Utilizing Word Index Approach with LSTM Architecture for Extracting Adverse Drug Reaction from Medical Reviews

Asmaa J. M. Alshaikhdeeb * and Yu-N Cheah

School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia; Email: yncheah@usm.my (Y.N.C.)
*Correspondence: ashaikhdeeb@student.usm.my (A.J.M.A.)

***Abstract*—Adverse Drug Reaction (ADR) detection from social reviews refers to the task of exploring medical online stores and social reviews for extracting any mention of abnormal reactions that occur after consuming a particular medical product by the consumers themselves. A variety of approaches have been used for extracting ADR from social/medical reviews. These approaches include machine learning, dictionary-based and statistical approaches. Yet, these approaches showed either a high dependency on using an external knowledge source for ADR detection or relying on domain-dependent mechanisms that might lose contextual information. This study aims to propose word sequencing with Long Short-Term Memory (LSTM) architecture. A benchmark dataset of MedSyn has been used in the experiments. Then, a word indexing, mapping, and padding method have been used to represent the words within the reviews as fixed sequences. Such sequences have been fed into the LSTM consequentially. Experimental results showed that the proposed LSTM could achieve an F1 score of up to 92%. Comparing such a finding to the baseline studies reveals the superiority of LSTM. The demonstration of the efficacy of the proposed method has taken different forms including the examination of word indexing with different classifiers, the examination of different features with LSTM, and through the comparison against the baseline studies.**

***Keywords*—adverse drug reaction extraction, word, mapping, word indexing, word embedding, long short-term memory**

I. Introduction

The art, business, and marketing field have been affected by the development and spreading rapidly of social media. The medical field is also one of the significant areas that have been engaged in social media where customers or users can provide and share their opinion about the medical product [1]. The peculiarities of the medical field constitute an obstacle to the application of existing systems for data mining of the opinion of the user in the medical area. While opinion mining for the medical text will assist in enhancing decision-making, providing answers to medical questions, and pharmacovigilance.

The exponential growth in E-commerce has led to the emergence of various online stores in the last decade. Numerous global and local online stores are available nowadays for consumers who are seeking online purchasing. A wide range of products and services are being purchased over the internet where online stores offer a unique experience of searching, surfing, and exploring catalogs of tremendous products [2]. Within these online stores, a variety of functionalities are being obtained such as subscriptions, online payments, promotions, and others. Among these functionalities, a distinctive feature has been attained which is represented by the reviews typed by the consumers themselves. This feature enables the regular consumer to express his/her feedback toward the product or the service. This space of social reviews opens a wide door for mining information whether by the research community, product/service providers, or the platform of the online store itself [3]. This process of mining the reviews aims at exploring opportunities to enhance the products/services, improving customer satisfaction, and identifying particular behavioral trends or patterns.

Considering the online stores that sell medical products, the opportunities that can be achieved through mining the reviews would seem bigger [4]. This can be depicted by identifying medical entities such as symptoms, syndromes, or side effects. Hence, a task known as Adverse Drug Reaction (ADR) detection from social reviews has emerged [5]. This task aims at exploring the social reviews collected from medical online stores to extract any mentioning of abnormal reactions that occur after consuming a particular medical product by the consumers themselves. The identification of ADR mentioned within the medical reviews can help companies to re-evaluate their medicines by discovering new drug interactions. Yet, the manual identification of ADRs would seem a tedious and time-consuming task.

The extraction process has been depicted by employing Natural Language Processing (NLP) and Artificial Intelligence (AI) techniques. The common NLP approach that has been widely examined by researchers for extracting ADR is the Trigger Terms (TTs). These trigger terms are the common frequent keywords that are occasionally accompanied by the ADRs [6]. For example,

Manuscript received August 30, 2022; revised October 26, 2022; accepted December 1, 2022; published June 16, 2023.

doi: 10.12720/jait.14.3.543-549

the phrase "I had a heavy pain last night" contains an ADR of 'pain' and trigger terms of 'had' or 'heavy'.

The literature on ADR extraction showed a high dependency on the trigger terms utilization within the ADR detection task [7, 8]. The utilization of trigger terms represents different challenging issues. First, it is necessary to use an external knowledge source for identifying TTs [9]. In addition, TTs cannot be ranged in a definite list or glossary since they are varying based on the background of the writer [6]. In particular, dealing with social reviews would contain feedback written by regular users where non-formal, ambiguous, and abbreviations terms occasionally occur. Therefore, limiting ADR detection through only TTs might not be a generic solution.

Recently, multiple research efforts have taken the advantage of statistical techniques such as the Latent Dirichlet Allocation (LDA) [10] and the Latent Semantic Analysis (LSA) [11] for the ADR extraction task. These techniques aim at exploiting context similarity through statistical analysis. However, such techniques are domain-dependent which means that they are significantly impacted by the domain terms provided during the training. Taking the advantage of recent deep learning architecture along with the modern text representation, a generic approach can be utilized to identify the ADR occurrences without the need for either external knowledge sources or statistical analysis.

Therefore, this paper aims to propose a generic feature of word indexing to map words within social reviews into indices. Consequentially, a Long Short-Term Memory (LSTM) architecture will train on word indexing to extract ADRs and enhance classification accuracy. The rest of the paper will be structured as follows: Section II highlights the related work, Section III explains the proposed method, and Section IV analyzes the experimental results and critically discuss it in the context of state-of-the-art comparison.

II. Related Work

Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques have been utilized to extract ADR from clinical reports, health records, and several datasets [12]. Trigger Terms (TTs) are the common NLP approach that has been utilized by researchers for extracting ADR [13]. The way of utilizing TTs has been examined differently within the literature, some studies have utilized the TTs through rule-based methods where a set of predefined rules are curated to perform the extraction. Trigger term denotes the keywords that occasionally occurred before or after the ADR. Hence, the rule-based technique can utilize a set of trigger terms stored in a list and any possible occurrence of these terms would indicate an occurrence of ADR either before or after the trigger term. On the other hand, other researchers have utilized the TTs through Machine Learning Techniques (MLT) where example data (i.e., labeled sentences) can be used to build a statistical model that measures the probability of containing an ADR. Labeled medical review sentences refer to the annotation of "1" and "0" where the first label refers to the existence of ADR, while the second label refers to the absence of ADR. In this regard, researchers have utilized TTs such as "having", "got", "had", "severe", 'occasional', 'joint', and others to train the MLT of predicting the occurrence of ADR. The common way to do that is through a representation known as N-gram or Bag-of-Words (BoW) [6]. This representation aims at aligning all the possible TTs as attributes and comparing each sentence in terms of containing these terms. After representing the TTs through BoW, an MLT algorithm will be trained on such a resulting matrix to build a predictive model that will be tested on the new ADR existence of new sentences.

The majority of recent studies have concentrated on enhancing, enriching, and extending the trigger terms to improve the detection of ADRs from social reviews. For example, Ebrahimi and Yazdavar *et al*. [9] have proposed the use of Unified Medical Language (UML) as an external medical knowledge source. UML can annotate terms based on its medical label. Hence, the authors have input medical review text into the UML to identify new TTs that can help detect ADRs. Consequentially, the authors have utilized the newly extracted TTs from UML through both a rule-based technique along with an MLT algorithm called Support Vector Machine (SVM) to detect ADRs.

On the other hand, Moh *et al*. [14] have treated the ADR extraction problem from the aspect of positive and negative sentimental polarities. The authors have collected tweets from Twitter where users express their feedback regarding medical products. Then, the authors utilized a knowledge source known as SentiWordNet to identify the polarity of adjectives and adverbs. This source can detect sentimental terms (e.g., good, bad, awesome, terrific, etc.) and classify them into either positive or negative. Hence, the authors have exploited SentiWordNet to explore additional TTs. Consequentially, the newly extracted TTs have been utilized through a rule-based technique to detect the occurrence of ADRs.

Kiritchenko and Mohammad *et al*. [15] have incorporated surface features into the TTs to detect ADRs from Twitter. The authors have utilized multiple types of surface features such as Twitter hashtags and emotions, negations (e.g., not, never), and punctuation (e.g., exclamation and question marks). Then, the authors represented both TTs and surface features as N-gram features to train a Support Vector Machine (SVM) algorithm for ADR detection.

Yousef and Tiun *et al*. [6] have extended the TTs through a statistical approach known as Pointwise Mutual Information (PMI). This approach calculates the probability of co-occurrence among terms. Hence, the authors have input medical reviews into the PMI to generate the most co-occurrence terms, especially the ones that occurred frequently with the ADRs. Using the MedSyn dataset, the authors have trained three MLT algorithms Naïve Bayes (NB), Logistic Regression (LR), and SVM on the new extended TTs for detecting ADRs.

Zhang and Cui *et al.* [7] have examined the role of grammatical syntactic properties in terms of generating new TTs that are associated with ADRs. The authors have utilized Part-of-Speech (POS) tagging which aims to give a syntactic tag for each term (e.g., verb, noun, adjective, etc.). Then, a set of selected syntactic patterns for ADR pairs have been used to train an SVM algorithm for detecting ADRs.

Pandya and Patel *et al*. [8] have taken the advantage of Electronic Health Records (EHR) where numerous clinical narratives and notes are written by physicians, nurses, and even by the patients themselves to generate new TTs. Using statistical correlation approaches, the authors have identified new TTs. Lastly, an LR algorithm has been used to predict the occurrence of ADRs.

Statistical approaches are also used for ADR extraction tasks. For example, Yates and Goharian *et al*. [10] applied the Latent Dirichlet Allocation (LDA) that aims to analyze the semantics of a particular text data. Similarly, Joshi and Attar *et al*. [16] have used the LDA for topic modeling in the context of ADR extraction. The authors have utilized a web scraping method to collect reviews from medical forums. Consequentially, some NLP approaches such as word tokenization and stemming have been applied. Then, the LDA has been employed as a topic modeling approach where a clustering method has been applied to categorize the reviews.

Abed and Jabber *et al*. [11] have utilized a statistical technique called Latent Semantic Analysis (LSA) to identify ADR in which the LSA can statistically analyze the semantics of particular data. The authors have used the MedSyn dataset and applied Count Vector (CV) and Term Frequency Inverse to analyze the text. To identify the ADR, the authors trained LR, NB, and SVM classifiers.

On the other hand, a more modern neural network architecture known as Long Short Term Memory (LSTM) has been used for ADR extraction. For example, Santiso and Perez *et al.* [17], Tange and Hu *et al*. [18], and Li and Huang *et al*. [19] have examined the LSTM architecture for the task of ADR extraction from medical reviews. The aforementioned studies have utilized the lexical features of the words such as term length and term position and encoded such information into the LSTM.

Rosa and Fenza *et al*. [20] have addressed the trustworthiness of explicitly mentioned ADRs within social reviews (specifically Twitter). The authors have intended to validate those ADRs mentioned through a Fuzzy Formal Concept Analysis (Fuzzy FCA) in which the ADR extracted from Twitter is aligned with a specific source of PubMed. The authors have identified the reliability correlations using a threshold value that counts the frequencies.

III. THE PROPOSED METHOD

The framework of this study contains multiple phases as depicted in Fig. 1. The first phase refers to the medical reviews dataset that will be used for the analysis of ADR existence. Then, the preprocessing tasks will be used including tokenization of review documents, removing unnecessary data including stopwords, and word stemming. Such preprocessing tasks would contribute toward improving the analysis. This will be followed by splitting the dataset into training and testing datasets. Consequentially, word mapping is then applied in which the words within the review documents will be replaced with index numbers. After that, these indices will be fed into a Long Short Terms Memory (LSTM) architecture which will be classified of the documents into ADR-contained and ADR-absence. Lastly, the evaluation will take a place where the classification accuracy is considered.

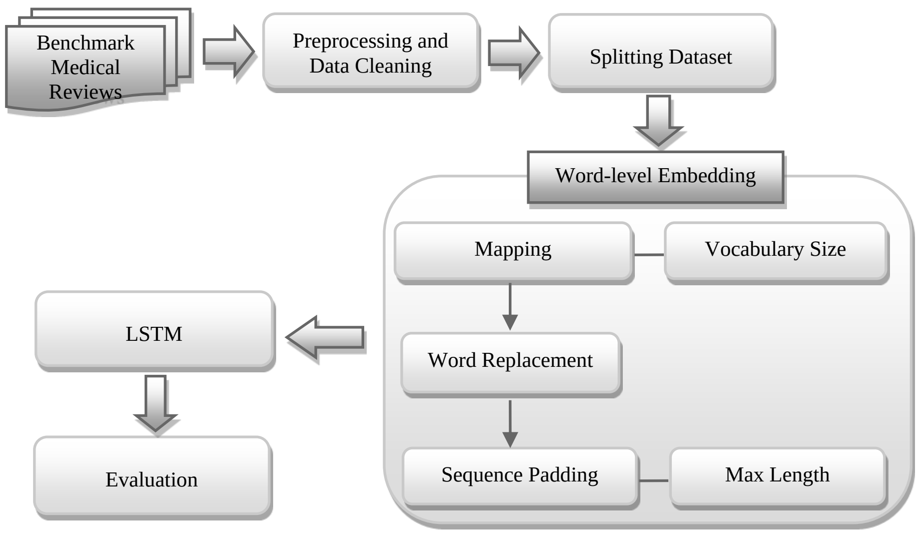

Figure 1. The implementation components.

A. Benchmark Medical Reviews

Attaining medical reviews from online stores would be a feasible task, especially with the emergence of the latest web scraping agents that can specify portions of the web content. However, the most challenging task lies in identifying side-effects or ADRs which require an expert in the domain. Furthermore, it requires intensive manual curation for the ADR entities. Fortunately, this sort of research effort has been conducted earlier by the study of Yates and Goharian [21] who collected reviews from medical online stores and accommodated an annotation task where the reviews that contained ADR was labeled as "1" and the reviews that do not contain ADR was labeled as '0', such a dataset is known as MedSyn. Table I shows the statistics of such a dataset.

TABLE I. DESCRIPTION OF MEDSYN DATASET [21]

Statistics	Description
# reviews	945
# sentences	1372
# words	12763
# unique words	2058
# ADRs	761
# unique ADRs	105

B. Pre-processing and Data Cleaning Phase

Since this study is dealing with textual information, pre-processing and data cleaning will be implemented. The first category of such preprocessing data is tokenization in which each review document will be separated into a series of words to facilitate further word-scale processing. The second category is stopword removal in which the absence of these words would not affect the contextual information of the text. Lastly, the third category is stemming in which each word will be converted into its root form by removing any added affixes while not affecting the meaning of the word.

C. Splitting Dataset

The data will be split into 70% for training and 30 % for testing conforming to the research study of Abed and Jabber *et al*. [11], Mohammad and Sabrina *et al.* [22]. The classifier will be trained on the training data to be able to construct the model that learns during the training the potential cases of the existence of ADR. Thus, the classifier will have the ability to handle new data and distinguish whether it contains ADR or not in the testing stage.

D. Word-Level Embedding

This research employs word-level embedding. Compared to knowledge-based techniques, word embedding techniques do not rely on any prior knowledge of the word. Instead, it aims to convert the words into a sequence of indices or numbers, facilitating the understanding of text relationships sequentially [23]. To apply such an embedding mechanism, three sub-tasks must be applied including mapping, word replacement, and sequence padding. These three sub-tasks will be tackled in the following subsections.

1) Mapping: To process the textual data through any classification method, it is necessary to map the text into numeric data. For this purpose, the list of the unique words will be created by removing the duplicated word from the stemmed review data excluding the stopwords. In addition, the words will be enumerated in the unique word set. To accomplish such a task, it is imperative to identify the vocabulary size of the datasets. Thus, the size of the unique stemmed word is 1636 words.

2) Word replacement: The word-level embedding approach was used to convert the training and testing data from stemmed review data to numbers in which each stemmed word will represent by a unique number from range [1–1636].

3) Sequence padding: Fundamentally, the classifier requires a fixed length of the input. Therefore, the representation of the input should have a fixed length for each review. To achieve this task, the zero number (0) should undergo the process of sequence padding. The sequence padding will be applied to make the other stemmed reviews equivalent to the fixed-length reviews. This will be achieved by adding a series of zeros to the ending of each stemmed review. For this purpose, it is essential to determine the longest-stemmed review in the dataset with the greatest number of words. The exact max length for all stemmed reviews in the dataset is 55 words. That means that 55 is considered a fixed length for all reviews in the dataset.

E. LSTM

Long Short-Term Memory (LSTM) is a modern architecture of neural networks that has an outstanding performance with sequential data classification [24–26]. A memory gate that allows LSTM to remember significant information represents the additional component compared to the traditional neural network. Moreover, another gate called the forgetting gate is also depicted in LSTM which allows its architecture to forget irrelevant information [27].

In this study, the proposed LSTM has four main components. The input layer is considered the first component in the LSTM. Since the longest stemmed review in the dataset has 55 words thus, the length of the input layer has been determined as 55 as shown in Fig. 2.

The second component is the LSTM cells which will contain both memory and forgetting gates. While the third component of LSTM is the hidden layer. Regarding the number of neurons in the LSTM cells and a hidden layer of LSTM, there is no specific rule for determining their sizes as stated by Soutner and Müller [28]. However, it depends on trial and error to achieve the best performance. For this matter, the size of the LSTM cells and hidden layer has been set as 32 for the LSTM cells and 256 for the hidden layer as shown in Fig. 2 and Table II.

Lastly, the output layer is the LSTM's fourth component in which the stemmed review is classified into '1' which indicates the existence of ADR in the review, and '0' which indicates the non-existence of ADR. This layer would have one neuron since it refers to a binary class label.

TABLE II. HYPER-PARAMETERS OF LSTM

Parameter	Quantity
Input dimension	55
Vocabulary size	1636
Epochs	40
LSTM	32
Fully connected	256
Output	1
Batch size	256

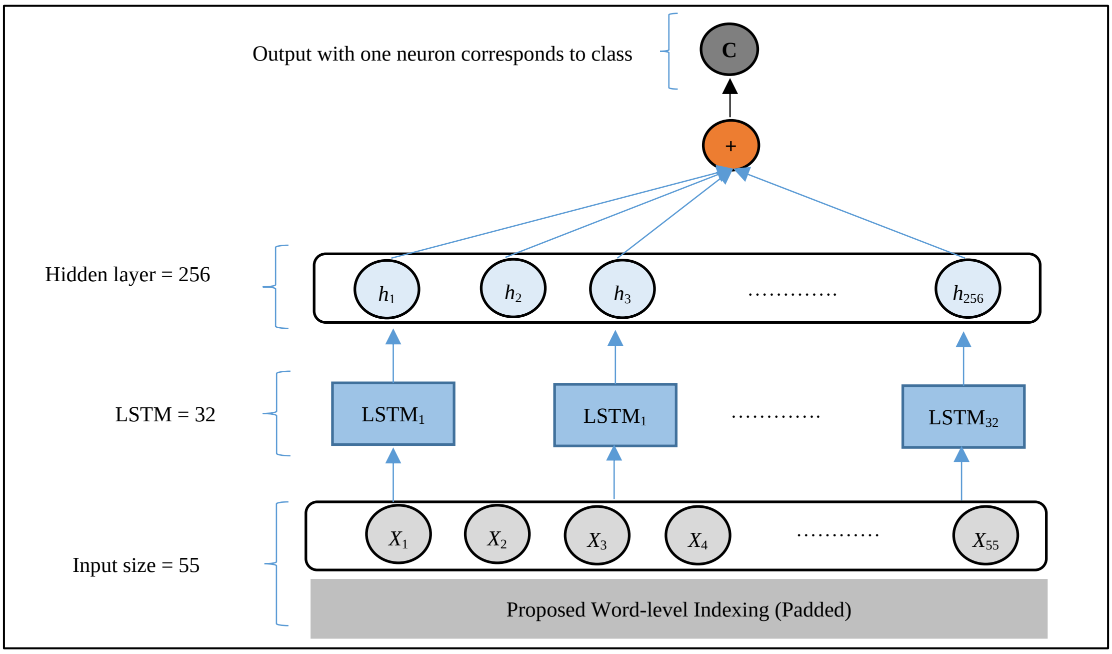

Figure 2. The proposed LSTM architecture.

F. Evaluation

The way of assessing the classification results is straightforward, in which the confusion matrix will be used as shown in Fig. 3. The confusion matrix is composed of four variables: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). where TP refers to the ADR-containing documents that are correctly classified into ADR, FP refers to the ADR-containing documents that are wrongly classified into non-ADR, FN refers to the ADR-free documents that are wrongly classified into ADR, and TN refers to the ADR-free documents that are correctly classified into non-ADR.

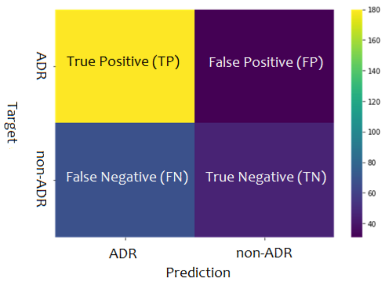

Figure 3. Confusion matrix.

Based on the aforementioned variables, Precision, Recall, and F1-score will be calculated using the following equations:

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

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

IV. Results and Discussion

Before the analysis of the experimental results, it is necessary to examine how the model fits. For this purpose, Fig. 4 considers the loss and accuracy through the iterations during training and testing.

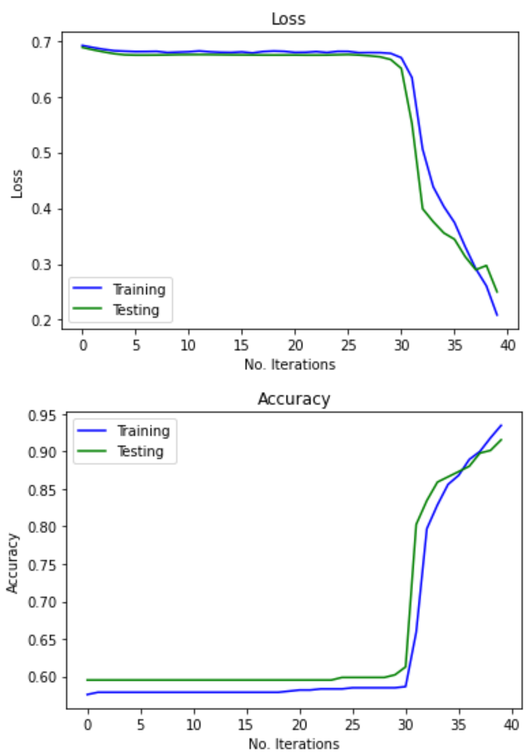

Figure 4. Loss and accuracy during training and testing.

As shown in Fig. 4, both training and testing loss followed a similar trend, in which they began with a high value and continue to decrease until reaching an error of

0.2. Similarly, both training and testing accuracy followed a similar trend too where they start with low values and gradually increased until reaching an accuracy above 90%. Hence, it is obvious that the model is neither under nor over-fitting. Then, the LSTM model has been validated through the testing data where 10 runs have been executed. Table III shows the results of precision, recall, and F1-score for each run.

TABLE III. THE RESULTS OF THE PRECISION, RECALL, AND F1 SCORE FOR 10 RUNS

Run	Precision	Recall	F1-score
Run 1	0.90	0.89	0.89
Run 2	0.87	0.87	0.87
Run 3	0.89	0.88	0.88
Run 4	0.88	0.87	0.88
Run 5	0.87	0.87	0.87
Run 6	0.92	0.92	**0.92**
Run 7	0.88	0.88	0.88
Run 8	0.90	0.90	0.90
Run 9	0.87	0.86	0.86
Run 10	0.89	0.89	0.89

As depicted in Table III, the minimum F1-score achieved was 0.86. Whereas, the maximum F1-score was 0.92. Comparing such results with the state-of-the-art (see Table IV) would reveal an outstanding performance of the word indexing and LSTM. For example, the study of Yousef and Tiun *et al*. [6], who used extended trigger terms along with different classifiers, achieved the highest F1-score of 69%. On the other hand, the study of Abed *et al.* [11] has achieved the highest F1-score of 82% using LSA and multiple classifiers. The lowest F1-score obtained by the proposed LSTM (i.e., 0.86) is outperforming the baseline studies (see Table IV). This demonstrates the efficacy of using LSTM with sequences of word indices for the task of ADR extraction.

The results achieved by combining word indexing and LSTM should be deciphered to clarify the reason for the ADR classification accuracy enhancement For this purpose, different experiments and comparisons were carried out. These experiments can be explained as follows:

(1) Comparison against studies that used the same dataset with trigger terms.
(2) Comparison against studies that used the same dataset with statistical approaches.
(3) Comparison against studies that used the same dataset with LSTM
(4) Examining word indexing with different machine learning techniques.
(5) Examining word indexing with different deep learning techniques.
(6) Examining LSTM with statistical features.
(7) Examining LSTM with lexical features.

All the above-mentioned tasks are feasible except for task (3) where no study has used the same benchmark dataset with LSTM. Therefore, task (3) can be subrogated by task (7) in which this study has successfully combined lexical features such as word length and position with LSTM. Table IV shows the comparison.

TABLE IV. COMPARISON AGAINST THE STATE-OF-THE-ART

Technique	Study	Method	F1-score
Task (1)	Yousef and Tiun *et al.*, [6]	TT + LR	68%
	Yousef and Tiun *et al.*, [6]	TT + SVM	69%
	Yousef and Tiun *et al.*, [6]	TT + NB	61%
Task (2)	Yates and Goharian *et al*. [10]	LDA + CRF	58%
	Abed and Jabber *et al*., [11]	LSA + LR	82%
	Abed and Jabber *et al*., [11]	LSA + SVM	81%
	Abed and Jabber *et al*., [11]	LSA + NB	71%
Task (4)	Our study	Word indexing + LR	58%
	Our study	Word indexing + SVM	62%
	Our study	Word indexing + NB	47%
Task (5)	Our study	Word indexing + CNN	63% ± 71%
Proposed	Our study	Word indexing + LSTM	**86% ± 92%**
Task (6)	Our study	LSA + LSTM	45% ± 73%
Task (7)	Our study	Lexical + LSTM	74% ± 81%

As shown in Table IV, for the task (1) and task (4), it is clear that when word indexing is used with different machine learning classifiers, the results of the F1-score tend to be low. This has been depicted where the accuracy, when word indexing is used, are 58%, 62%, and 47% for LR, SVM, and NB respectively compared to 68%, 69%, and 61% acquired by the study of Yousef and Tiun *et al*. [6]. for the same classifiers but with trigger terms. In the same regard, for the task (5), the word indexing has been examined with the Convolutional Neural Network (CNN) deep learning architecture, and the results of the F1-score range from 63% to 71%.

Examining LSTM with features other than word indexing still has not shown significant performance. For task (6), LSTM with the statistical feature of LSA has shown an F1-score from 45% to 73%. Comparing such results with the study of Abed *et al.* [11] (task (2)) who used LSA with LR, SVM, and NB which achieved an F1-score of 82%, 81%, and 71% respectively, LSTM with statistical features (LSA) is less effective (i.e., from 45% to 73%). In the same manner, for the task (7), the combination of lexical features and LSTM has shown an F1-score from 74% to 81% which is still insignificant compared to the literature.

However, the proposed combination of word indexing and LSTM has shown a minimum F1 score of 86%, while the maximum F1 score was 92%. Comparing even the minimum value of the F1-score with the other experiments and studies would reveal an outstanding performance. This demonstrates the efficacy of using LSTM with sequences of word indices for the task of ADR extraction.

V. CONCLUSION

This study concentrates on the task of extracting ADR mentions within medical social reviews. Unlike the literature where the dependence was on dictionary-based or statistical methods, this study aims to utilize the word sequences along with the LSTM. Using a benchmark dataset, the proposed LSTM showed superior performance compared to the baseline studies. For future directions, more sophisticated word embedding vectors such as Word2Vec or Glove could be used with the LSTM to improve classification accuracy.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Asmaa J. M. Alshaikhdeeb contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. Whereas, Yu-N Cheah helped supervise the research by providing critical feedback and helped shape the research. All authors had approved the final version.

FUNDING

This study has been supported by the Universiti Sains Malaysia.

REFERENCES

[1] K. Denecke and Y. Deng, "Sentiment analysis in medical settings: New opportunities and challenges," *Artificial Intelligence in Medicine*, vol. 64, no. 1, pp. 17–27, 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*, University of Chicago Press, 2020.
[3] G. O. Diaz and V. Ng, "Modeling and prediction of online product review helpfulness: A survey," in *Proc. the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 698–708.
[4] B. Audeh, F. Bellet, M.-N. Beyens, A. L.-L. 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] S. Gupta, M. Gupta, V. Varma, S. Pawar, N. Ramrakhiyani, and G. K. Palshikar, "Multi-task learning for extraction of adverse drug reaction mentions from tweets," in *Proc. European Conference on Information Retrieval*, 2018, pp. 59–71.
[6] 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.
[7] Y. Zhang, S. Cui, and H. Gao, "Adverse drug reaction detection on social media with deep linguistic features," *Journal of Biomedical Informatics*, vol. 106, 103437, 2020.
[8] 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*, vol. 24, no. 3, pp. 172–178, 2020.
[9] 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.
[10] A. Yates, N. Goharian, and O. Frieder, "Extracting adverse drug reactions from social media," in *Proc. the 29th AAAI Conference on Artificial Intelligence*, 2015.
[11] A. H. Abed, S. A. Jabber, and A. A.-J. Altameemi, "Extracting adverse drug reaction using latent semantic analysis from medical social media reviews," *ICIC Express Letters*, vol. 15, no. 8, pp. 907–914, 2021.
[12] H. Gurulingappa, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Identification of adverse drug event assertive sentences in medical case reports," in *Proc. ECML PKDD 2011 Workshop on Knowledge Discovery in Health Care and Medicine*, 2011, pp. 16–27.
[13] S. Santiso, A. Casillas, A. Pérez, M. Oronoz, and K. Gojenola, "Document-level adverse drug reaction event extraction on electronic health records in Spanish," *Natural Language Processing*, no. 56, pp. 49–56, 2016.
[14] 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, 18, 2017.
[15] S. Kiritchenko, S. M. Mohammad, J. Morin, and B. Bruijn, "NRC-Canada at SMM4H shared task: Classifying tweets mentioning adverse drug reactions and medication intake," in *Proc. the Social Media Mining for Health Applications Workshop at AMIA-2017*, Washington, DC, USA, 2018, pp. 1–11.
[16] C. Joshi, V. Z. Attar, and S. P. Kalamkar, "An unsupervised topic modeling approach for adverse drug reaction extraction and identification from natural language text," *Advances in Data and Information Sciences*, pp. 505–514, 2022.
[17] S. Santiso, A. Perez, and A. Casillas, "Exploring joint AB-LSTM with embedded lemmas for adverse drug reaction discovery," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 2148–2155, 2018.
[18] B. Tang, J. Hu, X. Wang, and Q. Chen, "Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF," *Wireless Communications and Mobile Computing*, vol. 2018, 2379208, 2018.
[19] D. Li, L. Huang, H. Ji, and J. Han, "Biomedical event extraction based on knowledge-driven tree-LSTM," in *Proc. the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 1421–1430.
[20] M. Rosa, G. Fenza, A. Gallo, M. Gallo, and V. Loia, "Pharmacovigilance in the era of social media: Discovering adverse drug events cross-relating Twitter and PubMed," *Future Generation Computer Systems*, vol. 114, pp. 394–402, 2021.
[21] A. Yates and N. Goharian, "ADRTrace: Detecting expected and unexpected adverse drug reactions from user reviews on social media sites," in *Proc. European Conference on Information Retrieval*, Berlin, Heidelberg, Germany, 2013, pp. 816–819.
[22] Y. R. N. Mohammad, T. Sabrina, and O. Nazlia, "Extended trigger terms for extracting adverse drug reactions in social media texts," *Journal of Computer Science*, vol. 15, no. 6, pp. 873–879, 2019,
[23] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency detection using a bidirectional LSTM," arXiv preprint, arXiv:1604.03209, 2016.
[24] Z. Ferdoush, B. N. Mahmud, A. Chakrabarty, and J. Uddin, "A short-term hybrid forecasting model for time series electrical-load data using random forest and bidirectional long short-term memory," *International Journal of Electrical & Computer Engineering*, vol. 11, no. 1, pp. 763–771, 2021.
[25] A. Nasser and H. Al-Khazraji, "A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance," *International Journal of Electrical & Computer Engineering*, vol. 12, no. 1, pp. 721–730, 2022.
[26] I. Pratheek and J. Paulose, "Prediction of answer keywords using char-RNN," *International Journal* of *Electrical and Computer Engineering (IJECE)*, vol. 9, no. 3, pp. 2164–2176, 2019.
[27] G. Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, 2020.
[28] D. Soutner and L. Müller, "Application of LSTM neural networks in language modelling," in *Proc. International Conference on Text, Speech and Dialogue*, 2013, pp. 105–112.