View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Crossref

Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2016, Article ID 3483528, 16 pages
http://dx.doi.org/10.1155/2016/3483528

Hindawi

Research Article

A New Data Representation Based on Training Data Characteristics to Extract Drug Name Entity in Medical Text

Mujiono Sadikin,[1,2] Mohamad Ivan Fanany,[2] and T. Basaruddin[2]

[1]*Faculty of Computer Science, Universitas Mercu Buana, l. Meruya Selatan No. 1, Kembangan, Jakarta Barat 11650, Indonesia*
[2]*Machine Learning and Computer Vision Laboratory, Faculty of Computer Science, Universitas Indonesia, Depok, West Java 16424, Indonesia*

Correspondence should be addressed to Mujiono Sadikin; mujiono.sadikin@mercubuana.ac.id

Received 27 May 2016; Revised 8 August 2016; Accepted 18 September 2016

Academic Editor: Trong H. Duong

Copyright © 2016 Mujiono Sadikin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One essential task in information extraction from the medical corpus is drug name recognition. Compared with text sources come from other domains, the medical text mining poses more challenges, for example, more unstructured text, the fast growing of new terms addition, a wide range of name variation for the same drug, the lack of labeled dataset sources and external knowledge, and the multiple token representations for a single drug name. Although many approaches have been proposed to overwhelm the task, some problems remained with poor *F*-score performance (less than 0.75). This paper presents a new treatment in data representation techniques to overcome some of those challenges. We propose three data representation techniques based on the characteristics of word distribution and word similarities as a result of word embedding training. The first technique is evaluated with the standard NN model, that is, MLP. The second technique involves two deep network classifiers, that is, DBN and SAE. The third technique represents the sentence as a sequence that is evaluated with a recurrent NN model, that is, LSTM. In extracting the drug name entities, the third technique gives the best *F*-score performance compared to the state of the art, with its average *F*-score being 0.8645.

1. Introduction

The rapid growth of information technology provides rich text data resources in all areas, including the medical field. An abundant amount of medical text data can be used to obtain valuable information for the benefit of many purposes. The understanding of drug interactions, for example, is an important aspect of manufacturing new medicines or controlling drug distribution in the market. The process to produce a medicinal product is an expensive and complex task. In many recent cases, however, many drugs are withdrawn from the market when it was discovered that the interaction between the drugs is hazardous to health [1].

Information, or objects extraction, from an unstructured text document, is one of the most challenging studies in the text mining area. The difficulties of text information extraction keep increasing due to the increasing size of corpora, continuous growth of human's natural language, and the unstructured formatted data [2]. Among such valuable information are medical entities such as drug name, compound, and brand; disease names and their relations, such as drug-drug interaction and drug-compound relation. We need a suitable method to extract such information. To embed those abundant data resources, however, many problems have to be tackled, for example, large data size, unstructured format, choosing the right NLP, and the limitation of annotated datasets.

More specific and valuable information contained in medical text data is a drug entity (drug name). Drug name recognition is a primary task of medical text data extraction since the drug finding is the essential element in solving other information extraction problems [3, 4]. Among derivative work of drug name extractions are drug-drug interaction [5], drug adverse reaction [6], or other applications (information

retrieval, decision support system, drug development, or drug discovery) [7].

Compared to other NER (name entity recognition) tasks, such as PERSON, LOCATION, EVENT, or TIME, drug name entity recognition faces more challenges. First, the drug name entities are usually unstructured texts [8] where the number of new entities is quickly growing over time. Thus, it is hard to create a dictionary which always includes the entire lexicon and is up-to-date [9]. Second, the naming of the drug also widely varies. The abbreviation and acronym increase the difficulties in determining the concepts referred to by the terms. Third, many drug names contain a combination of nonword and word symbols [10]. Fourth, the other problem in drug name extraction is that a single drug name might be represented by multiple tokens [11]. Due to the complexity in extracting multiple tokens for drugs, some researchers such as [12] even ignore that case in the MedLine and DrugBank training with the reason that the multiple tokens drug is only 18% of all drug names. It is different with another domain; that is, entity names in the biomedical field are usually longer. Fifth, in some cases, the drug name is a combination of medical and general terms. Sixth, the lack of the labelled dataset is another problem; it has yet to be solved by extracting the drug name entities.

This paper presents three data representation techniques to extract drug name entities contained in the sentences of medical texts. For the first and the second techniques, we created an instance of the dataset as a tuple, which is formed from 5 vectors of words. In the first technique, the tuple was constructed from all sentences treated as a sequence, whereas in the second technique the tuple is made from each sentence treated as a sequence. The first and second techniques were evaluated with the standard MLP-NN model which is performed in the first experiment. In the second experiment, we use the second data representation technique which is also applied to the other NN model, that is, DBN and SAE. The third data representation, which assumes the text as sequential entities, was assessed with the recurrent NN model, LSTM. Those three data representation techniques are based on the word2vec value characteristics, that is, their cosine and the Euclidean distance between the vectors of words.

In the first and second techniques, we apply three different scenarios to select the most possible words which represent the drug name. The scenarios are based on the characteristics of training data, that is, drug words distribution that is usually assumed to have a smaller frequency of appearance in the dataset sentences. The drug name candidate selections are as follows. In the first case, all test dataset is taken. In the second case, 2/3 of all test dataset is selected. In the third case, x/y ($x < y$) of the test dataset (where x and y are arbitrary integer numbers) are selected after clustering the test dataset into y clusters.

In the third experiment, based on the characteristics of the resulting word vectors of the trained word embedding, we formulate a sequence data representation applied to RNN-LSTM. We used the Euclidian distance of the current input to the previous input as an additional feature besides its vector of words. In this study, the vector of words is provided by word embedding methods proposed by Mikolov et al. [13].

Our main important contributions in this study are

(1) the new data representation techniques which do not require any external knowledge nor handcrafted features,

(2) the drug extraction techniques based on the words distribution contained in the training data.

Our proposed method is evaluated on DrugBank and MedLine medical open dataset obtained from SemEval 2013 Competition task 9.1; see https://www.cs.york.ac.uk/semeval-2013/task9/, which is also used by [11, 12, 14]. The format of both medical texts is in English where some sentences contain drug name entities. In extracting drug entity names from the dataset, our data representation techniques give the best performance with F-score values 0.687 for MLP, 0.6700 for DBN, and 0.682 for SAE, whereas the third technique with LSTM gives the best F-score, that is, 0.9430. The average F-score of the third technique is 0.8645, that is, the best performance compared to the other previous methods.

By applying the data representation techniques, our proposed approach provides at least three advantages:

(1) The capability to identify multiple tokens as a single name entity

(2) The ability to deal with the absence of any external knowledge in certain languages

(3) No need to construct any additional features, such as characters type identification, orthography feature (lowercase or uppercase identification), or token position

The rest of the sections of this paper are organized as follows: Section 2 explains some previous works dealing with name entity (and drug name as well) extraction from medical text sources. The framework, approach, and methodology to overcome the challenges of drug name extraction are presented in Section 3. The section also describes dataset materials and experiment scenarios. Section 4 discusses the experiment results and its analysis while Section 5 explains the achievement, the shortcoming, and the prospects of this study. The section also describes several potential explorations for future research.

2. Related Works

The entity recognition in a biomedical text is an active research, and many methods have been proposed. For example, Pal and Gosal [9] summarize their survey on various entity recognition approaches. The approaches can be categorized into three models: dictionary based, rule-based, and learning based methods [2, 8]. A dictionary based approach uses a list of terms (term collection) to assist in predicting which targeted entity will be included in the predicted group. Although their overall precision is more accurate, their recall is poor since they anticipate less new terms. The rule-based approach defines a certain rule which describes such pattern

formation surrounding the targeted entity. This rule can be a syntactic term or lexical term. Finally, the learning approach is usually based on statistical data characteristics to build a model using machine learning techniques. The model is capable of automatic learning based on positive, neutral, and negative training data.

Drug name extraction and their classification are one of the challenges in the Semantic Evaluation Task (SemEval 2013). The best-reported performance for this challenge was 71.5% in *F*-score [15]. Until now the studies to extract drug names still continue and many approaches have been proposed. CRF-based learning is the most common method utilized in the clinical text information extraction. CRF is used by one of the best [11] participants in SemEval challenges in the clinical text (https://www.cs.york.ac.uk/semeval-2013/). As for the use of external knowledge aimed at increasing the performance, the author [11] uses ChEBI (Chemical Entities of Biological Interest), that is, a dictionary of small molecular entities. The best achieved performance is 0.57 in *F*-score (for the overall dataset).

A hybrid approach model, which combines statistical learning and dictionary based, is proposed by [16]. In their study, the author utilizes word2vec representation, CRF learning model, and DINTO, a drug ontology. With this word2vec representation, targeted drug is treated as a current token in context windows which consists of three tokens on the left and three tokens on the right. Additional features are included in the data representation such as pos tags, lemma in the windows context, and an orthography feature as uppercase, lowercase, and mixed cap. The author also used Wikipedia text as an additional resource to perform word2vec representation training. The best *F*-score value in extracting the drug name provided by the method is 0.72.

The result of CRF-based active learning, which is applied to NER BIO (Beginning, Inside, Output) annotation token for extracting name entity in the clinical text, is presented in [17]. The framework of this active learning approach is a sequential process: initial model generation, querying, training, and iteration. The CRF Algorithm BIO approach was also studied by Ben Abacha et al. [14]. The features for the CRF algorithm are formulated based on token and linguistics feature and semantic feature. The best *F*-score achieved by this proposed method is 0.72.

Korkontzelos et al. studied a combination of aggregated classifier, maximum entropy-multinomial classifier, and handcrafted feature to extract drug entity [4]. They classified drug and nondrug based on the token features formulation such as tokens windows, the current token, and 8 other handcrafted features.

Another approach for discovering valuable information from clinical text data that adopts event-location extraction model was examined by Bjorne et al. [12]. They use an SVM classifier to predict drug or nondrug entity which is applied to DrugBank dataset. The best performance achieved by their method is 0.6 in *F*-score. The drawback of their approach is that it only deals with a single token drug name.

To overcome the ambiguity problem in NER mined from a medical corpus, a segment representation method has also been proposed by Keretna et al. [8]. Their approach treats each word as belonging to three classes, that is, NE, not NE, and an ambiguous class. The ambiguity of the class member is determined by identifying whether the word appears in more than one context or not. If so, this word falls into the ambiguous class. After three class segments are found, each word is then applied to the classifier learning. Related to their approach, in our previous work, we propose pattern learning that utilizes the regular expression surrounding drug names and their compounds [18]. The performance of our method is quite good with the average *F*-score being 0.81 but has a limitation in dealing with more unstructured text data.

In summarizing the related previous works on drug name entity extraction, we noted some drawbacks which need to be addressed. In general, almost all state-of-the-art methods work based on ad hoc external knowledge which is not always available. The requirement of the handcrafted feature is another difficult constraint since not all datasets contain such feature. An additional challenge that remained unsolved by the previous works is the problem of multiple tokens representation for a single drug name. This study proposes a new data representation technique to handle those challenges.

Our proposed method is based only on the data distribution pattern and vector of words characteristics, so there is no need for external knowledge nor additional handcrafted features. To overcome the multiple tokens problem, we propose a new technique which treats a target entity as a set of tokens (a tuple) at once rather than treating the target entity as a single token surrounded by other tokens such as those used by [16] or [19]. By addressing a set of the tokens as a single sample, our proposed method can predict whether a set of tokens is a drug name or not. In our first experiment, we evaluate the first and second data representation techniques and apply MLP learning model. In our second scenario, we choose the the second technique which gave the best result with MLP and apply it to two different machine learning methods: DBN and SAE. In our third experiment, we examined the third data representation technique which utilizes the Euclidian distance between successive words in a certain sentence of medical text. The third data representation is then fed into an LSTM model. Based on the resulting *F*-score value, the second experiment gives the best performance.

3. Method and Material

3.1. Framework. In this study, using the word2vec value characteristics, we conducted three experiments based on different data representation techniques. The first and second experiment examine conventional tuple data representation, whereas the third experiment examines sequence data representation. We describe the organization of these three experiments in this section. In general, the proposed method to extract drug name entities in this study consists of two main phases. The first phase is a data representation to formulate the feature representation. In the second phase, model training, testing, and their evaluation are then conducted to evaluate the performance of the proposed method.

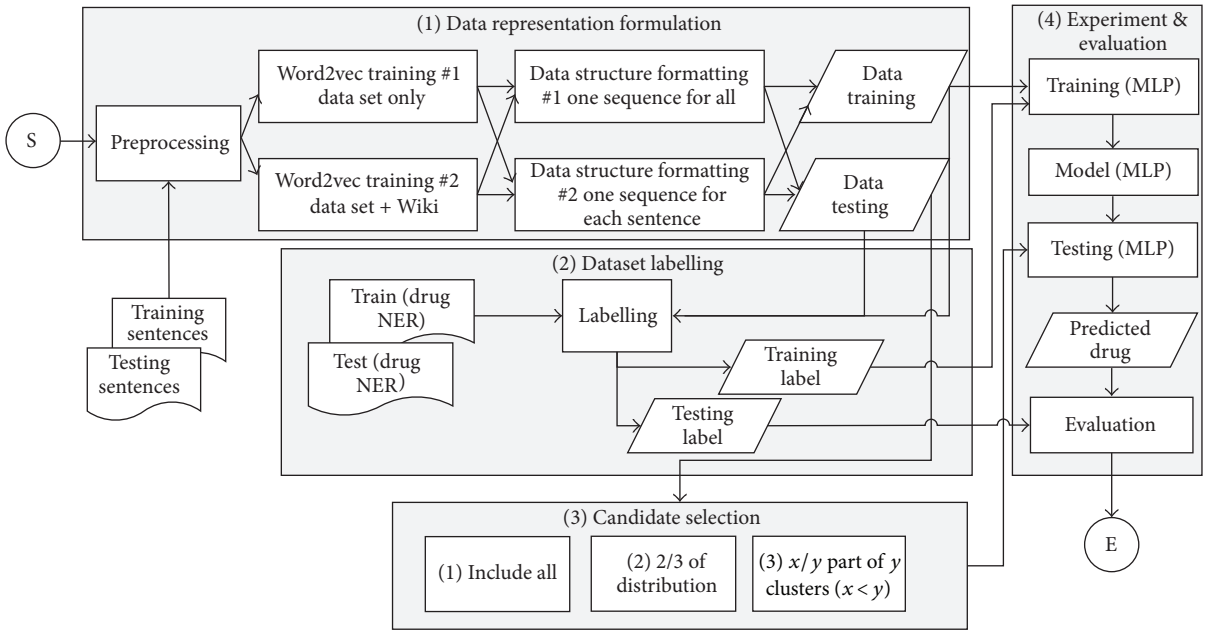


Figure 1: Proposed approach framework of the first experiment.

The proposed method of the first experiment consists of 4 steps (see Figure 1). The first step is a data representation formulation. The output of the first step is the tuples of training and testing dataset. The second step is dataset labelling which is applied to both testing and training data. The step provides the label of each tuple. The third step is the candidate selection which is performed to minimize the noises since the actual drug target quantity is far less compared to nondrug name. In the last step, we performed the experiment with MLP-NN model and its result evaluation. The detailed explanation of each step is explained in Sections 3.4, 3.6, and 3.9, whereas Sections 3.2 and 3.3 describe training data analysis as the foundation of this proposed method. As a part of the first experiment, we also evaluate the impact of the usage of the Euclidean distance average as the model's regularization. This regularization term is described in Section 3.7.1.

The framework of the second experiment which involves DBN and SAE learning model to the second data representation technique is illustrated in Figure 2. In general, the steps of the second experiment are similar to the first one, with its differences being the data representation used and the learning model involved. In the second experiment, the second technique is used only with DBN and SAE as the learning model.

The framework of the third experiment using the LSTM is illustrated in Figure 3. There are tree steps in the third experiment. The first step is sequence data representation formulation which provides both sequence training data and testing data. The second step is data labelling which generates the label of training and testing data. LSTM experiment and its result evaluation are performed in the third step. The detailed description of these three steps is presented in Sections 3.4, 3.4.3, and 3.9 as well.

3.2. Training Data Analysis. Each of the sentences in the dataset contains four data types, that is, drug, group, brand, and drug-n. If the sentence contains none of those four types, the type value is null. In the study, we extracted drug and drug-n. Overall in both DrugBank and MedLine datasets, the quantity of drug name target is far less compared to the nondrug target. Segura-Bedmar et al. [15] present the first basic statistics of the dataset. A more detailed exploration regarding token distribution in the training dataset is described in this section. The MedLine sentences training dataset contains 25.783 single tokens, which consist of 4.003 unique tokens. Those tokens distributions are not uniform but are dominated by a small part of some unique tokens. If all of the unique tokens are arranged and ranked based on the most frequent appearances in the sentences, the quartile distribution will have the following result presented in Figure 4. $Q1$ represents token numbers 1 to 1001 whose total of frequency is 20.688. $Q2$ represents token numbers 1002 to 2002 whose total of frequency is 2.849. $Q3$ represents token numbers 2003 to 3002 whose total of frequency is 1.264, and $Q4$ represents token numbers 3003 to 4003 whose total of frequency is 1.000. The figure shows that the majority of appearances are dominated by only a small amount of the total tokens.

Further analysis of the dataset tokens shows that most of the drug names of the targeted token rarely appear in the dataset. When we divide those token collections into three partitions based on their sum of frequency, as presented in Table 1, it is shown that all of the drug name entities targeted are contained in 2/3 part with less frequent appearances of each token (a unique token in the same sum of frequency). A similar pattern of training data token distribution also emerged in the DrugBank dataset as illustrated in Figure 5 and Table 2. When we look into specific token distributions,

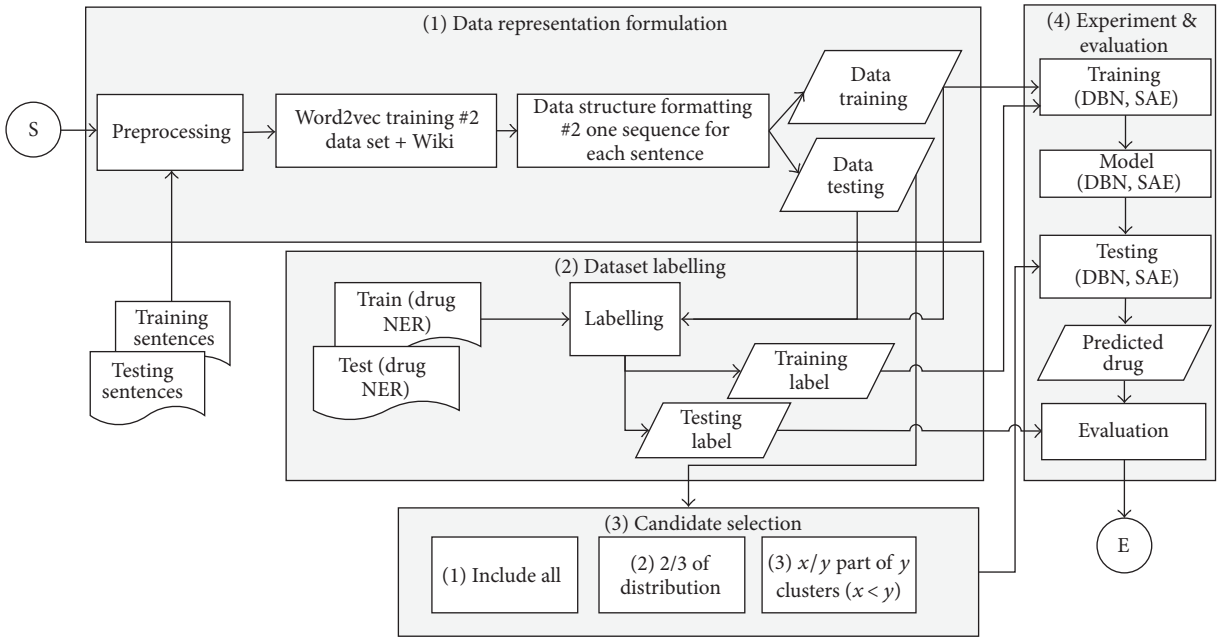


Figure 2: Proposed approach framework of the second experiment.

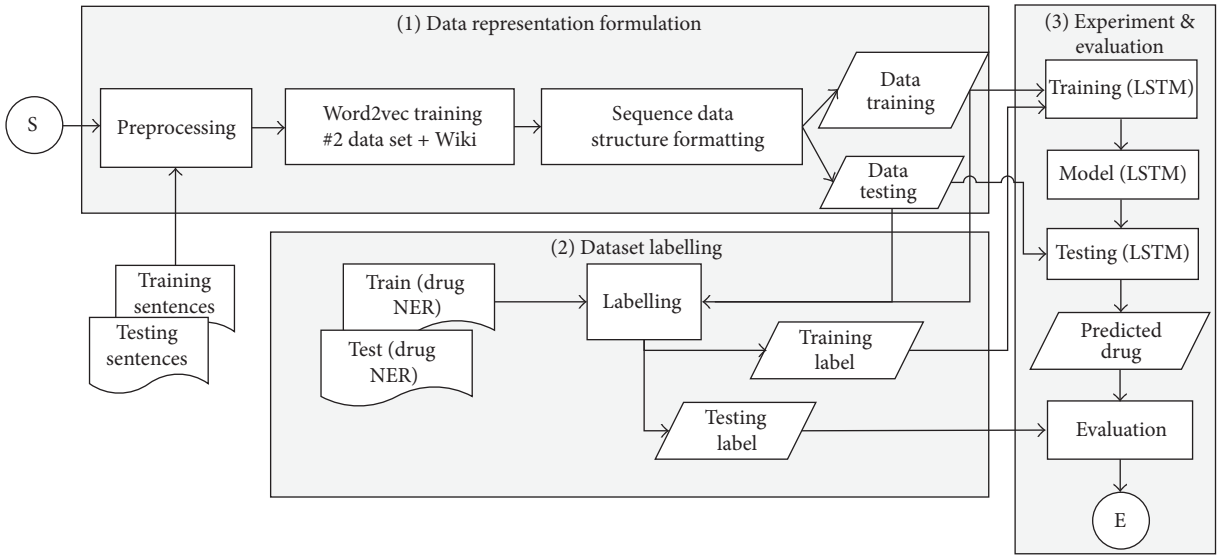


Figure 3: Proposed approach framework of the third experiment.

the position of most of the drug name targets is in the third part, since the most frequently appearing words in the first and the second parts are the most common words such as stop words ("of", "the", "a", "end", "to", "where", "as", "from", and such kind of words) and common words in medical domain such as "administrator", "patient", "effect", and "dose".

3.3. Word Embedding Analysis. To represent the dataset we utilized the word embedding model proposed by Mikolov et al. [13]. We treated all of the sentences as a corpus after the training dataset and testing dataset were combined. The used word2vec training model was the CBOW (Continuous Bag Of Words) model with context window length 5 and the vector dimension 100. The result of the word2vec training is the representation of word in 100 dimension row vectors. Based on the row vector, the similarities or dissimilarities between words can be estimated. The description below is the analysis summary of word2vec representation result which is used as a base reference for the data representation technique and the experiment scenarios. By taking some sample of drug targets and nondrug vector representation, it is shown that drug word has more similarities (cosine distance) to another drug than to nondrug and vice versa. Some of those samples are illustrated in Table 3. We also computed the Euclidean

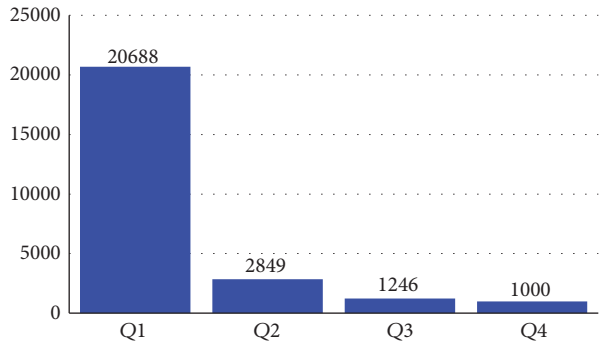


Figure 4: Distribution of MedLine train dataset token.

Table 1: The frequency distribution and drug target token position, MedLine.

1/3#	Σ sample	Σ frequency	Σ single token of drug entity
1	28	8,661	—
2	410	8,510	50
3	3,563	8,612	262

Table 2: The frequency distribution and drug target token position, DrugBank.

1/3#	Σ sample	Σ frequency	Σ single token of drug entity
1	27	33,538	—
2	332	33,463	33
3	5,501	33,351	920

distance between all of the words. Table 4 shows the average of Euclidean distance and cosine distance between drug-drug, drug-nondrug, and nondrug-nondrug. These values of the average distance show us that, intuitively, it is feasible to group the collection of the words into drug group and nondrug group based on their vector representations value.

3.4. Feature Representation, Data Formatting, and Data Labelling. Based on the training data and word embedding analysis, we formulate the feature representation and its data formatting. In the first and second techniques, we try to overcome the multiple tokens drawback left unsolved in [12] by formatting single input data as an N-gram model with $N = 5$ (one tuple piece of data consists of 5 tokens) to accommodate the maximum token which acts as a single drug entity target name. The tuples were provided from the sentences of both training and testing data. Thus, we have a set of tuples of training data and a set of tuples of testing data. Each tuple was treated as a single input.

To identify a single input, whether it is a nondrug or drug target, we use a multiclassification approach which classifies the single input into one of six classes. Class 1 represents nondrug whereas the other classes represent drug target which also identified how many tokens (words) perform the drug target. To identify which class a certain tuple belongs to the following is determined: The drug tuple is the tuple whose first token (token-1) is the drug type. If token-1 is not a drug,

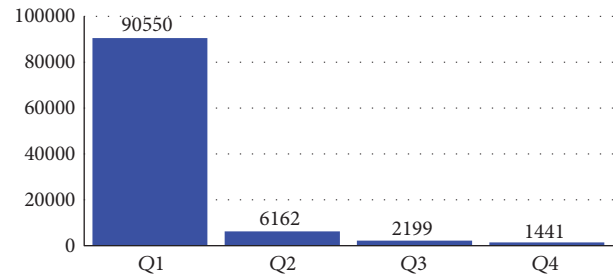


Figure 5: Distribution of DrugBank train dataset token.

regardless of whatever the rest of the 4 tokens are, then the tuple is classified as no drug. This kind of tuple is identified as class 1. If token-1 is a drug and token-2 is not a drug, regardless of the last 3 tokens, the tuple will be identified as class 2 and so on.

Since we only extracted the drug entity, we ignored the other token types, whether it is a group, brand, or another common token. To provide the label of each tuple, we only use the drug and drug-n types as the tuple reference list. In general, if the sequence of token in each tuple in dataset contains the sequence which is exactly the same with one of tuple reference list members, then the tuple in dataset is identified as drug entity. The detail of the algorithm used to provide the label of each tuple in both training data and testing data is described in Algorithm 1.

We proposed two techniques in constructing the tuple set of the sentences. The first technique treats all sentences as one sequence, whereas in the second technique, each sentence is processed as one sequence. The first and the second techniques are evaluated with MLP, DBN, and SAE model. The third technique treats the sentences of dataset as a sequence where the occurrence of the current token is influenced by the previous one. By treating the sentence as a sequence not only in the data representation but also in the classification and recognition process, the most suitable model to be used is RNN. We applied RNN-LSTM to the third technique.

3.4.1. First Technique. The first dataset formatting (one sequence for all sentences) is performed as follows. In the first step, all sentences in the dataset are formatted as a token sequence. Let the token sequence be

$$t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 \cdots t_n \quad (1)$$

with n being number of tokens in the sequences; then the dataset format will be

$$t_1 t_2 t_3 t_4 t_5; t_2 t_3 t_4 t_5 t_6; \cdots t_{n-4} t_{n-3} t_{n-2} t_{n-1} t_n. \quad (2)$$

A sample of sentences and their drug names are presented in Table 5. Taken from DrugBank training data Table 5 is the raw data of 3 samples with three relevant fields, that is, sentences, character drug position, and the drug name. Table 6 illustrates a portion of the dataset and its label as the result of the raw data in Table 5. Referring to the drug-n name field in the dataset, dataset number 6 is identified as a drug,

Table 3: Some of the cosine distance similarities between two kinds of words.

Word 1	Word 2	Similarities (cosine dist)	Remark
dilantin	tegretol	0.75135758	drug-drug
phenytoin	dilantin	0.62360351	drug-drug
phenytoin	tegretol	0.51322415	drug-drug
cholestyramine	dilantin	0.24557819	drug-drug
cholestyramine	phenytoin	0.23701277	drug-drug
administration	patients	0.20459694	non-drug - non-drug
tegretol	may	0.11605539	drug - non-drug
cholestyramine	patients	0.08827197	drug - non-drug
evaluated	end	0.07379115	non-drug - non-drug
within	controlled	0.06111103	non-drug - non-drug
cholestyramine	evaluated	0.04024139	drug - non-drug
dilantin	end	0.02234770	drug - non-drug

Table 4: The average of Euclidean distance and cosine similarities between groups of words.

Word group	Euclidean dist. avg	Cosine dist. avg
drug - non-drug	0.096113798	0.194855980
non-drug - non-drug	0.094824332	0.604091044
drug-drug	0.093840800	0.617929002

Table 5: Sample of DrugBank sentences and their drug name target.

Sentence	Drug position	Drug name
modification of surface histidine residues abolishes the cytotoxic activity of clostridium difficile toxin a	79–107	clostridium difficile toxin a
antimicrobial activity of ganoderma lucidum extract alone and in combination with some antibiotics.	26–50	ganoderma lucidum extract
on the other hand, surprisingly, green tea gallocatechins, (−)-epigallocatechin-3-o-gallate and theasinensin a, potently enhanced the promoter activity (182 and 247% activity at 1 microm, resp.).	33–56	green tea gallocatechins

whereas the others are classified as a nondrug entity. The complete label illustration of the dataset provided by the first technique is presented in Table 7. As described in Section 3.4, the value of vector dimension for each token is 100. Therefore, for single data, it is represented as $100 * 5 = 500$ lengths of a one-dimensional vector.

3.4.2. Second Technique. The second technique is used for treating one sequence that comes from each sentence of the dataset. With this treatment, we added special characters $*$, as padding, to the last part of the token when its dataset length is less than 5. By applying the second technique the first sentence of the sample provided a dataset as illustrated in Table 8.

3.4.3. Third Technique. Naturally, the NLP sentence is a sequence in which the occurrence of the current word is conditioned by the previous one. Based on the word2vec value analysis, it is shown that intuitively we can separate the drug word and nondrug word by their Euclidean distance. Therefore, we used the Euclidean distance between the current words with the previous one to represent the influence. Thus, each current input x_i is represented by $[xv_i xd_i]$ which is the concatenation of word2vec value xv_i and its Euclidian distance to the previous one, xd_i. Each x is the row vector with the dimension length being 200, the first 100 values are its word2vector, and the rest of all 100 values are the Euclidian distance to the previous. For the first word all values of xd_i are 0. With the LSTM model, the task to extract the drug name from the medical data text is the binary classification applied to each word of the sentence. We formulate the word sequence and its class as described in Table 9. In this experiment, each word that represents the drug name is identified as class 1, such as "plenaxis", "cytochrome", and "p-450", whereas the other words are identified by class 0.

3.5. Wiki Sources. In this study we also utilize Wikipedia as the additional text sources in word2vec training as used by [16]. The Wiki text addition is used to evaluate the impact of the training data volume in improving the quality of word's vector.

3.6. Candidates Selection. The tokens as the drug entities target are only a tiny part of the total tokens. In MedLine dataset, 171 of 2.000 tokens (less than 10%) are drugs, whereas in DrugBank, the number of drug tokens is 180 of 5.252 [15]. So the major part of these tokens is nondrug and other noises such as a stop word and special or numerical characters. Based on this fact, we propose a candidate selection step to eliminate those noises. We examine two mechanisms in the candidate selection. The first is based on token distribution. The second is formed by selecting x/y part of the clustering result of data test. In the first scenario, we only used 2/3 of the token, which appears in the lower 2/3 part of the total token. This is presented in Tables 1 and 2. However, in the second

```
Result: Labelled dataset
Input: array of tuple, array of drug;
output: array of label {Array of drug contains list of drug and drug-n only};
label[] <= 1 Initialization;
for each t in tuple do
    for each d in drug do
        if length (d) = 1 then
            if t[1] = d[1] then
                //match 1 token drug;
                label <= 2, break, exit from for each d in drug;
            else
            end
        else
            if length (d) = 2 then
                if t[1] = d[1] and t[2] = d[2] then
                    //match 2 tokens drug;
                    label <= 3, break, exit from for each d in drug;
                else
                end
            else
                if length (d) = 3 then
                    if t[1] = d[1] and t[2] = d[2] and t[3] = d[3] then
                        label <= 4, break, exit from for each d in drug;
                    else
                    end
                else
                    if length (d) = 4 then
                        if t[1] = d[1] and t[2] = d[2] and t[3] = d[3] and t[4] = d[4] then
                            label <= 5, break, exit from for each d in drug;
                        else
                        end
                    else
                        if length (d) = 5 then
                            if t[1] = d[1] and t[2] = d[2] and t[3] = d[3] and t[4] = d[4] and t[5] = d[5] then
                                label <= 6, break, exit from for each d in drug;
                            else
                            end
                        else
                        end
                    end
                end
            end
        end
    end
end
```

Algorithm 1: Dataset labelling.

Table 6: A portion of the dataset formulation as the results of DrugBank sample with first technique.

Dataset number	Token-1	Token-2	Token-3	Token-4	Token-5	Label
1	modification	of	surface	histidine	residues	1
2	of	surface	histidine	residues	abolishes	1
3	surface	histidine	residues	abolishes	the	1
4	histidine	residues	abolishes	the	cytotoxic	1
5	the	cytotoxic	activity	of	clostridium	1
6	*clostridium*	*difficile*	*toxin*	*a*	antimicrobial	5
7	difficile	toxin	a	antimicrobial	activity	1

Table 7: First technique of data representation and its label.

Token-1	Token-2	Token-3	Token-4	Token-5	Label
"plenaxis"	"were"	"performed"	"cytochrome"	"p-450"	2
"testosterone"	"concentrations"	"just"	"prior"	"to"	2
"beta-adrenergic"	*"antagonists"*	"and"	"alpha-adrenergic"	"stimulants,"	3
"carbonic"	*"anhydrase"*	"inhibitors,"	"concomitant"	"use"	3
"sodium"	*"polystyrene"*	*"sulfonate"*	"should"	"be"	4
"sodium"	*"acid"*	*"phosphate"*	"such"	"as"	4
"clostridium"	*"difficile"*	*"toxin"*	*"a"*	"—"	5
"nonsteroidal"	*"anti"*	*"inflammatory"*	*"drugs"*	"and"	5
"casein"	*"phosphopeptide-amorphous"*	*"calcium"*	*"phosphate"*	*"complex"*	6
"studies"	"with"	"plenaxis"	"were"	"performed."	1
"were"	"performed."	"cytochrome"	"p-450"	"is"	1

Table 8: Second technique of data representation and its label.

Token-1	Token-2	Token-3	Token-4	Token-5	Label
"modification"	"of"	"surface"	"histidine"	"residues"	1
"of"	"surface"	"histidine"	"residues"	"abolishes"	1
surface	histidine	residues	abolishes	the	1
"histidine"	"residues"	"abolishes"	"the"	"cytotoxic"	1
"the"	"cytotoxic"	"activity"	"of"	"clostridium"	1
"clostridium"	*"difficile"*	*"toxin"*	"a"	"∗"	5
"difficile"	"toxin"	"a"	"∗"	"∗"	1
"a"	"atoxin"	"∗"	"∗"	"∗"	1
"toxic"	"∗"	"∗"	"∗"	"∗"	1

mechanism we selected x/y ($x < y$) which is a part of total token after the tokens are clustered into y clusters.

3.7. Overview of NN Model

3.7.1. MLP. In the first experiment, we used multilayer perceptron NN to train the model and evaluate the performance [20]. Given a training set of m examples, then the overall cost function can be defined as

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m} J\left(W,b;x^i,y^i\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{nl-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl-1}\left(W_{j_i}^{(l)}\right)^2,$$
$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{wb}\left(x^{(\mathrm{i})}\right)-y^i\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{nl-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl-1}\left(W_{j_i}^{(l)}\right)^2. \quad (3)$$

In the definition of $J(W,b)$, the first term is an average sum-of-squares error term, whereas the second term is a regularization term which is also called a weight decay term. In this experiment we use three kinds of regularization: #0, L0 with $\lambda = 0$, #1, L1 with $\lambda = 1$, and #2 with $\lambda =$ the average of Euclidean distance. We computed the L2's λ based on the word embedding vector analysis that drug target and nondrug can be distinguished by looking at their Euclidean distance. Thus, for L2 regularization, the parameter is calculated as

$$\lambda = \frac{1}{n*(n-1)}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\mathrm{dist}\left(x^i,x^j\right), \quad (4)$$

where $\mathrm{dist}(x^i,x^j)$ is the Euclidean distance of x^i and x^j.

The model training and testing are implemented by modifying the code from [21] which can be downloaded at https://github.com/rasmusbergpalm/DeepLearnToolbox.

3.7.2. DBN. DBN is a learning model composed of two or more stacked RBMs [22, 23]. An RBM is an undirected graph learning model which associates with Markov Random Fields (MRF). In the DBN, the RBM acts as feature extractor where the pretraining process provides initial weights values to be fine-tuned in the discriminative process in the last layer. The last layer may be formed by logistic regression or any standard discriminative classifiers [23]. RBM was originally developed for binary data observation [24, 25]. It is a popular type of unsupervised model for binary data [26, 27]. Some derivatives of RBM models are also proposed to tackle continuous/real values suggested in [28, 29].

Table 9: Third technique of data representation and its label.

Sent.#1	Class	0	0	0	0	1	0	0
	Word	"drug"	"interaction"	"studies"	"with"	"plenaxis"	"were"	"performed"
Sent.#2	Class	1	1	0	0	0	0	0
	Word	"cytochrome"	"p-450"	"is"	"not"	"known"	"in"	"the"

3.7.3. SAE. An autoencoder (AE) neural network is one of the unsupervised learning algorithms. The NN tries to learn a function $h(w, x) \approx x$. The autoencoder NN architecture also consists of input, hidden, and output layers. The particular characteristic of the autoencoder is that the target output is similar to the input. The interesting structure of the data is estimated by applying a certain constraint to the network, which limits the number of hidden units. However, when the number of hidden units has to be larger, it can be imposed with sparsity constraints on the hidden units [30]. The sparsity constraint is used to enforce the average value of hidden unit activation constrained to a certain value. As used in the DBN model, after we trained the SAE, the trained weight was used to initialize the weight of NN for the classification.

3.7.4. RNN-LSTM. RNN (Recurrent Neural Network) is an NN, which considers the previous input in determining the output of the current input. RNN is powerful when it is applied to the dataset with a sequential pattern or when the current state input depends on the previous one, such as the time series data, sentences of NLP. An LSTM network is a special kind of RNN which also consists of 3 layers, that is, an input layer, a single recurrent hidden layer, and an output layer [31]. The main innovation of LSTM is that its hidden layer consists of one or more memory blocks. Each block includes one or more memory cells. In the standard form, the inputs are connected to all of the cells and gates, whereas the cells are connected to the outputs. The gates are connected to other gates and cells in the hidden layer. The single standard LSTM is a hidden layer with input, memory cell, and output gates [32, 33].

3.8. Dataset. To validate the proposed approach, we utilized DrugBank and MedLine open dataset, which have also been used by previous researchers. Additionally, we used drug label documents from various drug producers and regulator Internet sites located in Indonesia:

(1) http://www.kalbemed.com/

(2) http://www.dechacare.com/

(3) http://infoobatindonesia.com/obat/, and

(4) http://www.pom.go.id/webreg/index.php/home/produk/01.

The drug labels are written in Bahasa Indonesia, and their common contents are drug name, drug components, indication, contraindication, dosage, and warning.

3.9. Evaluation. To evaluate the performance of the proposed method, we use common measured parameters in data mining, that is, precision, recall, and F-score. The computation formula of these parameters is as follows. Let $C = \{C_1, C_2, C_3, \ldots, C_n\}$ be a set of the extracted drug name of this method, and $K = \{K_1, K_2, K_3, \ldots, K_l\}$ is set of actual drug names in the document set D. Adopted from [18], the parameter computations formula is

$$\begin{aligned} \text{Precision}\left(K_i, C_j\right) &= \frac{(\text{TruePositive})}{(\text{TruePositive} + \text{FalsePositive})} \\ &= \frac{\left(\left\|K_i \cap C_j\right\|\right)}{\left(\left\|C_j\right\|\right)}, \\ \text{Recall}\left(K_i, C_j\right) &= \frac{(\text{TruePositive})}{(\text{TruePositive} + \text{FalseNegative})} \\ &= \frac{\left(\left\|K_i \cap C_j\right\|\right)}{\left(\left\|K_i\right\|\right)}, \end{aligned} \tag{5}$$

where $\|K_i\|$, $\|C_j\|$, and $\|K_i \cap C_j\|$ denote the number of drug names in K, in C, and in both K and C, respectively. The F-score value is computed by the following formula:

$$\begin{aligned} &F\text{-score}\left(K_i, C_j\right) \\ &\quad = \frac{\left(2 * \text{Precision}\left(K_i, C_j\right) * \text{Recall}\left(K_i, C_j\right)\right)}{(\text{TruePositive} + \text{FalsePositive})}. \end{aligned} \tag{6}$$

4. Results and Discussion

4.1. MLP Learning Performance. The following experiments are the part of the first experiment. These experiments are performed to evaluate the contribution of the three regularization settings as described in Section 3.7.1. By arranging the sentence in training dataset as 5-gram of words, the quantity of generated sample is presented in Table 10. We do training and testing of the MLP-NN learning model for all those test data compositions. The result of model performances on both datasets, that is, MedLine and DrugBank, in learning phase is shown in Figures 6 and 7. The NN learning parameters that are used for all experiments are 500 input nodes, two hidden layers where each layer has 100 nodes with sigmoid activation, and 6 output nodes with softmax function; the learning rate = 1, momentum = 0.5, and epochs = 100. We used minibatch scenario in the training with the batch size being 100. The presented errors in Figures 6 and 7 are the errors for full batch, that is, the mean errors of all minibatches.

Table 10: Dataset composition.

Dataset	Train	Test		
		All	2/3 part	Cluster
MedLine	26,500	10,360	6,673	5,783
DrugBank	100,100	2,000	1,326	1,933

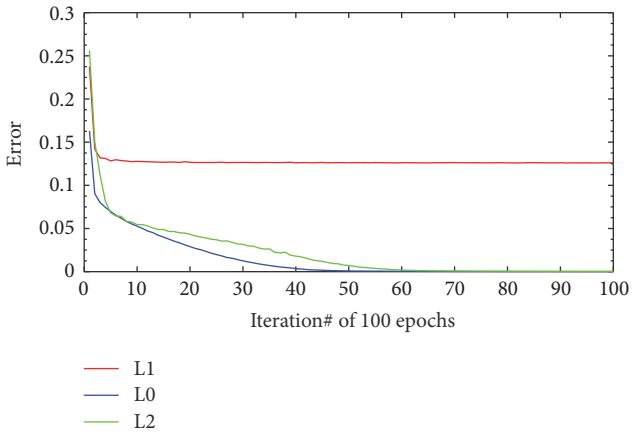


Figure 6: Full batch training error of MedLine dataset.

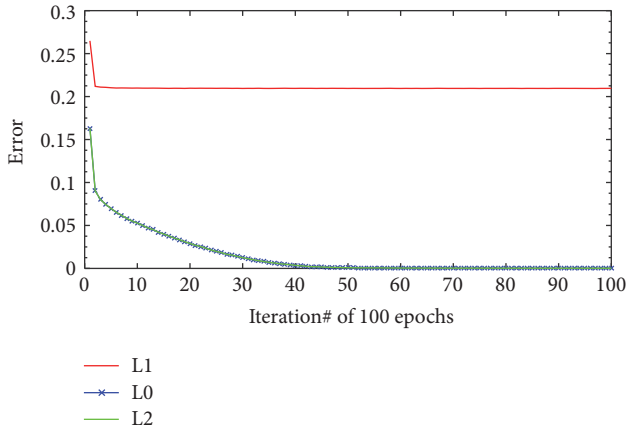


Figure 7: Full batch training error of DrugBank dataset.

The learning model performance shows different patterns between MedLine and DrugBank datasets. For both datasets, L1 regularization tends to stabilize in the lower iteration and its training error performance is always less than L0 or L2. The L0 and L2 training error performance pattern, however, shows a slight different behavior between MedLine and DrugBank. For the MedLine dataset, L0 and L2 produce different results for some of the iterations. Nevertheless, the training error performance of L0 and L2 for DrugBank is almost the same in every iteration. Different pattern results are probably due to the variation in the quantity of training data. As illustrated in Table 10, the volume of DrugBank training data is almost four times the volume of the MedLine dataset. It can be concluded that, for larger dataset, the contribution of L2 regularization setting is not too significant in achieving better performance. For smaller dataset (MedLine), however, the performance is better even after only few iterations.

4.2. Open Dataset Performance. In Tables 11, 12, 13, and 14, numbering (1), numbering (2), and numbering (3) in the most left column indicate the candidate selection technique with

(i) (1): all data tests being selected;

(ii) (2): 2/3 part of data test being selected;

(iii) (3): 2/3 part of 3 clusters for MedLine or 3/4 part of 4 clusters for DrugBank.

4.2.1. MLP-NN Performance. In this first experiment, for two data representation techniques and three candidate selection scenarios, we have six experiment scenarios. The result of the experiment which applies the first data representation technique and three candidate selection scenarios is presented in Table 11. In computing the F-score, we only select the predicted target which is provided by the lowest error (the minimum one). For MedLine dataset, the best performance is shown by L2 regularization setting where the error is 0.041818, in third candidate selection scenario with F-score 0.439516, whereas the DrugBank is achieved together by L0 and L1 regularization setting, with an error test of 0.0802; in second candidate selection scenario, the F-score was 0.641745. Overall, it can be concluded that DrugBank experiments give the best F-score performance. The candidate selection scenarios also contributed to improving the performance, as we found that, for both of MedLine and DrugBank, the best achievement is provided by the second and third scenarios, respectively.

The next experimental scenario in the first experiment is performed to evaluate the impact of the data representation technique and the addition of Wiki source in word2vec training. The results are presented in Tables 12 and 13. According to the obtained results presented in Table 11, the L0 regularization gives the best F-score. Hence, accordingly we only used the L0 regularization for the next experimental scenario. Table 12 presents the impact of the data representation technique. Looking at the F-score, the second technique gives better results for both datasets, that is, the MedLine and DrugBank.

Table 13 shows the result of adding the Wiki source into word2vec training in providing the vector of word representation. These results confirm that the addition of training data will improve the performance. It might be due to the fact that most of the targeted tokens such as drug name are uncommon words, whereas the words that are used in Wiki's sentence are commonly used words. Hence, the addition of commonly used words will make the difference between drug token and the nondrug token (the commonly used token) become greater. For the MLP-NN experimental results, the 4th scenario, that is, the second data representation with 2/3 partition data selection in DrugBank dataset, provides the best performance with 0.684646757 in F-score.

Table 11: The F-score performances of three of scenarios experiments.

MedLine	Prec	Rec	F-score	Lx	Error test
(1)	0.3564	0.5450	0.4310	L0	0.0305
(2)	0.3806	0.5023	0.4331	L1, L2	0.0432
(3)	0.3773	0.5266	0.4395	L2	0.0418
DrugBank	Prec	Rec	F-score	Lx	Error test
(1)	0.6312	0.5372	0.5805	L0	0.07900
(2)	0.6438	0.6398	0.6417	L0, L2	0.0802
(3)	0.6305	0.5380	0.5806	L0	0.0776

Table 12: The F-score performance as an impact of data representation technique.

Dataset	(1) One seq. of all sentences			(2) One seq. of each sentence		
MedLine	Prec	Rec	F-score	Prec	Rec	F-score
(1)	0.3564	0.5450	0.4310	0.6515	0.6220	0.6364
(2)	0.3806	0.5023	0.4331	0.6119	0.7377	**0.6689**
(3)	0.3772	0.5266	**0.4395**	0.6143	0.656873	0.6348
DrugBank	Prec	Rec	F-score	Prec	Rec	F-score
(1)	0.6438	0.5337	0.5836	0.7143	0.4962	0.5856
(2)	0.6438	0.6398	**0.6418**	0.7182	0.5804	**0.6420**
(3)	0.6306	0.5380	0.5807	0.5974	0.5476	0.5714

4.2.2. DBN and SAE Performance. In the second experiment, which involves DBN and SAE learning model, we only use the experiment scenario that gives the best results in the first experiment. The best experiment scenario uses the second data representation technique with Wiki text as an additional source in the word2vec training step.

In the DBN experiment, we use two stacked RBMs with 500 nodes of visible unit and 100 nodes of the hidden layer for the first and also the second RBMs. The used learning parameters are as follows: momentum = 0 and alpha = 1. We used minibatch scenario in the training, with the batch size of 100. As for RBM constraints, the range of input data value is restricted to $[0 \cdots 1]$ as the original RBM, which is developed for binary data type, whereas the range of vector of word value is $[-1 \cdots 1]$. So we normalize the data value into $[0 \cdots 1]$ range before performing the RBM training. In the last layer of DBN, we use one layer of MLP with 100 hidden nodes and 6 output nodes with softmax output function as classifier.

The used SAE architecture is two stacked AEs with the following nodes configuration. The first AE has 500 units of visible unit, 100 hidden layers, and 500 output layers. The second AE has 100 nodes of visible unit, 100 nodes of hidden unit, and 100 nodes of output unit. The used learning parameters for first SAE and the second SAE, respectively, are as follows: activation function = sigmoid and tanh; learning rate = 1 and 2; momentum = 0.5 and 0.5; sparsity target = 0.05 and 0.05. The batch size of 100 is set for both of AEs. In the SAE experiment, we use the same discriminative layer as DBN, that is, one layer MLP with 100 hidden nodes and 6 output nodes with softmax activation function.

The experiments results are presented in Table 14. There is a difference in performances when using the MedLine and the DrugBank datasets when feeding them into MLP, DBN, and SAE models. The best results for the MedLine dataset are obtained when using the SAE. For the DrugBank, the MLP gives the best results. The DBN gives lower average performance for both datasets. The lower performance is probably due to the normalization on the word vector value to $[0 \cdots 1]$, whereas their original value range is in fact between $[-1 \cdots 1]$. The best performance for all experiments 1 and 2 is given by SAE, with the second scenario of candidate selection as described in Section 3.6. Its F-score is 0.686192469.

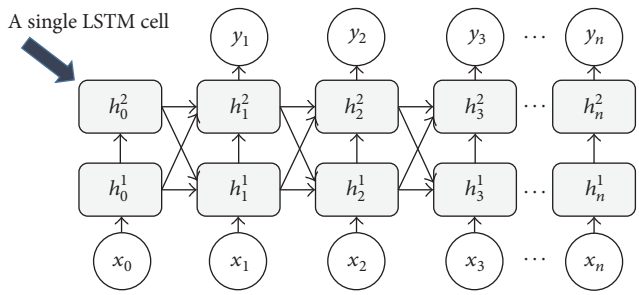


Figure 8: The global LSTM network.

4.2.3. LSTM Performance. The global LSTM network used is presented in Figure 8. Each single LSTM block consists of two stacked hidden layers and one input node with each input dimension being 200 as described in Section 3.4.3. All hidden layers are fully connected. We used sigmoid as an output activation function, which is the most suitable for binary classification. We implemented a peepholes connection LSTM variant where its gate layers look at the cell state [34]. In addition to implementing the peepholes connection,

Table 13: The *F*-score performances as an impact of the Wiki addition of word2vec training data.

Dataset	(1) One seq. of all sentences			(2) One seq. of each sentence		
MedLine	Prec	Rec	*F*-score	Prec	Rec	*F*-score
(1)	0.5661	0.4582	0.5065	0.614	0.6495	0.6336
(2)	0.5661	0.4946	**0.5279**	0.5972	0.7454	0.6631
(3)	0.5714	0.4462	0.5011	0.6193	0.6927	**0.6540**
DrugBank	Prec	Rec	*F*-score	Prec	Rec	*F*-score
(1)	0.6778	0.5460	0.6047	0.6973	0.6107	0.6511
(2)	0.6776	0.6124	**0.6433**	0.6961	0.6736	**0.6846**
(3)	0.7173	0.5574	0.6273	0.6976	0.6193	0.6561

Table 14: Experimental results of three NN models.

Dataset	MLP			DBN			SAE		
MedLine	Prec	Rec	*F*-score	Prec	Rec	*F*-score	Prec	Rec	*F*-score
(1)	0.6515	0.6220	0.6364	0.5464	0.6866	0.6085	0.6728	0.6214	0.6461
(2)	0.5972	0.7454	0.6631	0.6119	0.7377	0.6689	0.6504	0.7261	**0.6862**
(3)	0.6193	0.6927	0.6540	0.6139	0.6575	0.6350	0.6738	0.6518	0.6626
Average	0.6227	0.6867	0.6512	0.5907	0.6939	0.6375	0.6657	0.6665	0.6650
DrugBank	Prec	Rec	*F*-score	Prec	Rec	*F*-score	Prec	Rec	*F*-score
(1)	0.6973	0.6107	0.6512	0.6952	0.5847	0.6352	0.6081	0.6036	0.6059
(2)	0.6961	0.6736	**0.6847**	0.6937	0.6479	0.6700	0.6836	0.6768	0.6802
(3)	0.6976	0.6193	0.6561	0.6968	0.5929	0.6406	0.6033	0.6050	0.6042
Average	0.6970	0.6345	0.664	0.6952	0.6085	0.6486	0.6317	0.6285	0.6301

we also use a couple of forget and input gates. The detailed single LSTM architecture and each gate formula computation can be referred to in [33].

The LSTM experiments were implemented with several different parameter settings. Their results presented in this section are the best among all our experiments. Each piece of input data consists of two components, its word vector value and its Euclidian distance to the previous input data. In treating both input data components, we adapt the Adding Problem Experiment as presented in [35]. We use the Jannlab tools [36] with some modifications in the part of entry to conform with our data settings.

The best achieved performance is obtained with LSTM block architecture of one node input layer, two nodes' hidden layer, and one node output layer. The used parameters are learning rate = 0.001, momentum = 0.9, epoch = 30, and input dimension = 200, with the time sequence frame set to 2. The complete treatment of drug sentence as a sequence both in representation and in recognition, to extract the drug name entities, is the best technique, as shown by *F*-score performance in Table 15.

As described in previous work section, there are many approaches related to drug extraction that have been proposed. Most of them utilize certain external knowledge to achieve the extraction objective. Table 16 summarizes their *F*-score performance. Among the state-of-the-art techniques, our third data representation technique applied to the LSTM model is outperforming. Also, our proposed method does not require any external knowledge.

Table 15: The *F*-score performance of third data representation technique with RNN-LSTM.

	Prec	Rec	*F*-score
MedLine	1	0.6474	0.7859
DrugBank	1	0.8921	0.9430
Average			0.8645

4.3. Drug Label Dataset Performance. As additional experiment, we also use Indonesian language drug label corpus to evaluate the method's performance. Regarding the Indonesian drug label, we could not find any certain external knowledge that can be used to assist the extraction of the drug name contained in the drug label. In the presence of this hindrance, we found our proposed method is more suitable than any other previous approaches. As the drug label texts are collected from various sites of drug distributors, producers, and government regulators, they do does not clearly contain training data and testing data as in DrugBanks or MedLine datasets. The other characteristics of these texts are the more structured sentences contained in the data. Although the texts are coming from various sources, all of them are similar kind of document (the drug label that might be generated by machine). After the data cleaning step (HTML tag removal, etc.), we annotated the dataset manually. The total quantity of dataset after performing the data representation step, as described in Section 3.4, is

Table 16: The *F*-score performance compared to the state of the art.

Approach	*F*-score	Remark
The Best of SemEval 2013 [15]	0.7150	—
[11]	0.5700	With external knowledge, ChEBI
[16] + Wiki	0.7200	With external knowledge, DINTO
[14]	0.7200	Additional feature, BIO
[12]	0.6000	Single token only
MLP-SentenceSequence + Wiki (average)/Ours	0.6580	Without external knowledge
DBN-SentenceSequence + Wiki (average)/Ours	0.6430	Without external knowledge
SAE-SentenceSequence + Wiki (average)/Ours	0.6480	Without external knowledge
LSTM-AllSentenceSequence + Wiki + EuclidianDistance (average)/Ours	0.8645	Without external knowledge

Table 17: The best performance of 10 executions on drug label corpus.

Iteration	Prec	Recall	*F*-score
1	0.9170	0.9667	0.9412
2	0.8849	0.9157	0.9000
3	0.9134	0.9619	0.9370
4	0.9298	0.9500	0.9398
5	0.9640	0.9570	0.9605
6	0.8857	0.9514	0.9178
7	0.9489	0.9689	0.9588
8	0.9622	0.9654	0.9638
9	0.9507	0.9601	0.9554
10	0.9516	0.9625	0.9570
Average	0.93081	0.9560	0.9431
Min	0.8849	0.9157	0.9000
Max	0.9640	0.9689	0.9638

1.046.200. In this experiment, we perform 10 times cross-validation scenario by randomly selecting 80% data for the training data and 20% data for testing.

The experimental result for drug label dataset shows that all of the candidate selection scenarios provide excellent *F*-score (above 0.9). The excellent *F*-score performance is probably due to the more structured sentences in those texts. The best results of those ten experiments are presented in Table 17.

4.4. Choosing the Best Scenario. In the first and second experiments, we studied various experiment scenarios, which involve three investigated parameters: additional Wiki source, data representation techniques, and drug target candidate selection. In general, the Wiki addition contributes to improving the *F*-score performance. The additional source in word2vec training enhances the quality of the resulting word2vec. Through the addition of common words, from Wiki, the difference between the common words and the uncommon words, that is, drug name, becomes greater (better distinguishing power).

One problem in mining drug name entity from medical text is the imbalanced quantity between drug token and other tokens [15]. Also, the targeted drug entities are only a small part of the total tokens. Thus, majority of tokens are noise. In dealing with this problem, the second and third candidate selection scenarios show their contribution to reduce the quantity of noise. Since the possibility of extracting the noises is reduced then the recall value and *F*-score value increase as well, as shown in the first and second experiments results.

The third experiment which uses LSTM model does not apply the candidate selection scenario because the input dataset is treated as sentence sequence. So the input dataset can not be randomly divided (selected) as the tuple treatment in the first and second experiments.

5. Conclusion and Future Works

This study proposes a new approach in the data representation and classification to extract drug name entities contained in the sentences of medical text documents. The suggested approach solves the problem of multiple tokens for a single entity that remained unsolved in previous studies. This study also introduces some techniques to tackle the absence of specific external knowledge. Naturally, the words contained in the sentence follow a certain sequence pattern; that is, the current word is conditioned by other previous words. Based on the sequence notion, the treatment of medical text sentences which apply the sequence NN model gives better results. In this study, we presented three data representation techniques. The first and second techniques treat the sentence as a nonsequence pattern which is evaluated with the non-sequential NN classifier (MLP, DBN, and SAE), whereas the third technique treats the sentences as a sequence to provide data that is used as the input of the sequential NN classifier, that is, LSTM. The performance of the application of LSTM models for the sequence data representation, with the average *F*-score being 0.8645, rendered the best result compared to the state of the art.

Some opportunities to improve the performance of the proposed technique are still widely opened. The first step improvement can be the incorporation of additional hand-crafted features, such as the words position, the use of capital case at the beginning of the word, and the type of character, as also used in the previous studies [16, 37]. As presented in the MLP experiments for drug label document, the proposed methods achieved excellent performance when applied to the more structured text. Thus, the effort to make the sentence of the dataset, that is, DrugBank and MedLine, to be more structured can also be elaborated. Regarding the LSTM model and the sequence data representation for the sentences of medical text, our future study will tackle the multiple entity extractions such as drug group, drug brand, and drug compounds. Another task that is potential to be solved with

the LSTM model is the drug-drug interaction extraction. Our experiments also utilize the Euclidean distance measure in addition to the word2vec features. Such addition gives a good *F*-score performance. The significance of embedding the Euclidean distance features, however, needs to be explored further.

Competing Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by Higher Education Science and Technology Development Grant funded by Indonesia Ministry of Research and Higher Education Contract no. 1004/UN2.R12/HKP.05.00/2016.

References

[1] M. Sadikin and I. Wasito, "Translation and classification algorithm of FDA-Drugs to DOEN2011 class therapy to estimate drug-drug interaction," in *Proceedings of the The 2nd International Conference on Information Systems for Business Competitiveness (ICISBC '13)*, pp. 1–5, Semarang, Indonesia, 2013.

[2] H. Tang and J. Ye, "A survey for information extraction method," Tech. Rep., 2007.

[3] S. Zhang and N. Elhadad, "Unsupervised biomedical named entity recognition: experiments with clinical and biological texts," *Journal of Biomedical Informatics*, vol. 46, no. 6, pp. 1088–1098, 2013.

[4] I. Korkontzelos, D. Piliouras, A. W. Dowsey, and S. Ananiadou, "Boosting drug named entity recognition using an aggregate classifier," *Artificial Intelligence in Medicine*, vol. 65, no. 2, pp. 145–153, 2015.

[5] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, and T. Declerck, "The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions," *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 914–920, 2013.

[6] H. Sampathkumar, X.-W. Chen, and B. Luo, "Mining adverse drug reactions from online healthcare forums using Hidden Markov Model," *BMC Medical Informatics and Decision Making*, vol. 14, article 91, 2014.

[7] I. Segura-Bedmar, P. Martínez, and M. Segura-Bedmar, "Drug name recognition and classification in biomedical texts. A case study outlining approaches underpinning automated systems," *Drug Discovery Today*, vol. 13, no. 17-18, pp. 816–823, 2008.

[8] S. Keretna, C. P. Lim, D. Creighton, and K. B. Shaban, "Enhancing medical named entity recognition with an extended segment representation technique," *Computer Methods and Programs in Bimoedicine*, vol. 119, no. 2, pp. 88–100, 2015.

[9] G. Pal and S. Gosal, "A survey of biological entity recognition approaches," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 9, 2015.

[10] S. Liu, B. Tang, Q. Chen, and X. Wang, "Drug name recognition: approaches and resources," *Information*, vol. 6, no. 4, pp. 790–810, 2015.

[11] T. Grego and F. M. Couto, "LASIGE: using conditional random fields and ChEBI ontology," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval '13)*, vol. 2, pp. 660–666, 2013.

[12] J. Bjorne, S. Kaewphan, and T. Salakoski, "UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge," in *Proceedings of the 2nd Joint Conferernce on Lexical and Computational Semantic*, vol. 2, pp. 651–659, Atlanta, Ga, USA, 2013.

[13] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," https://arxiv.org/abs/1301.3781.

[14] A. Ben Abacha, M. F. M. Chowdhury, A. Karanasiou, Y. Mrabet, A. Lavelli, and P. Zweigenbaum, "Text mining for pharmacovigilance: using machine learning for drug name recognition and drug-drug interaction extraction and classification," *Journal of Biomedical Informatics*, vol. 58, pp. 122–132, 2015.

[15] I. Segura-Bedmar, P. Martinez, and M. Herrero-Zazo, "Semeval-2013 task 9: extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval '13)*, vol. 2, pp. 341–350, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013.

[16] I. Segura-bedmar and P. Mart, "Exploring word embedding for drug name recognition," in *Proceedings of the The 6th International Workshop on Health Text Mining and Information Analysis*, pp. 64–72, Lisbon, Portugal, September 2015.

[17] Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny, and H. Xu, "A study of active learning methods for named entity recognition in clinical text," *Journal of Biomedical Informatics*, vol. 58, pp. 11–18, 2015.

[18] M. Sadikin and I. Wasito, "Toward object interaction mining by starting with object extraction based on pattern learning method," in *Proceedings of the Pattern Learning Method Asia-Pacific Materials Science and Information Technology Conference (APMSIT '14)*, Shanghai, China, December 2014.

[19] Q.-C. Bui, P. M. A. Sloot, E. M. Van Mulligen, and J. A. Kors, "A novel feature-based approach to extract drug-drug interactions from biomedical text," *Bioinformatics*, vol. 30, no. 23, pp. 3365–3371, 2014.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," 2012.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[23] A. Fischer and C. Igel, "Progress in pattern recognition, image analysis, computer vision, and applications," in *17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3–6, 2012. Proceedings*, An Introduction to Restricted Boltzmann Machines, pp. 14–36, Springer, Berlin, Germany, 2012.

[24] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pp. 1064–1071, 2008.

[25] G. E. Dahl, R. P. Adams, and H. Larochelle, "Training restricted boltzmann machines on word observations," https://arxiv.org/abs/1202.5695.

[26] Y. Tang and I. Sutskever, *Data Normalization in the Learning of Restricted Boltzmann Machines*, Department of Computer

Science, Toronto University UTML-TR-11, Toronto, Canada, 2011, http://www.cs.toronto.edu/~tang/papers/RbmZM.pdf.

[27] G. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., vol. 7700 of *Lecture Notes in Computer Science*, pp. 599–619, Springer, Berlin, Germany, 2nd edition, 2012.

[28] H. Chen and A. F. Murray, "Continuous restricted Boltzmann machine with an implementable training algorithm," *IEE Proceedings—Vision, Image and Signal Processing*, vol. 150, no. 3, pp. 153–158, 2003.

[29] M. Welling, M. Rosen-Zvi, and G. E. Hinton, "Exponential family harmoniums with an application to information retrieval," in *Advances in Neural Information Processing Systems (NIPS) 17*, pp. 1481–1488, 2005.

[30] A. Ng, Deep Learning Tutorial, http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] J. Hammerton, "Named entity recognition with long short-term memory," in *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL (CONLL '03)*, vol. 4, pp. 172–175, 2003.

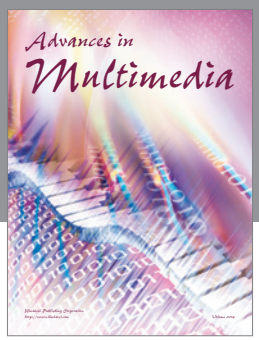
[33] C. Olah, "Understanding LSTM Networks," 2015, http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

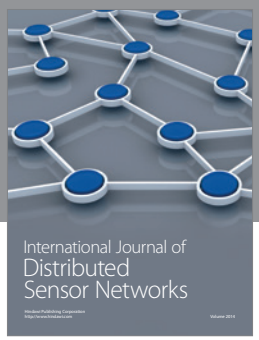
[34] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN '00)*, vol. 3, p. 6, Como, Italy, July 2000.

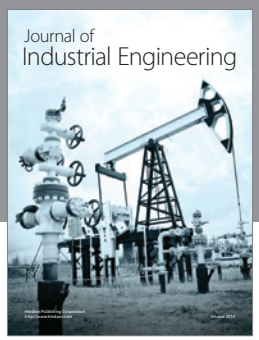
[35] S. Hochreiter, "LSTM Can Solve Hard," in *Advances in Neural Information Processing Systems*, Neural Information Processing Systems (NIPS), Denver, Colo, USA, 1996.

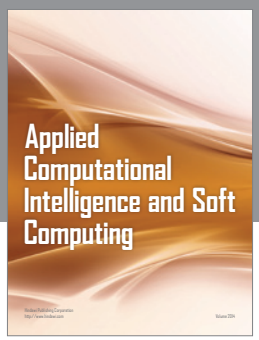
[36] S. Otte, D. Krechel, and M. Liwicki, "Jannlab neural network framework for java," in *Proceedings of the Poster Proceedings Conference (MLDM '13)*, pp. 39–46, IBAI, New York, NY, USA, 2013.

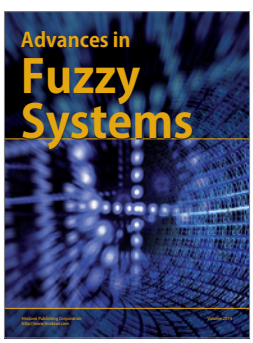
[37] R. Boyce and G. Gardner, "Using natural language processing to identify pharmacokinetic drug-drug interactions described in drug package inserts," in *Proceedings of the Workshop on Biomedical Natural Language Processing (BioNLP '12)*, pp. 206–213, BioNLP, Montreal, Canada, 2012.

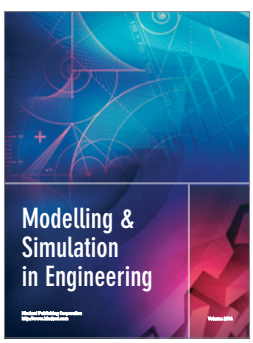


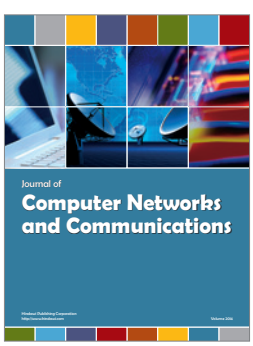












Hindawi
Submit your manuscripts at
http://www.hindawi.com