ICIC Express Letters
Volume **16**, Number **1**, January **2022**

ICIC International ©2022 ISSN 1881-803X
pp. 101–108

RECURRENT NEURAL NETWORK DOCUMENT EMBEDDING METHOD FOR ADVERSE DRUG REACTION DETECTION FROM MEDICAL REVIEWS

Ahmed Sabah Ahmed AL-Jumaili[1,*] and Huda Kadhim Tayyeh[2]

[1]Department of Bioinformatics (BI)
College of Biomedical Informatics
[2]Department of Informatics Systems Management (ISM)
College of Business Informatics
University of Information Technology and Communications
Al-Nidhal St., Baghdad 00964, Iraq
[*]Corresponding author: asabahj@uoitc.edu.iq; haljobori@uoitc.edu.iq

Received April 2021; accepted July 2021

Abstract. *Detecting side-effects or Adverse Drug Reaction (ADR) from social and medical reviews is a new research that caught several researchers' attention. A wide range of opportunities can be achieved through extracting such entities from evaluating medical products to improve customer satisfaction. In the previous studies, extracting ADRs was depicted through a common approach that is intended to exploit informative keywords that occasionally appeared along with the ADRs. However, such keywords require massive manual annotations. In addition, this keyword utilization seems an unsuccessful approach with non-formal written reviews. This paper proposes a document embedding method using Recurrent Neural Network (RNN). The proposed method is intended to generate a generalized embedding vector for each medical review that can articulate its context. Three classifiers including Logistic Regression (LR), Support Vector Machine (SVM), and Naïve Bayes (NB) have been utilized to predict the ADR presence based on generated embedding vectors by the proposed RNN. Using a benchmark dataset of medical reviews, the proposed document embedding demonstrated an enhancement in the F-measure results of NB and SVM classifiers over a baseline study. This enhancement reflects the efficacy of using document embedding to generalize the contextual features without using any external knowledge source.*
Keywords: Adverse drug reaction detection, Recurrent neural network, Document embedding, Logistic regression, Support vector machine, Naïve Bayes

1. **Introduction.** With the exponential growth of web 2.0, several domains of industries became much interested in growing their online business [1]. Enormous online businesses have become available nowadays where products are being shown and consumers are enabled to comment on such products [2]. The review section provides valuable information where mining consumers' feedbacks would represent a great opportunity toward determining user requirements. This would lead to a better understanding of user satisfaction and would open the door for further improvement [3].

In particular, the medical business domain has taken the optimal advantage of these reviews to investigate clinical consequences [4]. Assume an online drug store where users can purchase and comment on a particular medicine. The review section would contain plenty of significant information that previously was hard to be obtained. In particular, the side effects or any adverse reaction of a drug can be identified through people who experienced taking the exact drug [5,6]. This has not been depicted easily in the past in

DOI: 10.24507/icicel.16.01.101

which determining the Adverse Drug Reaction (ADR) was requiring much more sophisticated experiments with a large number of volunteers and a long time of execution. Now, we can have further evaluation of specific drugs when a wide range of people from different medical situations and medical backgrounds purchase, take and review a particular drug [7].

Therefore, researchers tend to exploit Artificial Intelligence (AI) for mining medical reviews to automatically detect ADRs. The process of training the machine was depicted in two paradigms: first by using rule-based algorithms [8], and second, using machine learning techniques [9]. The first paradigm aims at preparing a handcrafted set of rules and conditions that simulate the occurrence of ADR. Whereas, the second paradigm aims at training machine learning algorithms on an example set of annotated text to detect the future occurrence of ADR. In particular, the previous studies have widely examined both rule-based and machine learning techniques through an approach called Trigger Terms (TTs). Such an approach is intended to exploit informative keywords that have considerable adjacent occurrence with the ADR terms. For instance, the ADR "*headache*" is probably occurring with keywords such as "*have*", "*suffer*", and "*got*". In the literature, TTs are stored in a predefined list and represented through different N-gram adjacent terms to help both machine learning techniques and rule-based to predict the occurrence of ADRs. Relying on TTs would be insufficient especially when regular users express their feedback within simple and non-formal written language. Therefore, it is necessary to consider a method that does not rely on TTs where semantics and contexts can be interpreted even in a non-formal manner.

This paper aims to take the advantage of recent Natural Language Processing (NLP) techniques particularly the embedding vectors generated by specific architectures of Neural Network (NN). In specific, a document embedding method based on Recurrent Neural Network (RNN) will be proposed in this study. After generating the embedding, multiple machine learning classification methods will be used to predict the occurrence of ADRs including LR, SVM, and NB. Results of classification showed that the proposed document embedding using RNN has contributed toward improving the accuracy of both NB and SVM compared to the traditional TTs exploitation by a baseline study.

The structure of the paper can be seen as: summary of related work depicted in Section 2, an illustration of the proposed RNN document embedding represented in Section 3, the experimental results and their analysis addressed in Section 4, and lastly, the final conclusion is depicted in Section 5.

2. **Related Work.** The literature has depicted different ways of detecting ADRs, for example, Yu et al. [10] proposed an approach that utilizes N-gram compounds in which the unigram (i.e., single terms) and bigram (i.e., sequence of adjacent terms) are being exploited to detect the occurrence of ADRs. The N-gram approach has been utilized by two classifiers including NB and SVM to predict the occurrence of ADRs.

On the other hand, Pain et al. [11] utilized an approach known as Trigger Terms (TTs) which aims at exploiting keywords that frequently occur with ADRs. The authors have examined a collected data from Twitter where the hashtags represent the TTs. For predicting the ADRs occurrence, the authors have utilized the SVM classifier. In the same regard, Ebrahimi et al. [8] incorporate the TTs approach with a semantic source called Unified Medical Language System (UMLS) to enrich the TTs with new keywords for the sake of extracting ADRs. For the prediction task, the authors used two approaches including SVM classifier and rule-based classifier. Additionally, Plachouras et al. [12] have also represented an attempt to enrich the TTs through medical Gazetteers for extracting ADRs from Twitter collected data. For predicting the ADRs, an SVM has been used.

Afterward, most of the related works attempted to extend or enrich the TTs for the ADR extraction task. For example, Moh et al. [13] utilized a semantic knowledge source of SentiWordNet for expanding the list of TTs. In addition, Kiritchenko et al. [14] utilized domain-specific keywords for enriching the TTs. Whereas, Wu et al. [15] exploited specific abbreviations to enrich the TTs. Furthermore, Yousef et al. [9] utilized syntactic patterns along with Mutual Information (MI) to extract the most frequent patterns that might occur with the ADRs. The authors then used multiple classification methods including LR, SVM, and NB to predict the occurrence of ADRs based on the extracted patterns. Similarly, Zhang et al. [16] exploited the syntactic aspect to generate pairs of TTs for the sake of ADR extraction. Pandya et al. [17] have exploited the Electronic Health Records (EHR) from emergency departments where plenty of ADRs occasionally occurred. Such exploitation of EHR was aimed to generate new TTs.

From the literature, one could notice that TTs have been a major approach that has been used to extract ADRs. Relying on TTs would be insufficient especially when regular users express their feedback within simple and non-formal written language. Therefore, it is necessary to consider a method that does not rely on TTs where semantics and contexts can be interpreted even in a non-formal manner.

3. **Proposed Method.** As depicted in Figure 1, the proposed method's framework is composed of three phases. The first phase addresses the dataset that will be used in the experiments which consist of medical reviews collected from medical online shopping stores. The second phase represents the core contribution of this study where the proposed RNN document embedding will take a place. In particular, RNN will process the text documents of the medical reviews to generate embedding vectors where D refers to the documents and f denotes the embedding feature with a dimension of $m \times n$. Consequentially, the third phase represents the actual prediction of the ADRs occurrence based on the generated document embedding vectors. For this purpose, three machine learning classification methods are being used including LR, SVM, and NB.

3.1. **Dataset.** This study utilizes a dataset consisting of medical reviews which have been collected by Yates and Goharian [18]. The dataset focuses on the review section of multiple drugs online shopping stores and drug inquiries websites such as askapatient.com,

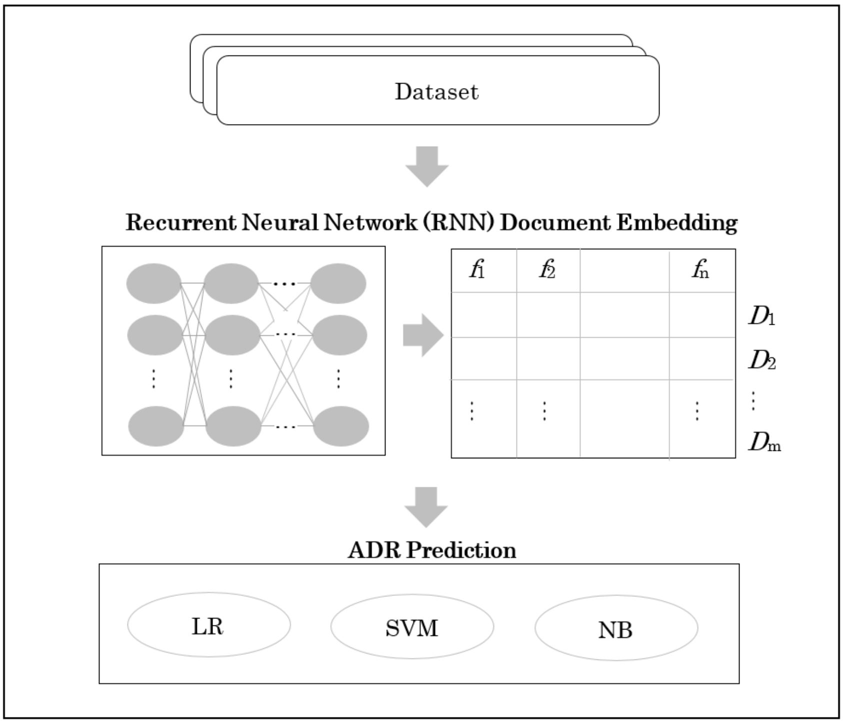

FIGURE 1. Proposed method's framework

drugs.com, and drugratingz.com. The dataset is composed of independent documents where each document denotes a review by a user where he/she described his/her experience and feedback about a particular medical product. Every document has been annotated by '0' or '1' where the first class refers to the absence of ADR, while the latter class refers to the presence of ADR. Table 1 depicts a sample of the dataset.

TABLE 1. Dataset's sample

Review documents	Text	Class label	ADR
D_1	Joint pain in fingers	1	"Pain"
D_2	Some relief in joint	0	–
D_3	Some wrist and fingers pain	1	"Pain"

As depicted in Table 1, some documents contain ADR such as D_1 and D_3 where the word '*pain*' has appeared; therefore, the documents have been annotated as '1'. While D_2 states a fact without mentioning any ADR; therefore, it has been annotated as '0'.

3.2. **The proposed RNN document embedding.** Before the RNN process and similar to the traditional Word2Vec architecture for word embedding, the documents will undergo a one-hot encoding that aims at obtaining initial vectors of such textual documents [19]. One-hot encoding is intended to address the distinctive terms within the text documents and represent them as columns and rows where the population of such a matrix are the correspondences among terms which leads to a sparse matrix. Let us assume the three documents in Table 1, and the one-hot encoding of terms can be depicted in Table 2.

TABLE 2. One-hot encoding

	joint	pain	in	fingers	some	relief	wrist	and
joint	1	0	0	0	0	0	0	0
pain	0	1	0	0	0	0	0	0
in	0	0	1	0	0	0	0	0
fingers	0	0	0	1	0	0	0	0
some	0	0	0	0	1	0	0	0
relief	0	0	0	0	0	1	0	0
wrist	0	0	0	0	0	0	1	0
and	0	0	0	0	0	0	0	1

In the traditional Word2Vec, the resulted sparse matrix will be processed via a feed-forward neural network architecture to predict a target term [20]. However, in the RNN document embedding architecture, the document itself will undergo a one-hot encoding, too. Table 3 depicts the one-hot encoding for documents.

TABLE 3. One-hot encoding for documents

	joint	pain	in	fingers	some	relief	wrist	and
D_1	1	1	1	1	0	0	0	0
D_2	1	0	1	0	1	1	0	0
D_3	0	1	0	1	1	0	1	1

As shown in Table 3, every document has been addressed in terms of containing the distinctive terms. Hence, each document has now an initial vector. Now, both terms' vectors and documents' vectors will be processed via the RNN architecture. Figure 2 depicts an example of processing the first document in Table 1 where the vectors of terms

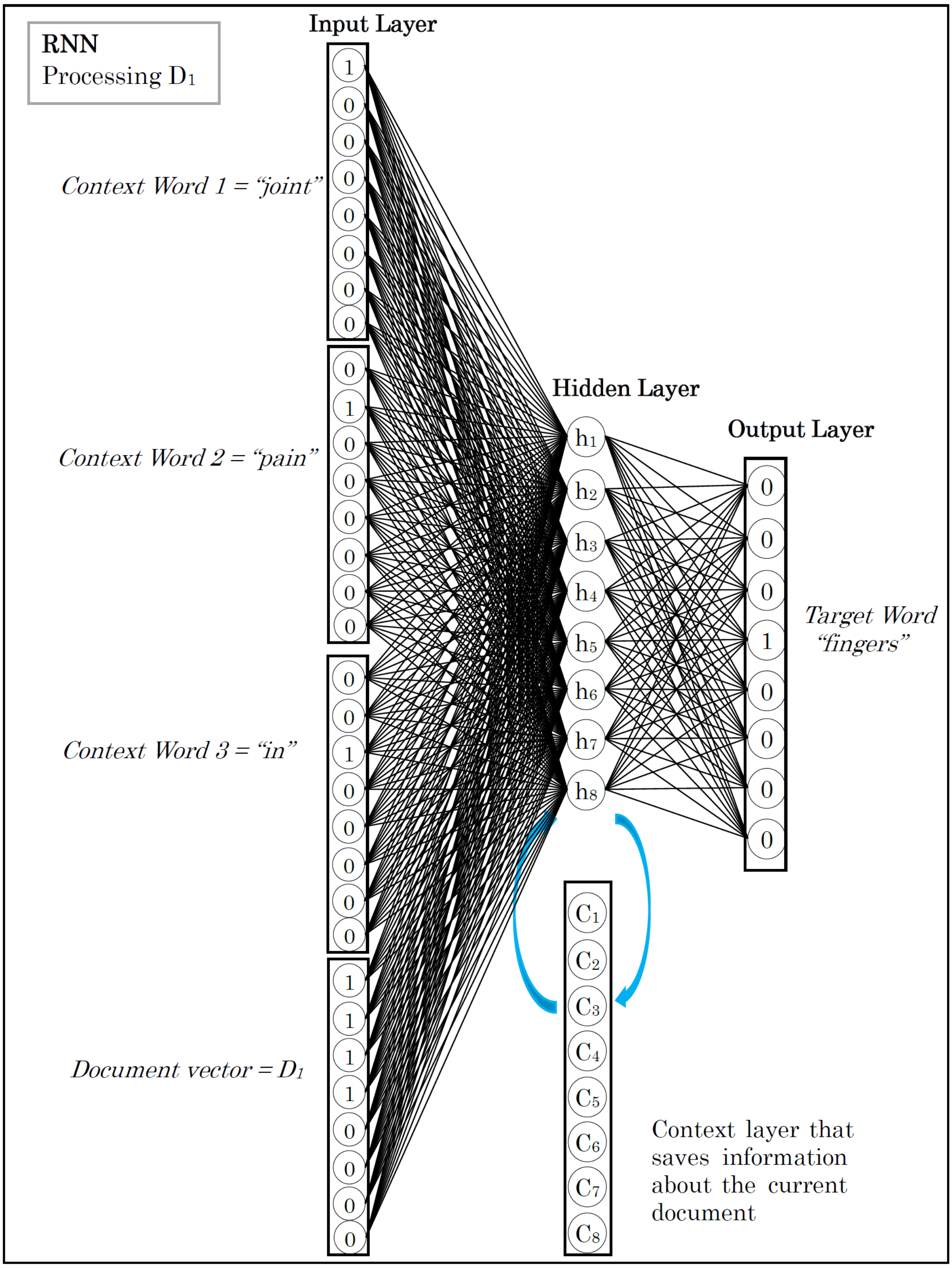

FIGURE 2. RNN document embedding architecture

(Table 2) inside such a document will be fed to the RNN architecture along with the document vector itself (Table 2) to get its embedding.

Similar to the traditional neural network architecture, RNN is composed of three layers including the input, hidden, and output layers as shown in Figure 2. Additionally, RNN contains a special layer called the context layer or memory layer [21]. This layer stores information about the current input in which the next input can use such information. Now, as shown in Figure 2, the vector of each term ('*joint*', '*pain*', '*in*') within D_1 (except the target term '*fingers*') will be processed through the input layer as context terms. This has been accompanied by the document vector itself. The output layer denotes the vector of the target term ('*fingers*') where the aim is to predict its vector. Typical to the learning within any neural network, the weights will be initiated randomly and adjusted through Backpropagation learning. Yet, the output of the hidden layer will be fed to the context layer in which the output of such context layer will be recursively back to the hidden layer. This is to avoid any possible vanishing or exploding gradients [22,23]. Lastly, once the architecture starts to predict outputs with reasonable accuracy, the hidden neurons' values will resemble the document embedding of D_1. Table 4 depicts the hyperparameters selected for the RNN architecture. The values of hyperparameters in Table 4 have been brought from the study of Lau and Baldwin [24] where these values demonstrated better performance compared to others. To implement the RNN document embedding parameters, the Python library of Doc2Vec has been brought from the Python package of Gensim [25].

Table 4. RNN document embedding hyperparameters

Parameter	Value
Vector dimension	300
Number of epochs	100
Alpha	0.025
Minimum count	5

3.3. **ADR prediction.** Once the document embedding vectors for all the medical review documents are being generated by the proposed RNN, multiple machine learning classifiers including LR, SVM, and NB will be trained on such vectors and tested by predicting the presence of ADR. These classifiers have been selected to facilitate consistent comparison against the baseline of Yousef et al. [9]. In addition, the same training and the testing splitting ratio of 70%-30% has been followed.

4. **Results.** Once each classifier finishes the prediction, the common information retrieval evaluation metrics including precision, recall, and F-measure will be used to assess the results of the prediction. To compute the precision, the following equation will be used [26,27]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

where TP denotes the True Positives, which corresponds to the case when the classifier's prediction matches the actual class. Whereas, FP denotes the False Positives, which corresponds to the case when the classifier's prediction tells that there is an ADR presence within the document, while it does not. On the other hand, to compute the recall, the following equation will be used

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{TN}} \tag{2}$$

where TN denotes the True Negatives, which corresponds to the case when the classifier's prediction tells that there is no presence of ADR within the document, while it does. Lastly, the F-measure denotes the harmony between precision and recall and it can be calculated through the following equation:

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

F-measure is considered to be the overall accuracy of the prediction; therefore, the results will be represented by it. In addition, a comparison is accommodated against the baseline of Yousef et al. [9] who used Trigger Terms (TTs) via Term Frequency Inverse Document Frequency (TFIDF) through the three classifiers. Table 5 depicts such results.

Table 5. Comparison results of F-measure between the baseline and the proposed methods

Classifiers	Baseline [9] (TTs via TFIDF)	Proposed (RNN document embedding)
NB	61%	**72%**
LR	**68%**	64%
SVM	69%	**71%**

As depicted in Table 5, results of ADR prediction for NB showed an outperformance obtained by the proposed RNN document embedding where the F-measure was 72% compared to 61% acquired by the baseline. However, for the LR classifier, the results of the F-measure obtained by the proposed method showed no enhancement. Instead, it shows lower F-measure of 64% compared to the F-measure of 68% obtained by the baseline study. This shortcoming occurs because LR can take the advantage of numerous features, and since the baseline was representing features based on TFIDF, the number of features would be significantly larger than the features of the proposed RNN embedding which has a dimension of 300. Yet, the results F-measure for SVM classifier showed superiority by the proposed RNN document embedding where the F-measure was 71% compared to 69% acquired by the baseline. The overall results prove that the proposed document embedding can generalize the contextual features of the medical review without the need for any predefined list of trigger terms or knowledge sources for the process of ADR detection.

5. **Conclusions.** This paper proposed a document embedding method through RNN architecture for detecting ADR from medical reviews. Multiple classifiers have been utilized to predict the presence of ADR including LR, SVM, and NB. Results of prediction showed that the proposed document embedding was able to improve the accuracy of detection over a baseline study. In the future direction, incorporating engineered features with the embedding vectors would contribute toward improving the prediction accuracy.

REFERENCES

[1] B. Alshaikhdeeb and K. Ahmad, Integrating correlation clustering and agglomerative hierarchical clustering for holistic schema matching, *Journal of Computer Science*, vol.11, no.3, pp.484-489, 2015.

[2] F. Lafontaine and J. Sivadasan, The recent evolution of physical retail markets: Online retailing, big box stores, and the rise of restaurants, in *The Role of Innovation and Entrepreneurship in Economic Growth*, A. Chatterji, J. Lerner, S. Stern and M. J. Andrews (eds.), University of Chicago Press, 2020.

[3] M. Fan, C. Feng, L. Guo, M. Sun and P. Li, Product-aware helpfulness prediction of online reviews, *The World Wide Web Conference (WWW'19)*, San Francisco, CA, USA, pp.2715-2721, 2019.

[4] B. Audeh, F. Bellet, M.-N. Beyens, A. Lillo-Le Louët and C. Bousquet, Use of social media for pharmacovigilance activities: Key findings and recommendations from the Vigi4Med project, *Drug Safety*, vol.43, no.9, pp.835-851, 2020.

[5] B. Biseda and K. Mo, Enhancing pharmacovigilance with drug reviews and social media, *arXiv Preprint*, arXiv: 2004.08731, 2020.

[6] B. Alshaikhdeeb and K. Ahmad, Feature selection for chemical compound extraction using wrapper approach with Naive Bayes classifier, *The 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, Langkawi, Malaysia, pp.1-6, 2017.

[7] Z. Li, Z. Yang, L. Luo, Y. Xiang and H. Lin, Exploiting adversarial transfer learning for adverse drug reaction detection from texts, *Journal of Biomedical Informatics*, vol.106, 2020.

[8] M. Ebrahimi, A. H. Yazdavar, N. Salim and S. Eltyeb, Recognition of side effects as implicit-opinion words in drug reviews, *Online Information Review*, vol.40, no.7, pp.1018-1032, 2016.

[9] R. N. Yousef, S. Tiun and N. Omar, Extended trigger terms for extracting adverse drug reactions in social media texts, *Journal of Computer Science*, vol.15, no.6, pp.873-879, 2019.

[10] F. Yu, M. Moh and T.-S. Moh, Towards extracting drug-effect relation from Twitter: A supervised learning approach, *2016 IEEE the 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, New York, USA, pp.339-344, 2016.

[11] J. Pain, J. Levacher, A. Quinqunel and A. Belz, Analysis of Twitter data for postmarketing surveillance in pharmacovigilance, *Proc. of the 2nd Workshop on Noisy User-Generated Text*, Osaka, Japan, pp.94-101, 2016.

[12] V. Plachouras, J. L. Leidner and A. G. Garrow, Quantifying self-reported adverse drug events on Twitter: Signal and topic analysis, *Proc. of the 7th 2016 International Conference on Social Media & Society*, New York, United States, pp.1-10, 2016.

[13] M. Moh, T.-S. Moh, Y. Peng and L. Wu, On adverse drug event extractions using Twitter sentiment analysis, *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol.6, no.1, p.18, 2017.

[14] S. Kiritchenko, S. M. Mohammad, J. Morin and B. de Bruijn, NRC-Canada at SMM4H shared task: Classifying tweets mentioning adverse drug reactions and medication intake, *Proc. of the Social Media Mining for Health Applications Workshop at AMIA-2017*, Washington, D.C., USA, pp.1-11, 2018.

[15] C. Wu, F. Wu, Z. Yuan, J. Liu, Y. Huang and X. Xie, MSA: Jointly detecting drug name and adverse drug reaction mentioning tweets with multi-head self-attention, *Proc. of the 12th ACM International Conference on Web Search and Data Mining*, New York, United States, pp.33-41, 2019.

[16] Y. Zhang, S. Cui and H. Gao, Adverse drug reaction detection on social media with deep linguistic features, *Journal of Biomedical Informatics*, vol.106, 2020.

[17] A. D. Pandya, K. Patel, D. Rana, S. D. Gupta, S. D. Malhotra and P. Patel, Global trigger tool: Proficient adverse drug reaction autodetection method in critical care patient units, *Indian Journal of Critical Care Medicine: Peer-Reviewed, Official Publication of Indian Society of Critical Care Medicine*, vol.24, no.3, pp.172-178, 2020.

[18] A. Yates and N. Goharian, ADRTrace: Detecting expected and unexpected adverse drug reactions from user reviews on social media sites, *European Conference on Information Retrieval*, Berlin, Heidelberg, Germany, pp.816-819, 2013.

[19] C. Seger, *An Investigation of Categorical Variable Encoding Techniques in Machine Learning: Binary Versus One-Hot and Feature Hashing*, Bachelor Thesis, School of Electrical Engineering and Computer Science (EECS), Royal Institute of Technology, Stockholm, Sweden, 2018.

[20] Y. Goldberg and O. Levy, Word2Vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method, *arXiv Preprint*, arXiv: 1402.3722, 2014.

[21] W. Zaremba, I. Sutskever and O. Vinyals, Recurrent neural network regularization, *arXiv Preprint*, arXiv: 1409.2329, 2014.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, vol.15, no.1, pp.1929-1958, 2014.

[23] S. Rikukawa, H. Mori and T. Harada, Recurrent neural network based stock price prediction using multiple stock brands, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.1093-1099, 2020.

[24] J. H. Lau and T. Baldwin, An empirical evaluation of Doc2Vec with practical insights into document embedding generation, *Proc. of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany, pp.78-86, 2016.

[25] B. Srinivasa-Desikan, *Natural Language Processing and Computational Linguistics: A Practical Guide to Text Analysis with Python, Gensim, spaCy, and Keras*, Packt Publishing Ltd., 2018.

[26] A. S. A. AL-Jumaili and H. K. Tayyeh, A hybrid method of linguistic and statistical features for arabic sentiment analysis, *Baghdad Science Journal*, vol.17, no.1, pp.385-390, 2020.

[27] H. K. Tayyeh and A. S. A. Al-Jumaili, Classifying confidential data using SVM for efficient cloud query processing, *Journal of Telecommunication Computing Electronics and Control (TELKOMNIKA)*, vol.17, no.6, pp.3155-3160, 2019.