated for each lexicon. In addition, the combination of several lexicons ensures higher lexical coverage [20].

To obtain the n -gram representation, the weight of each n -gram is calculated as follows

$$\omega_{ij} = (1 + \log(tf_{ij})) \times \log(N/df_i), \quad (3)$$

where ω_{ij} denotes the weight of the i -th n -gram in the j -th document (review); $j = 1, 2, \dots, N$; and tf_{ij} and df_i represent term and document frequency, respectively. Thus, review length is considered, and a relatively higher weight is assigned to rare n -grams.

For further processing, the n -grams are ranked according to their weights, and top n -grams are selected to enter the document representation layer in the DNN architecture.

The next two hidden layers are used to process the complex relationship between the document representation and output sentiment positive/negative classes. To avoid overfitting and to make the training more effective, we used dropout regularization (dropout rate of 0.2 and 0.5 for the input and the two hidden layers, respectively) and ReLU (rectified linear units), respectively. The mini-batch gradient descent algorithm with $b = 100$ mini-batches, a learning rate of 0.1 and 1,000 iterations provided us with good and stable convergence behavior. Different numbers n_{h1} and n_{h2} of ReLU in the two hidden layers = $\{2^4, 2^5, 2^6, 2^7\}$ were tested to obtain the optimal architecture. As presented below, the best results were obtained for $n_{h1}=2^5$ and $n_{h2}=2^4$ neurons. Note that we also experimented with one hidden layer but without improvement. The objective function was represented by cross-entropy loss. The overall complexity of the proposed model can be expressed as $O(b \times I \times (m \times n_{h1} + n_{h1} \times n_{h2} + n_{h2} \times n_o))$, where I is the number of iterations; m denotes the number of features in the document representation layer; and n_{h1} , n_{h2} and n_o represent the numbers of neurons in the first and second hidden layers and the output layer, respectively.

4 Data and Preprocessing

For the experiments, a large enough Amazon dataset that is openly accessible at Kaggle¹ was used. The dataset, provided by Xiang Zhang, was originally used in [23] to classify opinions in consumer reviews using temporal CNNs with character-level features. The dataset was collected from the Stanford Network Analysis Project since 1994 [24], resulting in ~34 million reviews from ~6.6 million users on ~2.4 million products. The mean character length of the reviews was 764 (90.9 words). Extremely short and long reviews were discarded, and duplicates were removed. Users' rating scores were used to categorize the consumer reviews into positive and negative classes. More precisely, labels 1 and 2 were converted to negative opinion, and the scores of 4 and 5 were transformed to positive opinion. We used the testing data from the original dataset, represented by 130,000 samples from each score category. Overall, the dataset comprised 400,000 reviews evenly distributed into positive and negative opinion classes. Review title and review content were used in the dataset.

In the data pre-processing step, we performed tokenization (using the following delimiters: “.,:“(?!”), removal of stopwords (using the Rainbow list for noise reduction), and transformation to lowercase letters.

5 Experimental Results

The experiments were conducted on the Amazon dataset of 400,000 reviews. To learn word embeddings, we used the Skip-Gram model trained on the Amazon dataset. As

¹ <https://www.kaggle.com/bittlingmayer/amazonreviews>

shown in Fig. 2, we experimented with different settings of the model; the best performance was achieved with 200 word embeddings and context window radius $c = 5$. The Skip-Gram model was trained in the Deeplearning4j environment (distributed, open-source DNN library written for Java, compatible with Scala or Clojure and integrated with distributed computing frameworks Apache Spark and Hadoop). Regarding the bag-of-words representation, the top 1,000 n -grams (unigrams, bigrams and trigrams) were generated according to their $tf.idf$ (term frequency – inverse document frequency) weights in agreement with the previous literature [25]. To obtain the word-sentiment associations, the AffectiveTweets package was employed.

In our experiments, three evaluation measures were considered: accuracy (Acc), area under receiver operating characteristic curve (AUC), and F -score. To evaluate the performance of the proposed model, stratified 5-fold cross-validation was performed. The mean values and standard deviations are presented.

In a further set of experiments, we examined the effects of the used word representations. Fig. 3 shows that the DNN model using lexicon-based sentiment features had the worst performance. More precisely, the DNNs with n -gram and Skip-Gram features increased accuracy by 2.7% and 3.0%, respectively, compared with DNN-LexSent. DNN-BoW and DNN-SkipGram performed similarly in terms of all the evaluation measures. The DNN-WSA model performed best with a 3.8% increase in accuracy compared with the DNN-SkipGram model. Overall, the combination of the three word representations performed significantly better than the baseline models at the 5% significance level using the Wilcoxon signed rank test.

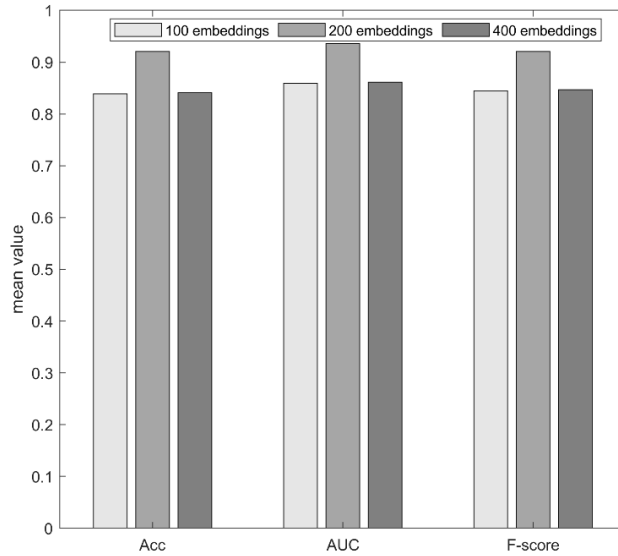


Fig. 2. The effect of the number of word embeddings on the performance of the DFFNN model with two hidden layers of $n_{h1}=2^5$ and $n_{h2}=2^4$ neurons.

To comprehensively evaluate the effectiveness of the DNN-WSA model, we compared its performance against the following existing models:

- Improved Naïve Bayes (INB-1) [26] accommodates the sentiment word using the SentiWordNet lexicon in the feature extraction component. Following [26], we extracted the unigrams, bigrams and sentiment patterns.
- Support vector machine with word sense disambiguation (SVM-WSD) [27] uses adverbs scored using the SentiWordNet lexicon as input features. Thus, positive and negative scores were assigned to adverbs, and SVM was trained using the LibLINEAR library. L2-regularized L2-loss SVM type was employed with cost parameter $C=1$.
- A multiple classifier model combining three baseline classifiers, namely NB, SVM and bagging (NB+SVM+Bagging) [28]. In agreement with the original study, we used unigrams as features and voting as the meta-classifier.
- LSTM [4] and CNN [4] were used to obtain the semantic sentence-level representation. Following [6], the dimension of hidden/cell states was set to 200, corresponding to the number of word embeddings. The CNN architecture comprised the convolutional layer with five filters of size 5 and a max pooling layer of size 4. For both models, the sentence representation was fixed and the number of words in the sentence corresponded to the review with maximum length. Document representation for both models was produced as the composition of sentence representation using GRUs. Stochastic gradient descent with Adam optimizer was used to train both models in the Deeplearning4j environment.

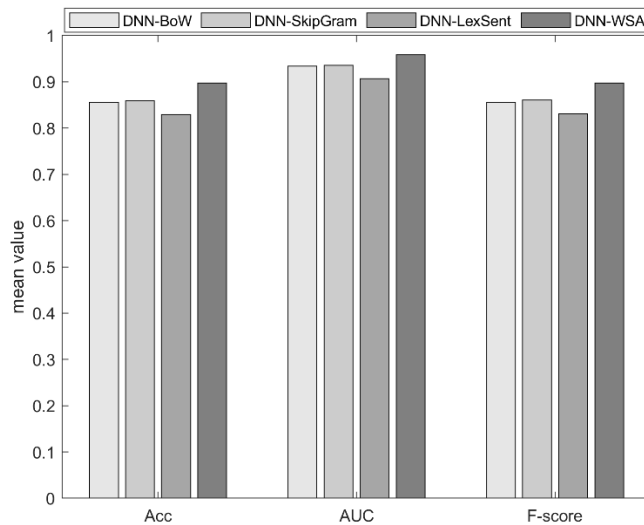


Fig. 3. The performance of FFDNN models using a) bag-of- n -grams (DNN-BoW), b) Skip-Gram word embeddings (DNN-SkipGram), c) lexicon-based sentiment polarity and intensity (DNN-LexSent), and d) all the word representations together (DNN-WSA). All the models were trained using two hidden layers with $n_{h1}=2^5$ and $n_{h2}=2^4$ neurons.

Table 1 shows the results of DNN-WSA in comparison with the above models. Note that the proposed model not only performed best in terms of all the used evaluation measures, but its performance was also significantly better at the 5% significance level using the Wilcoxon signed rank test, which demonstrates the effectiveness of the proposed model. SVM-WSD also performed well in terms of accuracy, especially when considering the computational time.

In this study, we adopted the testing time criterion - as suggested in related studies [19] - to show the real-time capacity of consumer review classifiers. The proposed DNN-WSA model performed the worst regarding time efficiency, but it can still be considered time efficient with approximately 7,700 reviews classified per second. Recall that the key determinants of the overall complexity are the numbers of iterations and features in the DNN model. Therefore, better time efficiency can be expected with the decrease in the number of n -grams. Overall, the DNN-WSA model performed well for both opinion categories, as indicated by the high value of AUC. The other two DNN models, LSTM and CNN, also performed well regarding AUC. Additionally, the high value of the F -score for DNN-WSA indicates a balanced performance in terms of precision (0.896 on average) and recall (0.899).

Table 1. Results of the experiments.

Model	Acc [%]	AUC	F -score	Testing time [s]
INB-1 [26]	78.84±0.70	0.833±0.008	0.796±0.005	5.862±0.363
SVM-WSD [27]	85.61±2.67	0.856±0.027	0.862±0.022	0.070±0.011
NB+SVM+Bagging [28]	83.10±0.68	0.906±0.006	0.833±0.006	0.105±0.011
LSTM [4]	84.05±0.28	0.917±0.003	0.841±0.002	2.042±0.128
CNN [4]	84.29±0.17	0.921±0.001	0.844±0.003	8.139±0.286
DNN-WSA (this study)	89.70±0.92	0.959±0.011	0.897±0.009	10.398±0.226

Notes: The best results are in bold. The experiments were conducted using AMD Opteron 6180 SE 2.50 GHz with twelve cores/threads and 256 GB RAM.

6 Conclusion

In this study, we proposed an efficient DNN model integrating word-sentiment associations for the opinion mining of consumer reviews. We proved the model’s performance improvement compared with baseline word representations by conducting extensive experiments on the Amazon dataset. We compared the proposed DNN-WSA model with several existing approaches, including both DNNs and other machine learning methods. Hence, the effectiveness of the proposed model was demonstrated. The results of the experiments suggest that word-sentiment associations might be more effective than word representation based on word embeddings only. Integrating the word-sentiment associations with n -gram representation provides further improvement. However, such a word representation model leads to a partly sparse dataset, which necessitates further requirements for the opinion mining machine learning methods. We showed that the proposed DNN model can handle such a word representation model.

In future research, a more thorough analysis can be performed by investigating the word-sentiment associations at the entity/aspect level. One of the limitations of the proposed model is that only local features were captured. Therefore, alternative DNN models with attention mechanisms could be considered to overcome this limitation. A cross-domain modification of the model is another problem that needs to be addressed. The n -gram feature extraction used in this study does not consider the semantic similarity or the discriminative ability of words. Therefore, enhanced n -gram representations [29] are recommended to reduce the dimensionality and sparsity of the data. The application of an effective feature selection method may also lead to lower computational complexity and improved time efficiency [30]. Alternative embedding-based schemes can also be utilized [31].

Acknowledgments. This article was supported by the scientific research project of the Czech Sciences Foundation Grant No. 19-15498S and by the Student Grant Competition SGS_2020.

References

1. Liu, B.: Sentiment analysis: Mining opinions, sentiments, and emotions. The Cambridge University Press (2015).
2. Zhang, L., Shuai W., Liu, B.: Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8(4), e1253 (2018).
3. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. In: Proc. of the Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 103–112 (2015).
4. Tang, D., Qin, B., Liu, T.: Document modelling with gated recurrent neural network for sentiment classification. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015).
5. Tang, D., Qin, B., Liu, T.: Learning semantic representations of users and products for document level sentiment classification. In: Proc. of the Annual Meeting of the Association for Computational Linguistics, pp. 1014–1023 (2015).
6. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing, pp. 1650–1659 (2016).
7. Xu, J., Chen, D., Qiu, X., Huang, X.: Cached long short-term memory neural networks for document-level sentiment classification. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing, pp. 1660–1669 (2016).
8. Do, H. H., Prasad, P. W. C., Maag, A., Alsadoon, A.: Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems with Applications* 118, 272–299 (2019).
9. Moraes, R., Valiati, J. F., Neto, W. P.: Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications* 40, 621–633 (2013).
10. Barushka, A., Hajek, P.: Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. *Applied Intelligence* 48(10), 3538–3556 (2018).

11. Glorot, X., Bordes, A., Bengio, Y.: Domain adaption for large-scale sentiment classification: A deep learning approach. In: Proc. of the 28th Int. Conf. on Machine Learning, ICML, pp. 513–520 (2011).
12. Zhai, S., Zhang, Z. M.: Semisupervised autoencoder for sentiment analysis. In: Proc. of AAAI Conf. on Artificial Intelligence, AAAI, pp. 1394–1400 (2016).
13. 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, pp. 3111–3119. NIPS, vol. 26 (2013).
14. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, vol. 32, pp. 1188–1196. JMLR (2014).
15. Pennington, J., Socher, R., Manning, C. D.: GloVe: Global vectors for word representation. In: Proc. of the Conf. on Empirical Methods on Natural Language Processing, pp. 1532–1543 (2014).
16. Li, Z., Zhang, Y., Wei, Y., Wu, Y., Yang, Q.: End-to-end adversarial memory network for cross-domain sentiment classification. In: Proc. of the Int. Joint Conference on Artificial Intelligence, pp. 2237–2243 (2017).
17. Zhang, Z., Zou, Y., Gan, C.: Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. *Neurocomputing* 275, 1407–1415 (2018).
18. Sun, C., Du, Q., Tian, G.: Exploiting product related review features for fake review detection. *Mathematical Problems in Engineering*, 1–7 (2016).
19. Hajek, P., Barushka, A., Munk, M.: Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Computing and Applications*, 1–16, (2020).
20. Bravo-Marquez, F., Frank, E., Mohammad, S. M., Pfahringer, B.: Determining word-emotion associations from tweets by multi-label classification. In: 2016 IEEE/WIC/ACM Int. Conf. on Web Intelligence, pp. 536–539. IEEE (2016).
21. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004).
22. Kiritchenko, S., Zhu, X., Mohammad, S. M.: Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50, 723–762 (2014).
23. Zhang, X., LeCun, Y.: Text understanding from scratch. arXiv preprint arXiv:1502.01710 (2015).
24. McAuley, J., Leskovec, J.: Hidden factors and hidden topics: Understanding rating dimensions with review text. In: Proc. of the 7th ACM Conf. on Recommender Systems, pp. 165–172 (2013).
25. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg!. In: Fifth Int. AAAI Conf. on Weblogs and Social Media, pp. 538–541 (2011).
26. Kang, H., Yoo, S. J., Han, D.: Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications* 39(5), 6000–6010 (2012).
27. Kausar, S., Huahu, X., Shabir, M. Y., Ahmad, W.: A sentiment polarity categorization technique for online product reviews. *IEEE Access* 8, 3594–3605 (2019).
28. Catal, C., Nangir, M.: A sentiment classification model based on multiple classifiers. *Applied Soft Computing* 50, 135–141 (2017).
29. Chen, X., Xue, Y., Zhao, H., Lu, X., Hu, X., Ma, Z.: A novel feature extraction methodology for sentiment analysis of product reviews. *Neural Computing and Applications* 31(10), 6625–6642 (2019).

30. Barushka, A., Hajek, P.: Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, 1–19 (2020).
31. Onan, A.: Deep learning based sentiment analysis on product reviews on Twitter. In: *Int. Conf. on Big Data Innovations and Applications*, pp. 80–91. Springer, Cham (2019).