ADVERSE DRUG REACTION DETECTION USING LATENT SEMANTIC ANALYSIS

Ahmed Adil. N [1], Nazlia Omar [2]

University of Al-Anbar, Al-anbar, Iraq[1]

Center for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology2
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia

E-mail: E-mail: ahmed.adil.nafea@gmail.com, nazlia@ukm.edu.my

***Abstract*—Detecting adverse drug reactions (ADRs) is one of the important information for determining the view of the patient on one drug. Most studies have investigated the extraction of ADRs from social networks, in which users share their opinion on a particular medication. Some studies have used trigger terms to detect ADRs. Such studies showed remarkable performance in terms of extracting ADR. However, these terms only would not be sufficient since it needs to be extended periodically when new side effects or new medical-related entities are being discovered. In addition, the feature space with trigger terms would lack latent semantic. This study aims to propose a semantic method based on Latent Semantic Analysis (LSA) for improving the detection of ADR. A benchmark dataset has been used in the experiments along with several pre-processing operations that have been applied including stop word removal, tokenization, and stemming with three classifiers that were trained on the proposed LSA, namely Support Vector Machine, Naïve Bayes, and linear regression; In addition, two representations of documents were used namely TF and TFIDF. Results showed that the proposed LSA outperformed outperform the baseline extended trigger terms by achieving 82% of F-measure for the dataset. Such superiority highlights the use of LSA where the semantic correspondences have been identified correctly rather than using a predefined list of trigger terms.**

***Keywords*—Adverse drug reaction; latent semantic analysis; Naïve Bayes; support vector machine; logistic regression.**

I. INTRODUCTION

The rise of social networks has contributed toward expanding the textual information dramatically in the last years. Regular users nowadays would have the ability to freely express their minds toward plenty of subjects (Kiritchenko et al. 2014). One of these subjects is the product review where a user can evaluate a specific product and describing its advantages and disadvantages based on his/her experience with the product (Liu et al. 2017). ADR detection has been depicted in the literature where numerous studies have crawled data from social networks such as Twitter or from drug websites. In such data collection, the comments or reviews by regular users have been addressed in order to extract the ADR mentions. For example, a review of 'after I took this medicine, I felt dizzy' contains an ADR of 'dizzy' where the user in this review is describing a side-effect from taking a particular medicine.

Several studies proposed different techniques for ADR extraction. Most of the studies have utilizes machine learning technique classifiers such as SVM and NB. For the feature space, most of the studies have used the trigger terms (Ebrahimi et al. 2016; Kiritchenko et al. 2018; Mohammad Yousef et al. 2019; Pain et al. 2016; Plachouras et al. 2016). Yet, using trigger terms only would not be sufficient since it needs to be extended periodically when new side effects or new medical-related entities are being discovered. In addition, the feature space with trigger terms would lack latent semantic. For example, 'I took this medicine' and 'I consume pills' both sentences have trigger terms of 'took' and 'consume'. Examining the two words in the feature space would be ineffective since they have the same meaning. Therefore, it is necessary to address a semantic technique for improving detection accuracy.

This study aims to propose a semantic method based on Latent Semantic Analysis (LSA) for improving the detection of ADR. A benchmark dataset has been used in the experiments along with several pre-processing operations that have been applied including stop word removal, tokenization, and stemming with three classifiers that were trained on the proposed LSA, namely Support Vector Machine, Naïve Bayes, and linear regression; In addition, two representations of documents were used namely TF and TFIDF.

II. RELATED WORK

The literature has shown great interest in the task of ADR detection. The benchmark dataset of medical reviews was firstly presented by Yates & Goharian (2013). The authors also utilized trigger terms with the rule-based technique to identify the studies with ADR. They extract ADR automatically from user feedback on different social media platforms to classify adverse reactions not reported by the United States Food and Drug Administration (FDA). They used different lexicons, identify patterns, and created a range of synonyms, including variations in medical terminology and identify trends. They identify "expected" and "unexpected" ADRs. The context language (drug) was used to determine the frequency of unexpected detected ADR.

Pain et al. (2016) presented an ADR detection technique using SVM to the classifier. The proposed method utilized a set of keywords and hashtags trigger terms that were frequently occurred with ADR. The authors used a medical review collected data from Twitter to provide an automatic

drug–effect detection. The proposed features can identify numerous types of drug–effect entities. Their research described developing Post-marketing surveillance (PMS) methods specifically in Particularly for messy types of text found on Twitter.

Ebrahimi et al. (2016) employed a set of medical concepts with specifically named entities as trigger terms to determine the side effects of drugs from medical reviews. POS tagging was utilized to identify the syntactic tag of terms. Two classifiers, namely, a rule-based classification method and SVM, were adopted to detect the side effects of drugs. This research developed a method to identify side effects in medication reports as a subtask to identify implicit perceptions in medical literature and distinguish side effects and disease symptoms.

Plachouras et al. (2016) applied a set of trigger terms or Gazetteer features, along with an N-gram representation, to extract adverse drug events from Twitter reviews. This research presented a system for large-scale pharmacovigilance support. They tackled the question of adverse event extraction from tweets via training and testing a supervised binary classifier. The authors implemented the SVM classification method to accommodate the final extraction by using words and keywords, surface characteristics, a list of gazetteers, POS tags, and sentiment analysis.

A group of researchers from NRC-Canada Kiritchenko et al. (2018) At the AMIA-2017 Workshop on Social Media Mining for Health Applications (SMM4H), engaged in two joint activities. Task 1 was about classifying tweets with reference to ADR, while Task 2 focused on classifying tweets describing personal intake of medications. With regard to both tasks, Vector Machine Classifiers were trained using a variety of surface-specific features, feelings, and domain-specific features through the presentation of an SVM technique for ADR extraction. The authors filtered the trigger terms to use a domain-specific one for improving the accuracy of detection. Experiments were conducted using Twitter medical reviews.

Emadzadeh et al. (2017) this study has used latent semantic analysis with a hybrid semantic analysis in order to combine the Unified Medical Language System (UMLS) in order to improve the performance in terms of extracting ADR. They propose to their corresponding standardized identifiers a modular NLP pipeline for mapping (normalizing) colloquial mention of ADRs. For evaluation, they use a publicly available, annotated corpus of 2008 tweets (Nikfarjam et al. 2015).

The study of Mohammad Yousef et al. (2019) tackled the extraction of ADR from social networks where users express their views on a specific medication. Obtaining entities mainly depends on specific terms that may occur before or after ADR, called trigger terms. Those terms should be extended, however. The aim of this study was to propose an extension of the trigger terms based on the multiple N-gram representations. Where used in three classificatory, namely SVM, LR, and NB the proposed extension is being trained. In addition, two document representations including the TFIDF and TF were used. The experiments were conducted using secondary data from drug websites.

III. Proposed Method

The methodology of this study consists of five phases fig.1. The first phase is the preparation of annotated drug reviews where the dataset used is from a benchmark dataset by Yates & Goharian (2013) in which Mohammad Yousef et al. (2019) modified some of some structure by adding more meaningful columns of the data. The second phase will contain pre-processing tasks such as tokenization, stop word removal, and stemming. The third phase aims to represent the terms in a vector space representation using both Count Vector (TF) and term frequency-inverse document frequency (TF-IDF). The fourth phase contains the semantic analysis using the proposed LSA. The fifth phase will address the classification where three classifiers will be used including SVM, NB, and LR. Each phase is discussed in further detail in the next subsections.

A. Dataset

The dataset used is from a benchmark dataset by Yates & Goharian (2013) in which Mohammad Yousef et al. (2019) modified some of some structure by adding more meaningful columns of the data. The dataset used in this study containing 2500 reviews (labeled 246 documents). Each document contains one or more sentences. All documents contain 944 sentences. Those sentences are collected from the Twitter platform. The total number of ADR are 982 for all documents. These documents are written in the English language. The review dataset is collected from Drug Review Sites on Social Media, namely, drugratingz.com, askapatient.com, and drugs.com. Table 1 shows a sample example of the dataset.

Table 1: Sample of the dataset

DOC	SEN	CLASS	REVIEW	ADR
1	1	1	My joint pain is very severe.	['pain']
2	1	0	I was fine in the beginning.	[]
2	2	1	Lower back pain.	['pain']
2	3	0	Swelling of hands.	[]
3	1	1	General Muscle Aches and Fatigue.	['FATIGUE']
4	1	1	Numbness in toes	['Numbness']
4	2	1	Can't walk, everything aches.	['aches']

Table 2 shows the dataset details.

Table 2: Dataset details

Attribute	Total
Number of total reviews	2500 (labeled 246)
Number of sentences	944
Number of ADR	982

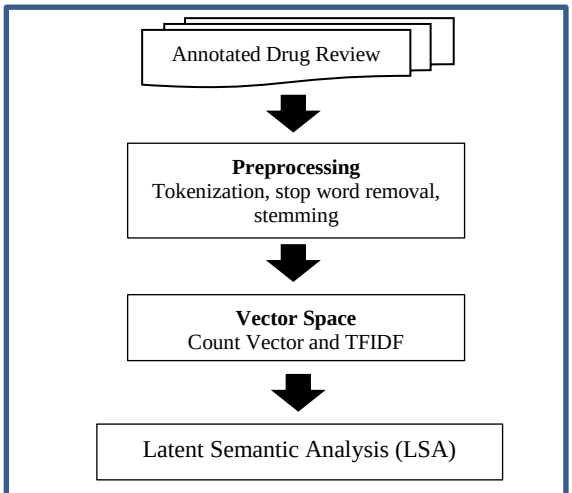


Fig. 1 Proposed LSA methods

B. Preprocessing

In this stage, the process of splitting the text when running on a set of pre-processing algorithms to prepare it for the next stages. The above tasks can be described as follows:

1. Stop word removal: This activity is aimed at eliminating a language's common words that don't hold any important details of their own. At the pre-processing point, these terms are often omitted to reduce the number of less informative features known as noise data. Fig. 2 Shows an example of stripping of the stop-words.

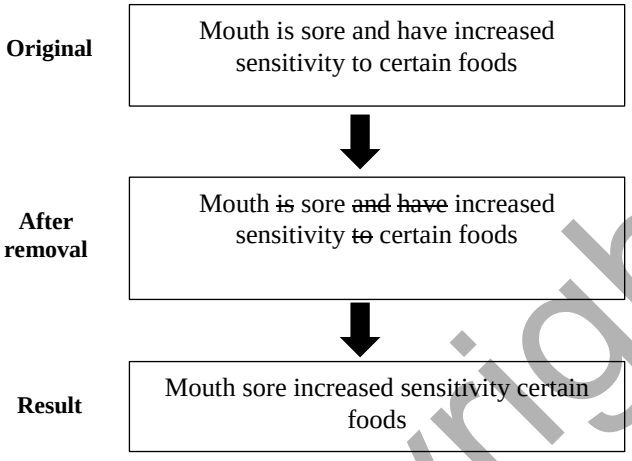


Fig. 2 Example of removing stop-words

2. Tokenization: is a process that attempts to transform the text into a sequence of sentences and then convert those sentences into sequences of tokens (i.e. words). Fig. 3 shows the tokenization process.

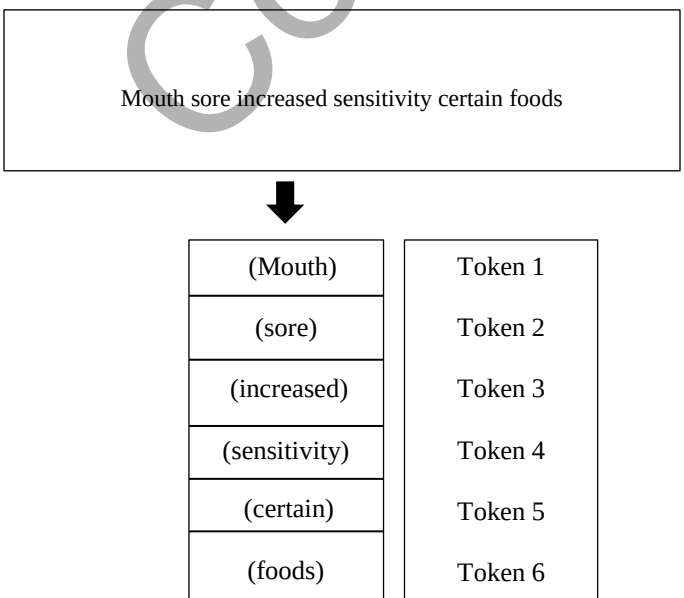


Fig. 1 Example of tokenization

3. Stemming: The final stemming preprocessing step will be applied. This mission aims at restoring the origin of words by removing the various suffixes. In this study, Porter's Stemmer algorithm (Porter 1980) was used for this manner It is based on the idea that suffixes in English It is one of the most popular methods of stemming proposed in 1980. Fig. 4 shows a description of a function with stemming words.

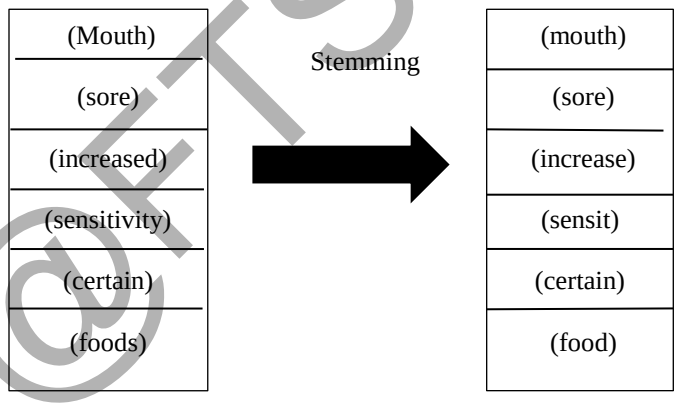


Fig. 2 Example of the stemming process

C. Term Representation

In this stage, the data will be represented the number of occurrences of the word in the documents by the term frequency (TF) or Term Frequency with Inverse Document Frequency (TF-IDF).

Term Frequency (TF): - In this process, the number of occurrences of the word in the document is represented at the term frequency. The formula used to solve the problem concerning frequency is:

$$W_d\,(t) = \; TD(t,\, d\,)$$

where *TD (t, d)* is the Word *T* frequency in document d

Inverse Document Frequency (IDF): - IDF seeks to have high weight for unusual conditions, and low typical conditions weights. The formula reads as:

$$IDF_t \; = log\, n \; \left(\frac{N}{N_t}\right)$$

where N_t is the number of documents that contain the word, and where N is the number of English documents.

Term Frequency with Inverse Document Frequency TF-IDF: - This method is a combination of two preceding TF and IDF methods. The formula regarding weighting as follows:

Word	IDF
Shoot	$log_2\left(\frac{3}{1}\right) = 0.477$
knee	$log_2\left(\frac{3}{1}\right) = 0.477$
feet	$log_2\left(\frac{3}{1}\right) = 0.477$
infrequ	$log_2\left(\frac{3}{1}\right) = 0.477$
experience	$log_2\left(\frac{3}{1}\right) = 0.477$
severe	$log_2\left(\frac{3}{1}\right) = 0.477$
joint	$log_2\left(\frac{3}{2}\right) = 0.176$
pain	$log_2\left(\frac{3}{3}\right) = 0$

$$W_t = TF(t,d).\ IDF_t$$

where $TF\ (t,d)$ refers to the term frequency t in document d and IDF (refers to the inverse document frequency of term t.

For example, consider three statements (i.e. documents) D1, D2, and D3, which have sentences as shown in the Table 3.

Table **Error! No text of specified style in document.**:
Example of three documents

Sample of medical text documents	
D_1=	*Shoot pain knee feet*
D_2=	*infrequ joint pain*
D_3=	*experience severe joint pain.*

To calculate the TF, first be determined for every word found in the statements. The singular terms are segmented as in Table 4

Table 4: Calculating the term frequency

	D_1	D_2	D_3
Shoot	1	0	0
knee	1	0	0
feet	1	0	0
infrequ	0	1	0
experience	0	0	1
severe	0	0	1
joint	0	1	1
pain	1	1	1

As shown in Table 3.4, number (1) is the word present in the phrase corresponding to the sentence given, while (0) is the absence of the word corresponding to the statements given.

Therefore, *IDF* will be calculated for each word corresponding to the specified three documents, note that N = total number of documents which is 3, and N_t is the number of word appearances in the three documents. *IDF* for each term can be determined based on Equation (3.4), as shown in Table 5.

Table 5: IDF calculation

Finally, by multiplying the TF and IDF, can be obtained from TF-IDF; this multiplication is shown in Table 6.

Table 6: TF-IDF calculation

	D_1	D_2	D_3
Shoot	0.477	0	0
knee	0.477	0	0
feet	0.477	0	0
infrequ	0	0.477	0
experience	0	0	0.477
severe	0	0	0.477
joint	0	0.176	0.176
pain	0	0	0

D. Proposed Latent Semantic Analysis

The Latent Semantic Analysis is a technique commonly used in the processing of NLP to define the similarities between two text classes (Froud et al. 2013). It attempts to analyze the relationships between two sets of documents by constructing a vector space for the meanings of both documents' phrases, expressions, and concepts. It can be achieved by vectoring the terms into two rows and columns where the terms are displayed in the rows and the documents in the columns represented. Using the frequency principle of terms theory, LSA can determine the essential relationship by counting the frequency of terms (Islam & Hoque 2010). Given the high dimensionality of the words in question, a post-processing technique called Singular Value Decomposition (SVD) is applied to minimize the dimensionality of the word matrix. In particular, SVD aims to reduce the number of rows without loosing the structure of similarity between columns.

Basically, LSA implements the matrix using TF OR TF-IDF by identifying the occurrences of words in respect documents. Hence, the Singular Value Decomposition (SVD) is applied in order to reduce the dimensionality of word vector. The following equation can be used for calculating SVD:

$$SVD\ = S\Sigma U^T$$

LSA first utilizes either Count Vector or TFIDF where all the unique words are grouped in separated attributes. Hence,

LSA inputs either CV or TFIDF matrix and output the same dimension matrix but with more sophisticated values that adequately indicate the semantic behind every term. This is conducted through a process known as Singular Value Decomposition (SVD). Then it will be classified by one of the classifications (SVM, NB, and LR) that he used in the baseline (Mohammad Yousef et al. 2019).

To illustrate the SVD, let X be an array containing three sentences for D_1 and D_2 with D_3 which are the dataset statements used:

This is a simple example of the work of the LSA. The term frequency representation has been stated as in Table 4.

In order to get the SVD, Y has to be calculated where Y is the union of documents in terms of words Y = X^T* X where X^T is the transpose of X. In addition, Z has to be calculated where Z is the union of words in terms of documents Z = X * X^T. First, the matrix X and its transpose X^T will be represented as follow:

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad X^T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Since $Y = X X^T$, so it can be represented as follow:

$$Y = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Hence, the results of the previous multiplication will be equivalent as follow:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 2 & 3 \end{bmatrix}$$

Similarly, $Z = X^T X$, so it can be calculated as follow:

$$Z = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

Therefore, to compute the SVD, using the equation (3,6) following formula has to be applied:

$$SVD\ (X) = S\Sigma U^T$$

where S is the eigenvector of Y, and U is the eigenvector of Z, and Σ is the root square of the eigenvalue of Z.

$$Eigenvector\ of\ Y = S = \begin{bmatrix} 0.29511 & 0.000i \\ 0.29511 & 0.000i \\ 0.29511 & 0.000i \\ 0.31639 & 0.000i \\ 0.38848 & 0.000i \\ 0.38848 & 0.000i \\ 0.70488 & 0.000i \\ 1 & 0.000i \end{bmatrix}$$

$$Eigenvector\ of\ Z = U = \begin{bmatrix} 0.75965 & 0.000i \\ 0.81442 & 0.000i \\ 1 & 0.000i \end{bmatrix}$$

$$Transpos\ of\ (U) = U^T = \begin{bmatrix} 0.75965 & 0.81442 & 1 \\ 0.000 & 0.000 & 0.000 \end{bmatrix}$$

$$Eigenvalue\ of\ Z = \begin{bmatrix} 6.3885 \\ 3.1873 \\ 1.4242 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sqrt{6.3885} & 0 & 0 \\ 0 & \sqrt{3.1873} & 0 \\ 0 & 0 & \sqrt{1.4242} \end{bmatrix} = \begin{bmatrix} 2.52 & 0 & 0 \\ 0 & 1.78 & 0 \\ 0 & 0 & 1.19 \end{bmatrix}$$

$$SVD\ (X) = S\Sigma U^T = \begin{bmatrix} 0.56493438498 & 0.427811405436 & 0.3511809 \\ 0.56493438498 & 0.427811405436 & 0.3511809 \\ 0.56493438498 & 0.427811405436 & 0.3511809 \\ 0.60567107202 & 0.458660331964 & 0.3765041 \\ 0.74367425664 & 0.563166869248 & 0.4622912 \\ 0.74367425664 & 0.563166869248 & 0.4622912 \\ 1.34936447184 & 1.021841697888 & 0.8388072 \\ 1.914318 & 1.4496676 & 1.19 \end{bmatrix}$$

Here the complex matrix is completed in finding the semantic. LSA first utilizes either Count Vector or TFIDF where all the unique words are grouped in separated attributes. Hence, LSA inputs either CV or TFIDF matrix and output the same dimension matrix but with more sophisticated values that adequately indicate the semantic behind every term. This is conducted through a process known as Singular Value Decomposition (SVD). Then it will be classified by one of the

classifications (SVM, NB, and LR) that he used in the baseline (Mohammad Yousef et al. 2019).

E. Classification

Machine learning is implemented in this step for classifying ADRs. Classification methods like SVM, NB, and LR are used to evaluate f-measure efficiency.

The first method of classification is SVM, which functions by determining an appropriate separator in a 2-dimensional space between data instances. SVM aims at the establishment of the optimal hyperplane with the following decision function (Ebrahimi et al. 2016)

$$f(\vec{x}) = sgn\big((\vec{x} \times \vec{w}) + b\big) = \begin{cases} +1: & (\vec{x} \times \vec{w}) + b > 0 \\ -1: & Otherwise \end{cases}$$

SVM maps the optimum hyperplane with the optimum margin. Assume a positive and negative data instances partitioned by a hyperplane and the shortest path p+(p-) is lying between the nearest positive and nearest negative instances (Moghaddam & Ester 2011). The margin of this hyperplane, in this case, is given as $p_+ + p_-$.

NB operates by defining the probabilities for the data instances of classes. You can measure the likelihood using the following equation (Elhadad et al. 2019)

$$\mathrm{P(C_i|d)} = \frac{\mathrm{P(C_i)P(d|C_i)}}{\mathrm{P(d)}}$$

where, given the predictor (x, attributes), P(Ci) is the posterior probability of class Ci.

LR functions by evaluating the linear class probability equation, which can be seen as follows (Montgomery et al. 2015)

$$y = a + bX$$

where X is the dependent variable, the y-intercept is a, and b is the line slope.

After implemented the classification (ADR) using the machine learning SVM, NB, and LR, it is necessary to validate the results of the categorization performed by the classifier. For the evaluation involving the f-measure can be calculated based on the explanation of the following equation.

$$F - measure = 2 \times \frac{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}}{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}}$$

where TN number of connections classified as normal, and in actual, they are normal connections, FP number of connections categorized as an intrusion, while they are normal connections, FN number of connections categorized as normal, while they are intrusions and TP number of connections categorized as intrusions, and in actual, they are intrusions.

The three classifiers are trained on the extracted patterns produced by the proposed LSA. This training aims to build a model that can classify new data in the testing phase. During the training, the model of each classifier learns the cases of the potential occurrence of ADRs. Table 5 shows the experimental settings.

Table 6: Experimental settings.

Experiment	Description
Feature	1. Baseline trigger terms with TF-IDF (Unigram, Bigram, Trigram, and Quadgram)
	2. Baseline trigger terms with count vector (Unigram, Bigram, Trigram, and Quadgram)
	3. Proposed LSA with TF-IDF (Unigram)
	4. Proposed LSA with count vector (Unigram)
Classifiers	1. SVM 2. NB 3. LR
Dataset	Benchmark dataset by (Yates & Goharian 2013) which is then updated by (Mohammad Yousef et al. 2019)
Training and Testing	70% for training and 30% for testing

IV. Experimental Results

The results acquired via the TF with classifiers including SVM, LR, and NB are being shown. the estimate was used f-measure. Note that, the results of classifiers will be shown based on the baseline research ADR using (trigger terms) Opposite the proposed using (LSA)

As shown in Table7, the results of f-measure for all classifiers using the proposed LSA via TF with SVM, NB, and LR have outperformed the ones by the baseline trigger terms. The performance of f-measure using SVM has improved from 67% (using trigger terms) to 81% (using LSA). As well as, the performance of f-measure using NB has improved from 61% (using trigger terms) into 68% (using LSA). Finally, the performance of f-measure using LR has improved from 67% using trigger terms to 82% (using LSA). Fig. 5 displays the f-measure of compare results proposed and baseline Via TF results with (SVN, NB, and LR).

The results acquired via the TF-IDF with classifiers including SVM, LR, and NB are being shown. the estimate was used f-measure. Note that, the results of classifiers will be shown based on the baseline research ADR using (trigger terms) Opposite the proposed using (LSA).

As shown in Table 8, the results of f-measure for all classifiers using the proposed LSA via TF-IDF with SVM, NB, and LR have outperformed the ones by the baseline

trigger terms. The performance of f-measure using SVM has improved from 69% (using trigger terms) to 80% (using LSA). As well as, the performance of f-measure using NB has improved from 61% (using trigger terms) into 72% (using LSA). Finally, the performance of f-measure using LR has improved from 68% using trigger terms to 80% (using LSA).

Such superiority is referred to as the use of LSA where the semantic correspondences have been identified correctly rather than using a predefined list of trigger terms. In a comparison between plain vector space model or the so-called N-gram representation against the feature space generated by LSA, (Hutchison et al. 2018) have demonstrated better f-measure of classification. This is because LSA can handle synonymy problems within a particular dataset. In addition, LSA can work well on the dataset with diverse topics which exactly would fit the adverse drug reaction datasets where various medical discourses are being tackled. Fig. 6 displays the f-measure of compare results proposed and baseline Via TF-IDF results with (SVN, NB, and LR).

Table 7: A comparison of results on the proposed approach and baseline via TF Results

	SVM	**NB**	**LR**
Baseline	0.67	0.61	0.67
Proposed approach	0.81	0.68	0.82

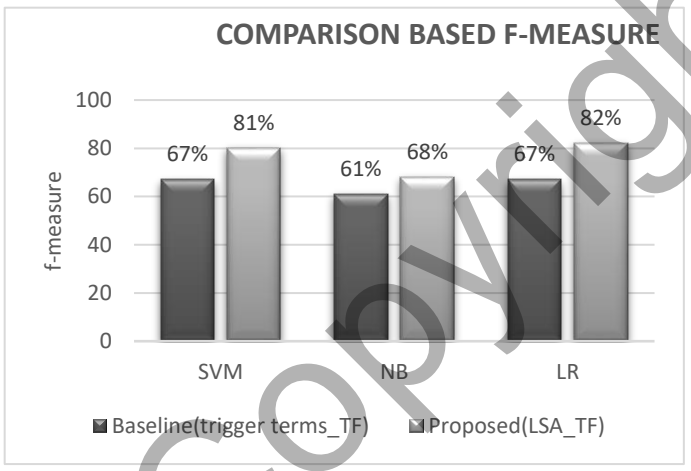
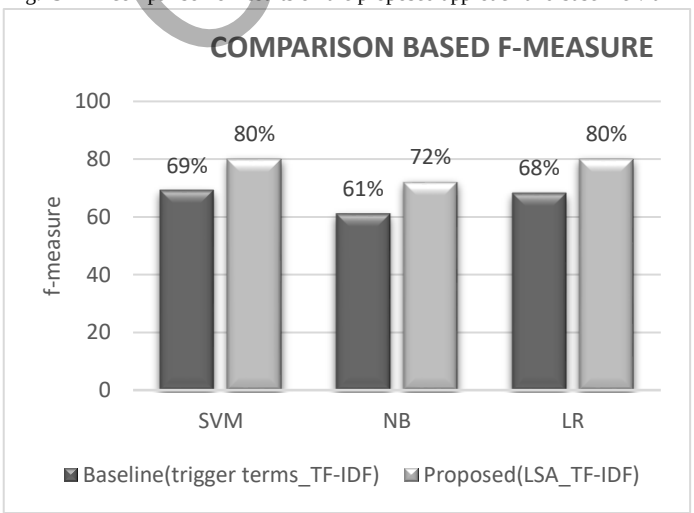


Fig. 5 A comparison of results on the proposed approach and baseline via TF



results

Table 8: A comparison of results on the proposed approach and baseline via TF-IDF results

	SVM	**NB**	**LR**
Baseline	0.69	0.61	0.68
Proposed approach	0.80	0.72	0.80

Fig. 6 A comparison of results on the proposed approach and baseline via TF-IDF results

Apart from the traditional baseline which utilized conventional approaches such as SVM, NB, and others, it is necessary to compare the proposed method against recent methods that employed deep learning techniques. In fact, (Lee et al. 2017) have used a deep learning approach of CNN to extract ADRs and acquired an f-measure of 64.5%. they used different Twitter data set in the PSB 2016 Social Media Shared Task. Comparing such results against the obtained ones by the proposed method reveals that the proposed method is still Competitive.

However, other studies such as (Wang et al. 2019) whom utilized much sophisticated deep learning approaches, have obtained an f-measure higher the proposed method as 84.4%. Yet, their approaches were requiring a pre-trained data of embedding for the medical words. Considering the LSA that has been utilized by the proposed method, it is clear that the proposed method is still considered to be less complicated.

V. Conclusion

This study proposed an LSA for detecting ADRs. These LSA were compared with the baseline ones by using one benchmark dataset. Experiments involved three classifiers, namely, SVM, NB and LR. The proposed LSA achieved higher results than the baseline ones when TF and LR classification were used. Further studies on feature types would facilitate the process of detecting ADRs.

Acknowledgments

First and foremost, praise is to Almighty Allah for all His blessings All praise the Almighty. I would like to express my sincere appreciation to my supervisors Prof. Nazlia Omar for her supports and encouragement for me to writing of this thesis.

References

Ebrahimi, M., Yazdavar, A. H., Salim, N. & Eltyeb, S. 2016. Recognition of side effects as implicit-opinion words in drug reviews. *Online Information Review* 40(7). 1018-1032.

Elhadad, M. K., Li, K. F. & Gebali, F. 2019. Sentiment Analysis of Arabic and English Tweets. Workshops of the International Conference on Advanced Information

Networking and Applications. Springer, Cham. 334-348.

Emadzadeh, E., Sarker, A., Nikfarjam, A. & Gonzalez, G. 2017. Hybrid semantic analysis for mapping adverse drug reaction mentions in tweets to medical terminology. *AMIA Annual Symposium Proceedings*, 679.

Froud, H., Lachkar, A. & Ouatik, S. A. 2013. Arabic text summarization based on latent semantic analysis to enhance Arabic documents clustering. *arXiv preprint arXiv:1302.1612*.

Hutchison, P. D., Daigle, R. J. & George, B. 2018. Application of latent semantic analysis in AIS academic research. *International Journal of Accounting Information Systems* 31. 83-96.

Islam, M. M. & Hoque, A. L. 2010. Automated essay scoring using generalized latent semantic analysis. *Computer and Information Technology (ICCIT), 2010 13th International Conference on*, 358-363.

Kiritchenko, S., Mohammad, S. M., Morin, J. & de Bruijn, B. 2018. NRC-Canada at SMM4H shared task: classifying Tweets mentioning adverse drug reactions and medication intake. *arXiv preprint arXiv:1805.04558*.

Kiritchenko, S., Zhu, X. & Mohammad, S. M. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50. 723-762.

Lee, K., Qadir, A., Hasan, S. A., Datla, V., Prakash, A., Liu, J. & Farri, O. 2017. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. *Proceedings of the 26th International Conference on World Wide Web*, 705-714.

Liu, Y., Bi, J.-W. & Fan, Z.-P. 2017. Ranking products through online reviews: A method based on sentiment analysis technique and intuitionistic fuzzy set theory. *Information Fusion* 36. 149-161.

Moghaddam, S. & Ester, M. 2011. AQA: aspect-based opinion question answering. *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, 89-96.

Mohammad Yousef, R. N., Tiun, S. & Omar, N. 2019. Extended Trigger Terms for Extracting Adverse Drug Reactions in Social Media Texts. *Journal of Computer Science* 15(6). 873-879.

Montgomery, D. C., Peck, E. A. & Vining, G. G. 2015. *Introduction to linear regression analysis*. John Wiley & Sons.

Nikfarjam, A., Sarker, A., O'connor, K., Ginn, R. & Gonzalez, G. 2015. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association* 22(3). 671-681.

Pain, J., Levacher, J., Quinqunel, A. & Belz, A. 2016. Analysis of Twitter data for postmarketing surveillance in pharmacovigilance.

Plachouras, V., Leidner, J. L. & Garrow, A. G. 2016. Quantifying self-reported adverse drug events on Twitter: signal and topic analysis. *Proceedings of the 7th 2016 International Conference on Social Media & Society*, 6.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3). 130-137.

Wang, C.-S., Lin, P.-J., Cheng, C.-L., Tai, S.-H., Yang, Y.-H. K. & Chiang, J.-H. 2019. Detecting Potential Adverse Drug Reactions Using a Deep Neural Network Model. *Journal of medical Internet research* 21(2). e11016.

Yates, A. & Goharian, N. 2013. ADRTrace: detecting expected and unexpected adverse drug reactions from user reviews on social media sites. *European Conference on Information Retrieval*, 816-819.