Exploring the Adaptation of Recurrent Neural Network Approaches for Extracting Drug–Drug Interactions from Biomedical Text

Wen-Juan Hou and Bamfa Ceesay

***Abstract*—Information extraction (IE) is the process of automatically identifying structured information from unstructured or partially structured text. IE processes can involve several activities, such as named entity recognition, event extraction, relationship discovery, and document classification, with the overall goal of translating text into a more structured form. Information on the changes in the effect of a drug, when taken in combination with a second drug, is known as drug–drug interaction (DDI). DDIs can delay, decrease, or enhance absorption of drugs and thus decrease or increase their efficacy or cause adverse effects. Recent research trends have shown several adaptation of recurrent neural networks (RNNs) from text. In this study, we highlight significant challenges of using RNNs in biomedical text processing and propose automatic extraction of DDIs aiming at overcoming some challenges. Our results show that the system is competitive against other systems for the task of extracting DDIs.**

***Index Terms*—Drug–drug interaction, deep learning, embedding, machine learning.**

I. Introduction

A pharmacological effect that occurs when a given drug is altered by the action of another drug, leading to unwanted clinical effects, is referred to as the drug–drug interaction (DDI) [1]. Identifying DDIs is a major challenge in drug development. Previous attempts have established formal approaches for pharmacokinetic DDIs [2], but there are no feasible solutions for pharmacodynamics DDIs because the endpoint is often a serious adverse event rather than a measurable change in drug concentration. When drugs are co-administered, one drug may increase or decrease the effect of the other or lead to an unexpected effect.

In recent years, the 2011 [3] and 2013 [4] DDI Extraction challenges have been held to promote the implementation and comparative assessment of natural language processing techniques in the field of pharmacovigilance. In the 2013 challenge, the DDIs needed to be classified into four predefined DDI types: advice, effect, mechanism, and int [5]. "Advice" is assigned when a recommendation or advice regarding the concomitant use of two drugs is described. For example, consider the following sentence: "Concurrent therapy with **ORENCIA** and **TNF antagonist** is not recommended." This sentence is classified as advice. "Effect" is assigned when the effect of the DDI is described. The effect can be one of the following: a pharmacological effect, a clinical finding, signs or symptoms, unspecific modification of the effect or action of the drugs, an increase in toxicity or a protective effect, and therapeutic failure. For example, the sentence "This may indicate that **ibuprofen** could enhance the toxicity of **methotrexate**" is classified as "effect." "Mechanism" is assigned when a pharmacokinetic mechanism, which is a mechanism by which a drug has been absorbed, distributed, metabolized, and excreted, is affected. For example, the following sentence is a mechanism type: *"Concomitant use of **calcium** supplements and **L-lysine** may increase calcium absorption."* "Int" is assigned when the sentence states that an interaction occurs and does not provide any information about the interaction. For example, the sentence "*A possible drug interaction of **FOSCAVIR** and intravenous **pentamidine** has been described*." is a type of "Int."

Exploring DDIs has received much attention in both industry and academia. A variety of methods have been published to extract DDI information.

First, some computational approaches such as network-based algorithms were proposed to predict DDIs [6]-[9]. Furthermore, machine learning approaches can automatically build classifiers for relation extraction. Generally, such approaches use the contextual features derived from natural language processing techniques such as shallow parsing or full dependency parsing. Some studies have applied machine learning approaches to deal with DDI extraction problems [8], [10]-[12].

In recent years, the machine learning community has made great advances in deep learning, and deep learning-based methods have been employed for related tasks. Researches using Long Short-Term Memory (LSTM) [13], convolutional neural networks (CNNs) [14], and recurrent neural networks (RNNs) [15] were explored [16]-[19]. In addition, several similarity-based mining techniques were applied to solve the DDI extraction problem [20]-[23]. Another way to extract DDIs is to employ the syntactic information from biomedical literature using text-mining techniques. Zheng *et al*. [24] presented a graph kernel, which made full use of different types of context to identify DDIs from biomedical literature.

A. Challenges with Recurrent Neural Network

Recurrent neural networks (RNNs) and LSTM are widely adapted for their success in solving sequence learning

Manuscript received January 20, 2020; revised November 24, 2020. This work was supported in part by the National Taiwan Normal University.

The authors are with the Department of Computer Science and Information Engineering, National Taiwan Normal University. No. 88, Tingzhou Road, Sec. 4, Taipei 116, Taiwan (e-mail: emilyhou@csie.ntnu.edu.tw, bmfceesay@csie.ntnu.edu.tw).

doi: 10.18178/ijmlc.2021.11.4.1046

problems. RNNs and their derivative models are often designed to sequentially process data for a certain period of time. Given a particular node in an RNN, the model processes any given input at the previous nodes, one after another in a sequential fashion. A basic RNN model with a recurrent layer *f* and a feedforward layer *g* as shown in Fig. 1 below, allows information transferring in recurrent layers from one node to another.

This generic model is usually trained by unfolding the layers and passing information in a feedforward manner. This can lead to two observable problems during training:

1) A large network from unrolled RNN.
2) Duplicates of instances contributed to the gradient and current layer gradient are the product of previous layers.

The usually approach to training RNN models is through backpropagation through time (BPTT) [25]. For *N* number of instances, the objective of BPTT is to unroll a given recurrent network into a feedforward network with *N* instances of the original network, one instance at a given time stamp. Nielsen *et al.* [26] shows that the gradient tends toward exponentially large or small as the number of layers increases. This means that for *N* number of layers, the gradient can be computed as:

$$\prod_{i=1}^{N} w_i \theta_i \,, \tag{1}$$

where w_i and θ_i are the weight and activation function at layer *i*, respectively. When a recurrent network is unrolled, the weights are the duplicates of the original network. Therefore the gradient can also be expressed as:

$$w \prod_{i=1}^{N} \theta_i \,. \tag{2}$$

The resurfacing of the same parameter for computing the gradient, makes it very unstable for a large value of *N*. The problem of exploding gradients occurs when, for example the gradient values exponentially grow more than the vector norms under consideration. From (2), it can be noted that we are multiplying with the same weight multiple times. However, multiplication with very small values will quickly decrease the resulting gradient. This leads to a vanishing gradient problem [27].

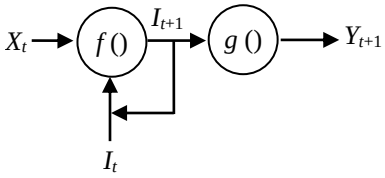

Fig. 1. Basic RNN model.

B. Unstable Gradient Problem

There are several attempts to solving the above mentioned problems. For example, in the case of the exploding gradient problem, the problem can be limited by:

1) Terminating backpropagation earlier to avoid lower values of gradients. This is not optimal since it does not consider all weights.
2) Manually reducing or penalizing the gradient.
3) Assigning a maximum threshold limit to the gradient.

For the vanishing gradient, the following approach are usually adapted to solving or limiting the problem:

1) Manually initializing the weight so as to avoid potential for the vanishing gradient.
2) Using the echo state networks (ESNs) to solve the vanishing gradient.
3) Using long-short term memory networks (LSTMs).

The vanishing gradient problem is a frequent problem that most RNN models encounter during the training phase. Recent studies have shown the emergence of several RNN derivatives aims at dealing with this problem. Such significant contributions include the LSTM and GRU models [8], attention based networks such as transformer [28] and transformerXL [29].

The LSTM derivative introduces a gating concept to bypass units and remember information for a longer time stamp. However, these models still have a sequential path that has potentials to introduce problems to gradients. Attention mechanisms have seen significant success in solving the unstable gradient problem. Elbayed *et al.* [30] used a single 2D convolutional neural network across both sequences. Each layer of this network re-codes source tokens on the basis of the output sequence produced. The attention-like property is therefore pervasive throughout such network. It is said to outperform both RNN/LSTM and Attention based models like the Transformer.

The problem of extracting DDIs has attracted the attention of many researchers. This paper explores the challenges using RNN in determining and classifying DDIs from biomedical literature and proposes methods to overcome the shortcomings with RNN models and their derivatives. We proposed a sentence level attention mechanism to determine the relevancy of a given sentence to a DDI and used a sequence learning model adopted in [29] to model the likelihood of two drug entities participating in a drug–drug interaction. We analyze this approach with the objectives of overcoming weakness in our previous study [31]. In that study a neural embedding approach based on the LSTM is used to solving the DDI extraction task. In contrast to that study, the proposed approach in this paper aims at exploring the challenges in deploying RNNs to the DDI extraction task.

II. Experimental Data

DDIs have not only gained popularity among researchers but have also become themes for major information challenges such as the SemEval. SemEval 2013 provided a DDI Extraction challenge with a benchmark corpus. This study uses the 2013 DDI Extraction corpus, which provides both annotated training and test ground truth data. Fig. 2 presents sample annotated data from the 2013 DDI Extraction task.

As shown in Fig. 2, the annotation gives the following tags:

1) The document tag. This provides the *id* attribute, which gives the source document id in the MedLine or DrugBank corpuses.
2) The sentence tag. This provides two attributes: (1) an *id* attribute, which gives the identification tag of the

sentence from the source document; and (2) a text attribute, which gives the text of the sentence.

3) The entity tag. This tag provides the entity information in the sentence with attributes including identification tag (*id*), character spans in the text (*charOffset*), the type of entity (*type*), and name of the entity (text).
4) The pair tag. This tag annotates the entity pairs participating in a DDI within the sentence's text. The attributes include a pair identification attribute (*id*), two entity attributes (*e*1 and *e*2) that annotate the entities, a *ddi* attribute that has a true or false value indicating whether the entity pair (*e*1, *e*2, *pair*) is involved in a DDI, and a *type* attribute that annotates the DDI type.

```
<?xml version="1.0" encoding="UTF-8"?>
<document id="DDI-MedLine.d134">
<sentence id="DDI-MedLine.d134.s2"
text="An intravenous injection of
perchlorate given later also produces
a complete and immediately beginning
depletion of pertechnetate already
accumulated in the thyroid, within a
period of 195 min after 99m-TcO-4-injection
 with a corresponding increase in blood
levels.">
<entity id="DDI-MedLine.d134.s2.e0"
charOffset="28-38" type="drug_n"
text="perchlorate"/>
<entity id="DDI-MedLine.d134.s2.e1"
charOffset="116-128" type="drug"
text="pertechnetate"/>
<pair id="DDI-MedLine.d134.s2.p0"
 e1="DDI-MedLine.d134.s2.e0"
 e2="DDI-MedLine.d134.s2.e1"ddi="true"
 type="mechanism"/>
</sentence>
</document>
</xml>
```

Fig. 2. An example of an annotated document of the DDI corpus.

There are 5,021 DDIs annotated from 730 and 175 DrugBank [32] texts and MedLine abstracts, respectively. For the classification task, DDIs are classified into the following four predefined types, as mentioned in Section II: advice, *effect*, *mechanism* and *int*.

The corpus is split into building data sets for training and testing. The training dataset comprises randomly selected DrugBank texts and MedLine abstracts. From the annotated dataset, 572 DrugBank and 142 MedLine abstracts were used for training. The remaining 158 DrugBank texts and 33 MedLine abstracts were used as test datasets. Segura-Bedmar *et al.* [3] provided detailed descriptions of the method used to collect and process documents from DrugBank and MedLine.

III. Method

The DDI extraction task has two components: (1) DDI identification and (2) DDI classification into predefined types.

A. Data Preprocessing

Sentences that have no mention of drug entities or have only a single drug entity mentioned cannot be considered for DDI extraction. Such a sentence is irrelevant because in DDI extraction, two different drug entities are required for DDI to happen.

We consider data abstraction as important to keep the identity of drug entities in a given sentence. In the data abstraction process, the objective is to normalize the text in the sentence by removing numerical characters and changing upper case letters to lower case. Pair entities participating in a DDI are also replaced with special characters. The characters "#" and "d" are used to represent the participating drug entities associated to a given DDI. For example, consider the sentence:

- ***Barbiturates*** *and* ***glutethimide*** *should not be administered to patients receiving* ***coumarin drugs.***

Using the data abstraction process mentioned above, this sentence can be presented as:

- *# and d should not be administered to patients receiving #.*

It can be noted that the drug entities **Barbiturate** and **coumarin** are the target drugs and are abstracted using "#", whereas the agent of DDI, **glutethimide** is abstracted using "d". The objectives of data abstraction is to preserve the reusability and identity of entities that appear in a DDI as targets or agents.

B. Embedding

The object in embedding in-text mining is to map lexical items or variables such as sentences or words to a corresponding numerical vector that can be used in learning algorithms. In these cases, embedding results are considered as a low-dimensional continuous vector representation of variables. Neural embedding has two important major benefits:

1) Reducing the dimensions of variables.
2) Meaningfully representing data in a transformed space.

For in-text mining and processing, neural embedding has three notable applications:

1) Determining the nearest neighbor relationships among lexical variables. This can be used in recommendation systems based on specific user interests or cluster categories.
2) Neural embedding can be used for generating input space for learning supervised models.
3) Visualization of concepts and relations among lexical variables.

Neural network embeddings have three primary purposes:

1) To find the nearest neighbors in the embedding space. These can be used to make recommendations based on user interests or cluster categories.
2) To be used as inputs to a machine learning model for a supervised task.
3) To visualize concepts and relations between categories.

The fundamental baseline approach to neural embedding is one-hot encoding. This approach maps variables to a vector of 0s and a single 1 representing a given variable. For illustration, consider the sample result from our data preprocessing.

- *# and d should not be administ to patient receiv #*

For one-hot encoding, we can represent this expression as follows:

$$V = \left[\#, and, d, should, not, be, ad\ \min ist, to, patient, receiv, \#\right] \quad (3)$$

and

$$V_{encode} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

From (4), it can be observed that the one-hot encoding technique has two main limitations:

1) The number of variables is directly proportional to the dimensionality of the transformed vectors. This implies that the resulting vectors can be very large.
2) The approach does not distinguish between similar categories and relationships between variables in the embedding space.

The first limitation can be explained as follows: given a set of variables, each variable in the set requires adding another one-hot encoded vector to represent each variable. For the purpose of illustration, the given 37,000 lexical variables will require 37,000-dimensional vectors. It thus becomes obvious that training any learning model on such a representation is infeasible. The second limitation considers the similarity relationship between variables. One-hot encoding produces a cosine similarity score of 0 for each comparison between encoded representations of variables. This limitation thus ignores the significance of relationships that might exist among variables.

Given a sentence containing two drug entities w_1 ="#" and w_2 ="d", each word is represented in a two dimensional space, a word embedding space and a positional embedding space. In general, a sentence is presented as:

$$S = \{w_1, w_2, w_3, \cdots\cdots, w_N\}, \quad (5)$$

where w_i is the representation of word at index i in the sentence. A function $f()$ is used to map words or relative position of words to a column vector. Let h_i and k_i represent the one-hot encoding and relative positional encoding of word w_i at position p_i. For a word embedding ∂_w and position embedding ∂_k, the mapping is implemented as:

$$f(w_i) = \partial_w h_i \quad (6)$$

and

$$f(p_i) = \partial_w k_i. \quad (7)$$

These information are then processed at a sentence level, and each sentence is presented by a sequence of words.

C. Sentence Based Attention

The objective of the sentence based attention mechanism in this study is to determine how sentences correlate to each other. Since the occurrence of a DDI is only within a sentence, we used the self-attention mechanism to understand the correlation of words in a sentence. This can be achieved by determining relevant instances of the DDI vector representation correlated to the rest of the sentence. For a DDI vector representation D and the attention matrix A, The correlation C_i between a sentence S_i and D can be estimated as:

$$C_i = S_i^T AD. \quad (8)$$

The softmax θ, is defined as:

$$\theta_i = \frac{\exp(C_i)}{\sum_{j=1}^{N} \exp(C_j)}. \quad (9)$$

The final representation includes all the relevant information and expressed as:

$$S = \sum_{j=1}^{N} \theta_j S_j. \quad (10)$$

The relevant information space is directly propotional to the number of sentences containing a given DDI. In this study we adapt the self-attention model proposed in [29]. The attention used in this study is shown as in Fig. 3.

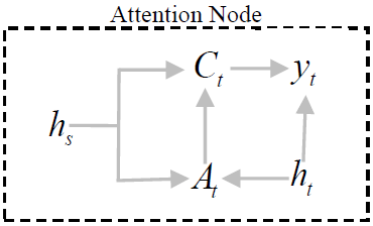

Fig. 3. Hidden node with global attention.

From Fig. 3, the input hidden states, h_s, are stacked together and used to compute the context vector C_t, which presents the context of individual sentences with respect to weight alignments and the hidden state. The global alignment vector depends on the input hidden state h_s and the output hidden state h_t and is calculated as follows:

$$A_t(s) = \frac{\exp(score(h_t, h_s))}{\sum \exp(score(h_t, h_s))}, \quad (11)$$

where score(h_t, h_s) is the dot product presenting the similarity between the hidden input and output states. The final target sentence for an attention is thus computed from the context vector and the hidden state h_t.

$$y_t = \tanh(w[C_t, h_t]). \quad (12)$$

A predictive distribution is obtained from this attention result using softmax as follows:

$$softmax(w, h_t) = p(y_t \mid y_{t-1}, s). \quad (13)$$

D. The Recurrent Unit

The recurrent unit in this study is an LSTM model. The general LSTM module used in this paper is shown in Fig. 4. This module is a representation of the repeating module. A single module has four neural layer units: three sigmoid units and a *tanh* layer unit. The first sigmoid unit in the figure serves as a filter that determines what information is allowed to go through the cell stages. This unit considers input I_t at time *t* and previous hidden units at H_{t-1} and outputs a score between 0 and 1, representing the significance of input in the next cell stage. The score value 1 represents highly significant and value 0 represents the least significant input. The second sigmoid unit and the first *tanh* unit serve as the updating units. The sigmoid units pass information to be updated and the *tanh* unit creates an updated value. Finally, this updated value is added to the network cell to replace the old cell information. In Fig. 4, the symbol $\otimes$ is a point-wise element operation and $\oplus$ is an element-by-element addition [31]. In the DDI recognition model, the LSTM is a module to perform binary prediction to determine DDI entities. In contrast, the DDI classification model is a module to perform the multi-classification task to predict the DDI entities predefined types.

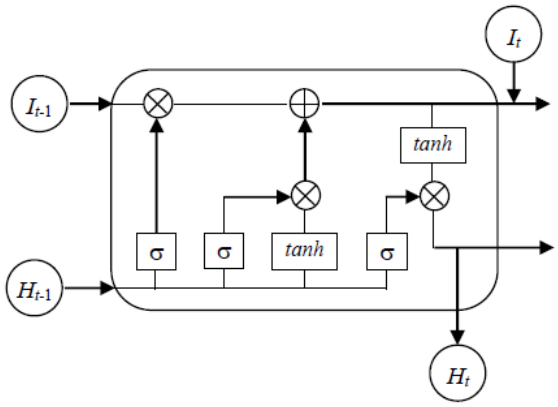

Fig. 4.General LSTM module used in DDI recognition and classification.

E. Classification and Prediction

One-hot encoding does not rely on supervision to create representations of variables. If embedding can be learned as a supervised task, it could help improve the embedding results. Such embedding results could not form parameter vectors (weight vectors) that could be used as input parameters for learning models but could be adjusted to reduce the loss on a learning task. This yields embedding representations that consider placing similar or related variables relatively closer to each other. In this paper, learning is considered an optimization problem of a loss function with regards to the embedding. The embedding results are adjusted during training to minimize the loss on the supervised task. The word embedding and LSTM learning adapted in this study are similar to those used in [31]. Here, we also use cross entropy, a commonly applied distance/loss measure. This choice is indicated in the discrete case by derivation from the formulation of the loss function:

$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log \hat{y}_j , \quad (14)$$

where $|V|$ is the size of the vocabulary and output $\hat{y}$ is an estimate of *y*. This study adopts the same application of entropy as a loss function in training neural networks, with embedding proposed by Hou and Ceesay [31].

IV. Experimental Results

The evaluation metric is relation-oriented and based on the standard precision, recall, and F-score metrics. Note that only relations are evaluated since entities will be included in the test dataset. In our task, we evaluate the results of the system considering the following evaluation criteria that are used in SemEval 2013:

1) Macro evaluation: a DDI is correctly detected only if the system is able to assign the correct prediction and the correct type to it. In other words, a pair is correct only if both prediction and type are correct. When the prediction is 0, the type may be empty or null.
2) Micro evaluation: a pair is correct when its prediction type matches the gold annotation.

Evaluation results are based on the standard precision, recall and F-score metrics: Precision is the percentage of DDIs found by the learning system that are correct. That is, precision is the ratio between the number of DDIs correctly detected (true positives, *TP*) and the total number of DDIs that were found by the system (true positives + false positives, *TP*+*FP*):

$$P = \frac{TP}{TP + FP} . \quad (15)$$

Recall is the percentage of DDIs presented in the corpus that are found by the system. In other words, recall is the ratio between the number of DDIs correctly detected (true positives) and the total number of drug entities in the gold standard (true positives + false negatives, *TP*+*FN*):

$$R = \frac{TP}{TP + FN} . \quad (16)$$

F-score is the harmonic mean of precision and recall:

$$F = \frac{2 \times P \times R}{P + R} . \quad (17)$$

We considered macro-average measures of precision, recall, and f-measure for DDIs in this study. While the micro-averaged F-score is calculated by constructing a global contingency table and then calculating precision and recall, the macro-averaged F-score is calculated by first calculating precision and recall for each type and then taking the average of these. Thus, the precision for mechanism relationships can be defined as the ratio between the number of DDIs correctly classified as mechanism and the total number of DDIs that were classified as mechanism (including the ones wrongly assigned to this type). Similarly, the recall for mechanism relationships is defined as the ratio between the number of DDIs correctly classified as mechanism and the total number of DDIs with the mechanism type in the gold standard. The precision and recall for the rest of the DDI types are defined in a similar manner. The overall evaluation considered two aspects.

1) System's performance in identifying interaction pairs.
2) System's performance in classifying the interactions pairs into predefined types.

TABLE I: DRUGBANK DATASET MICRO-AVERAGED RESULTS FOR ENTITY DETECTION

Team	Recall	Precision	F-score
FBK-irst	0.838	0.816	0.827
WBI	0.755	0.814	0.783
SCAI	0.681	0.796	0.734
UTurku	0.638	0.843	0.726
UC3M	0.758	0.656	0.703
Our system	**0.893**	**0.884**	**0.888**

TABLE II: MEDLINE DATASET MICRO-AVERAGED RESULTS FOR ENTITY DETECTION

Team	Recall	Precision	F-Score
FBK-irst	0.505	0.558	0.530
WBI	0.421	0.625	0.503
UWMTRIAD	0.630	0.387	0.479
SCAI	0.526	0.431	0.474
UC3M	0.642	0.313	0.421
Our system	**0.703**	**0.762**	**0.731**

TABLE III: DRUGBANK DATASET MACRO-AVERAGED RESULTS FOR TYPE CLASSIFICATION

Team	Recall	Precision	F-Score
FBK-irst	0.639	0.708	0.672
WBI	0.575	0.666	0.617
UTurku	0.507	0.777	0.614
NIL_UCM	0.498	0.651	0.565
UC3M	0.566	0.557	0.561
UWMTRIADS	0.485	0.487	0.486
Our system	**0.748**	**0.750**	**0.749**

TABLE IV: MEDLINE DATASET MACRO-AVERAGED RESULTS FOR TYPE CLASSIFICATION

Team	Recall	Precision	F-Score
FBK-irst	0.514	0.384	0.440
WBI	0.333	0.376	0.353
UWMTRIADS	0.413	0.297	0.345
UCOLORADO_SOM	0.380	0.212	0.272
SCAI	0.197	0.420	0.269
Our system	**0.683**	**0.561**	**0.572**

In this study, our system adopts an automatic information extraction (IE) approach, and no post-processing is applied to the system's output. This is the general requirement of the SemEval 2013 task, which was used to evaluate our system. As a whole, our system was successful compared to other participating systems, as shown in Table I, Table II, Table III and Table IV.

V. CONCLUSION

Extracting DDI information from biomedical text is a promising area of research for understanding the effect of one drug in the presence of another. The availability of a large amount of data adds complications to the understanding of DDIs and their effects. In this work, we explored text mining methods to automatically extract drug–drug information from text. It may be observed that there are significant differences in micro-averaged and macro-averaged results for participating systems. Generally, systems have better results with the DrugBank dataset and weaker results with Medline. Participating systems were also better at predicting interaction pairs than at identifying interaction types.

Our system achieved the best results of 0.893 recall rate, 0.884 precision rate, and 0.888 F-score. In comparison, the first rank system at SemEval 2013 achieved 0.838 recall, 0.816 precision and 0.827 F-score (Table I). For the Medline dataset, our system also performed best (Table II). Similarly, Table III and Table IV present the results for labeling of interactions for both DrugBank and Medline corpora for macro-averages for all types. Our system ranked top 2nd and 1st, respectively.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Both authors contributed design and implementation of the research, the analysis of the result and the writing of the manuscript.

REFERENCES

[1] L. Yin, F. Bryan, H. Hong, Y.-C. Tu, W. Ju, and C. Feng, "Characterization of the mechanism of drug-drug interactions from PubMed using MeSH terms," *PLoS One*, vol. 12, p. e0173548, 2017.

[2] B. Richard, G. Gregory, and H. Henk, "Using natural language processing to identify pharmacokinetic drug-drug interactions described in drug package inserts," *Association for Computational Linguistics*, 2012.

[3] S. B. Isabel, M. Paloma, and S. C. Daniel, "The 1st DDIExtraction-2011 challenge task: Extraction of drug-drug Interactions from biomedical texts," in *Proc. the 1st Challenge Task on Drug-Drug Interaction Extraction (DDIExtraction 2011)*, 2011, pp. 1–9.

[4] S.-B. Isabel, M. Paloma, and Z. M. Herrero, "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013)," in *Proc. Second Joint Conference on Lexical and Computational Semantics*, 2013, pp. 341–350.

[5] H.-Z. Mar á, S.-B. Isabel, M. Paloma, and D. Thierry, "The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions," *Journal of Biomedical Informatics*, vol. 46, pp. 914–920, 2013.

[6] C. Aurel, M. Shannon, A. Alana, and B. Y. Reis, "Pharmacointeraction network models predict unknown drug-drug interactions," *PloS one*, vol. 8, 2013.

[7] F. X. Cheng and Z. M. Zhao, "Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties," *Journal of the American Medical Informatics Association*, vol. 21, pp. e278–e286, 2014.

[8] D. G. Huang, Z. C. Jiang, L. Zou, and L. S. Li, "Drug–drug interaction extraction from biomedical literature using support vector machine and long short term memory networks," *Information Sciences*, vol. 415, pp. 100–109, 2017.

[9] K. Park, D. Kim, S. Ha, and D. Lee, "Predicting pharmacodynamic drug-drug interactions through signaling propagation interference on protein-protein interaction networks," *PloS one*, vol. 10, 2015.

[10] Q. C. Bui, P. M. A. Sloot, E. M. Van Mulligen, and J. Kors, "A novel feature-based approach to extract drug-drug interactions from biomedical text," *Bioinformatics*, vol. 30, pp. 3365–3371, 2014.

[11] C. M. F. Mahbub and L. Alberto, "FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information," in *Proc. Second Joint Conference on Lexical and Computational Semantics*, 2013, pp. 351–355.

[12] S. Kim, H. B. Liu, Y. Lana, and W. W. John, "Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach," *Journal of Biomedical Informatics*, vol. 55, pp. 23–30, 2015.
[13] H. Sepp and S. Jürgen, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.
[14] L. Yann, B. Yoshua *et al*., "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, p. 1995, 1995.
[15] R. J. Williams and Z. David, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.
[16] S. S. Kumar and A. Ashish, "Drug-drug interaction extraction from biomedical texts using long short-term memory network," *Journal of Biomedical Informatics*, vol. 86, pp. 15–24, 2018.
[17] S.-P. V ćtor, S.-B. Isabel, and M. Paloma, "Exploring convolutional neural networks for drug-drug interaction extraction," *Database*, 2017.
[18] Y. J. Zhang, W. Zheng, H. F. Lin, J. Wang, Z. H. Yang, and D. Michel, "Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths," *Bioinformatics*, vol. 34, pp. 828–835, 2017.
[19] Z. H. Zhao, Z. H. Yang, L. Luo, H. F. Lin, and J. Wang, "Drug-drug interaction extraction from biomedical literature using syntax convolutional neural network," *Bioinformatics*, vol. 32, pp. 3444–3453, 2016.
[20] A. Ibrahim, F. Achille, H. Oktie, Z. Ping, and S. Mohammad, "Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 44, pp. 104–117, 2017.
[21] D.-S. Cao, N. Xiao, Y.-J. Li *et al*., "Integrating multiple evidence sources to predict adverse drug reactions based on a systems pharmacology model," *CPT: Pharmacometrics and Systems Pharmacology*, vol. 9, pp. 498–506, 2015.
[22] S. Dhanya, F. Shobeir, and G. Lise, "A probabilistic approach for collective similarity-based drug-drug interaction prediction," *Bioinformatics*, vol. 32, pp. 3175–3182, 2016.
[23] V. Santiago, U. Eugenio, S. Lourdes *et al*., "Similarity-based modeling in large-scale prediction of drug-drug interactions," *Nature Protocols*, vol. 9, p. 2147, 2014.
[24] W. Zheng, H. F. Lin, Z. H. Zhao *et al*., "A graph kernel based on context vectors for extracting drug-drug interactions," *Journal of Biomedical Informatics*, vol. 61, pp. 34–43, 2016.
[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
[26] M. A. Nielsen, *Neural Networks and Deep Learning*, Determination press San Francisco, CA, USA, 2015.
[27] B. Yoshua, S. Patrice *et al*., "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, 1994.
[28] A.-R. Rami, C. Dokook, C. Noah, G. Mandy, and J. Llion, "Character-level language modeling with deeper self-attention," in *Proc. the AAAI Conference on Artificial Intelligence*, 2019, pp. 3159–3166.
[29] Z. H. Dai, Z. L. Yang, Y. M. Yang *et al*., "Transformer-xl: Attentive language models beyond a fixed-length context," arXiv preprint arXiv:1901.02860, 2019.
[30] E. Maha, B. Laurent, and V. Jakob, "Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction," arXiv preprint arXiv:1808.03867, 2018.
[31] W.-J. Hou and C. Bamfa, "Extraction of drug-drug Interaction Using Neural Embedding," *Journal of Bioinformatics and Computational Biology*, vol. 16, no. 6, 1840027, pp. 1–22, 2018.
[32] D. S. Wishart, Y. D. Feunang, A. C. Guo *et al*. DrugBank. [Online]. Available: https://www.drugbank.ca/

Wen-Juan Hou received a B.Sc. and an M.Sc. in information and computer education from the National Taiwan Normal University, in Taipei, Taiwan, R.O.C., in 1991 and 1993, respectively. She completed a Ph.D. in computer science and information engineering from the National Taiwan University, in Taipei, Taiwan, R.O.C., in 2007. She is currently the assistant professor of the Department of Computer Science and Information Engineering at the National Taiwan Normal University. Her areas of research interest include text mining in biomedical documents, semantic analysis in texts, and related topics in natural language processing.

Bamfa Ceesay received a B.Sc. in computer science and information engineering from the National Taipei University of Technology, in Taipei, Taiwan R.O.C in 2011, followed by an M.Sc. in computer science and information engineering from the National Taiwan Normal University in 2014. His areas of research interest include text mining in biomedical documents, event extraction, bioinformatics, and natural language processing.