IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2019, accepted July 14, 2019, date of publication July 23, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930641

Drug-Drug Interaction Extraction Based on Transfer Weight Matrix and Memory Network

JUAN LIU[1,2], ZHONG HUANG[1,2,3], FUJI REN[2,3,4], (Senior Member, IEEE), AND LEI HUA[2]

[1]School of Physics and Electronic Engineering, Anqing Normal University, Anqing 246133, China
[2]Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, Hefei 230009, China
[3]School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[4]Faculty of Engineering, University of Tokushima, Tokushima 770-8501, Japan

Corresponding author: Zhong Huang (huangzhong3315@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61432004 and Grant 61702012, in part by the Natural Science Foundation of Anhui Province of China under Grant 1908085MF195, in part by the Natural Science Research Project of the Education Department of Anhui Province under Grant KJ2017A368 and Grant KJ2017A549, and in part by the Open Project of the Anhui Province Key Laboratory under Grant ACAIM180203.

ABSTRACT Extracting drug–drug interaction (DDI) in the text is the process of identifying how two target drugs in a given sentence interact. Previous methods, which were limited to conventional machine learning techniques, we are susceptible to issues such as "*vocabulary gap*" and unattainable automation processes in feature extraction. Inspired by deep learning in natural language preprocessing, we addressed the aforementioned problems based on dynamic transfer matrix and memory networks. A TM-RNN method is proposed by adding the transfer weight matrix in multilayer bidirectional LSTM to improve robustness and introduce a memory network for feature fusion. We evaluated the TM-RNN model on the DDIExtraction 2013 Task. The proposed model achieved an overall F-score of 72.43, which outperforms the latest methods based on support vector machine and other neural networks. Meanwhile, the experimental results also indicated that the proposed model is more stable and less affected by negative samples.

INDEX TERMS Drug–drug interaction extraction, memory network, multilayer bidirectional LSTM, transfer weight matrix.

I. INTRODUCTION

Drug–drug interaction (DDI) is a situation where one drug increases or decreases the effect of another drug entity [1], [2]. According to the survey of [3], the number of individuals who take multiple drugs simultaneously has considerably increased. The interactions amongst these drugs may be harmful to the human body. Hence, building a reliable DDI system or database is necessary to avoid certain drug abuse medical accidents. Meanwhile, with the rapid growth of biomedical scientific publications (for example, the MedLine database has doubled in size in the past ten years), the need for an automatic DDI extraction system is urgent.

In recent years, a growing number of researchers have focused on DDI extraction based on conventional machine learning techniques and achieved meaningful works. Mathematically, DDI extraction can be considered a classification problem, that is, a decision should be made whether a relation (binary classification) or what kind of relation (multiclass classification) exists between two drug entities. The most representative methods are the rule-based and statistical machine learning (SML) methods. However, these methods are susceptible to issues, such as 'vocabulary gap' and unattainable automation processes in feature extraction. Inspired by deep learning in natural language preprocessing, we propose a drug–drug interaction extraction method based on transfer weight matrix and memory network. Fig. 1 illustrates the basic outline of the proposed framework.

Firstly, a pretrained word embedding, which can map a single word into a fixed-size dense vector, is used to represent input sentences. Secondly, a bidirectional long short-term memory (LSTM) with dynamic transfer weight is used. A sequence model is constructed for the input sentences, and a network with different depths and structures is generated. This step aims to extract the long-distance dependency of words in the drug relationship sentence and improve robustness. Finally, the extracted features of the multilayer LSTM with dynamic transfer weight are decomposed into memory

The associate editor coordinating the review of this manuscript and approving it for publication was Jihad Aljaam.

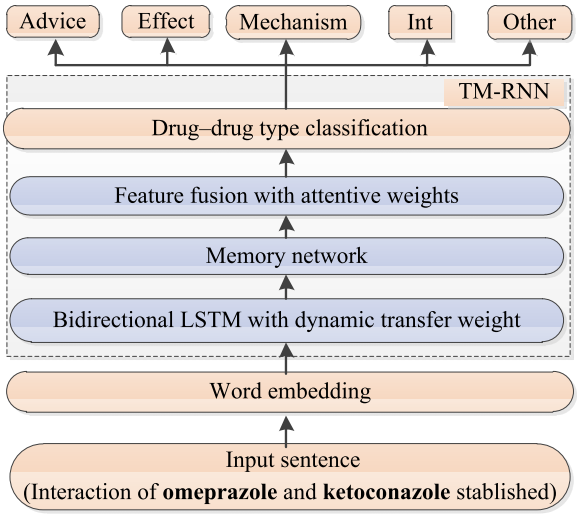


FIGURE 1. **The framework of the proposed method, the bold words are entities mentioned in input sentence.**

and attentive information via matrix transformations. A feature fused strategy is introduced into the proposed model for drug–drug type classification.

The main innovation points are as follows:

(1) The proposed bidirectional LSTM with transfer weight matrix can not only encode input sentences with dynamic input and capture the long-term dependency of input sentences, but also generate a network with varying depths and structures. Thus, overfitting is overcome, and robustness is improved. In the drug–drug interaction extraction task, this technique renders the model insensitive to negative samples, which can help build the model without using various pretreatments (e.g. predefined filter rules).

(2) The memory network, which is introduced into the proposed model, is used to split the deep semantic feature of input sentences into memory space for information storage and attention spaces for attentive weight calculation. Feature fusion is implemented on the basis of attentive weights, thereby avoiding the crosstalk amongst input information and highlighting the important effect of keywords on drug–drug interaction. In this manner, the model becomes interpretable instead of just a black box.

(3) Compared with other recurrent neural network (RNN)-based methods, the biggest difference of the proposed TM-RNN is that the transfer weight matrix (which is firstly used in convolutional neural network (CNN)) is introduced into the multilayer bidirectional LSTM to generate different networks. A memory network addresses the attentive hidden state instead of only the last hidden state to highlight the role of different words in a sentence. As such, the overfitting problem can be prevented, and full advantage can be taken of all hidden states.

The rest of this paper is organized as follows: Section II summarizes the related work. Section III details the proposed TM-RNN along with the important components; Section IV evaluates the experimental results; Section V concludes our work.

II. RELATED WORK

A large, gold standard dataset has been annotated in DDI task due to the first community-wide competition: DDIExtraction 2013 [2]. DDI extraction has captured much interest recently, and a series of studies [4] on this challenging workshop has been reported. Previous works have exploited many technologies for biomedical relation extraction, especially feature engineering approaches, and have been proven to be effective in many fields (i.e. protein–protein interaction [5]–[7] and DDI extraction [4], [8]). In general, these methods fall under two categories, namely, rule- and SML-based methods.

Rule-based methods often use predefined patterns or rules to match the pretreated or labeled sequences [9]. Despite the sophisticated design in patterns or rules, such methods suffer from low recall, which deviate them from practical usage. Different from rule-based methods, most SML-based methods consider the relation extraction task as a standard classification task. That is, the relation (binary classification) or the kind of relation (multiclass classification) between the two entities should be determined. SML-based methods [10], [11] have been proven to perform much better than rule-based methods and have achieved state-of-the-art results on many relation extraction tasks. Conventional SML-based methods for DDI extraction often collect n-grams or part-of-speech (POS) around two aimed drug entities as features. These features are then converted into discrete (i.e. one-hot) representations. Lastly, these representations are fed into classifiers, such as support vector machine (SVM) [12] or maximum entropy model (MXENT), to determine the type of relation between two target entities. However, such methods cannot encode sequence information and fail to capture semantic information amongst input words. For the former issue, these models are insufficient to capture the semantic information and account for differences in word order or syntactic structure (e.g. 'cat sat on mat' versus 'mat sat on cat'). The second issue can be rendered as 'vocabulary gap'. The words 'rely' and 'depend' are far from each other in one-hot representation despite their similarity in meaning and grammar.

Neural network-based methods [3], [13]–[16] have been utilised by several works for DDI tasks because of the natural advantage of recurrent neural network (RNN) in dealing with sequential data and the great success in natural language preprocessing [17], [18]. Liu *et al.* [13] firstly proposed the use of CNN for a DDI extraction task, which outperformed conventional SML methods. However, CNN-based methods can only extract a limited size of word window of input sentence, thus making the capture of long-term dependency difficult due to the special structure of the CNN. Compared with CNN-based methods, Sahu and Anand [3] showed that RNN-based methods can achieve better results in the

DDI extraction task, because RNN-based methods handle the input as sequential data rather than word windows.

III. METHOD

In this section, we firstly introduce the basic concepts of word embedding since our method is built on it; secondly, we introduce the RNN with transfer weight; thirdly, we introduce our feature fusion model; at last, we define the loss function and show how to train the model.

A. WORD EMBEDDING

Word embedding [18], [19] is the collective name for a set of language modelling and feature learning techniques in natural language processing, in which words or phrases from the vocabulary are mapped into fix-sized dense vectors. The basic idea under word embedding is that words with a similar context have similar semantics. Many methods including Glove [20], word2vec [19], Senna [17] had been proposed to train word embedding. In this paper, we adopt the word embedding trained by word2vec [21] to initialize the words appearing in DDI extraction task (all pre-trained word embedding can be obtained from.[1])

For the mathematical notation, we use $X = [w_1, \cdots, w_t \cdots, , w_N]$ to represent the input sentence, where each vector $w_t \in R^d$ in X is the word embedding for t-th input word.

B. MULTILAYER BIDIRECTIONAL LSTM WITH TRANSFER WEIGHT MATRIX

Almost all methods for the DDI classification task search for a way to encode input raw sentences to capture semantic or grammatical information well. Manually generated features, such as n-grams, parser tree, and dependency parser tree, are widely used in DDI extraction tasks. However, the accuracy of generating features is difficult to guarantee. Meanwhile, the propagation of errors can seriously affect the performance of the model. Hence, a bidirectional LSTM with dynamic transfer weight is firstly proposed to encode input sentences and eventually capture word order and long word dependency information. LSTM is a variant of the basic RNN [22]. Equation (1) demonstrates the full description for LSTM.

$$\begin{aligned} i_t &= \sigma\left(W_{ix} w_t + W_{ih} h_{t-1} + b_i\right) \\ f_t &= \sigma\left(W_{fx} w_t + W_{fh} h_{t-1} + b_f\right) \\ o_t &= \sigma\left(W_{ox} w_t + W_{oh} h_{t-1} + b_o\right) \\ g_t &= \tanh\left(W_{gx} w_t + W_{gh} h_{t-1} + b_g\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh\left(c_t\right), \end{aligned} \tag{1}$$

where w_t represents the t-th pretrained word embedding vector of input sentence; h_{t-1} and h_t represent the previous and current hidden states, respectively; (W_{ix}, W_{ih}, b_i), (W_{fx}, W_{fh}, b_f), (W_{ox}, W_{oh}, b_o) and (W_{gx}, W_{gh}, b_g) are the weight and bias matrixes of the input, forget and output gates and memory cell, respectively; $\sigma(\cdot)$ and $\tanh(\cdot)$ are activation functions. The implied concept under LSTM uses the parameters of the input (W_{ix}, W_{ih}, b_i), output (W_{ox}, W_{oh}, b_o) and forget (W_{gx}, W_{gh}, b_g) gates to control the flow of information and use the memory cell c_t to store the gradient, which can overcome the gradient vanishing problem.

[1] http://bio.nlplab.org/

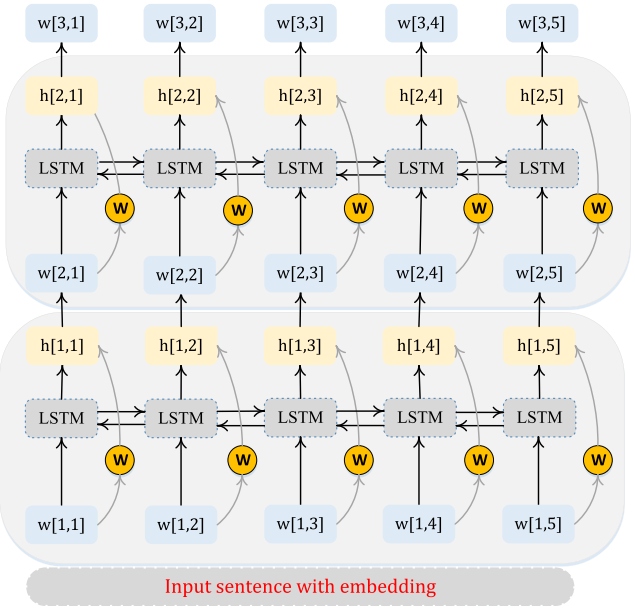


FIGURE 2. A multi-layer bi-directional LSTM with dynamic transfer weight (The gray line donates transfer weight matrix).

Addressing to the DDI task, we feed a word embedding vector w_t into LSTM in each time step and generate the hidden state h_t as features for the latter work. To address previous and future features well, we use a multilayer bidirectional LSTM to extract the features, as shown in Fig. 2.

The multilayer bidirectional LSTM is introduced into TM-RNN, and $w[k, t]$ is used to represent the t-th input of the k-layer and distinguish the input data of various layers at different moments. Given this representation, the input embedding vector w_t in the first layer can be regarded as $w[1, t]$. Meanwhile, the t-th hidden output of the k-layer can be calculated as follows:

$$\begin{aligned} h\left[k,t\right]^f &= LSTM\left(w\left[k,t\right], h\left[k,t-1\right]^f\right) \\ h\left[k,t\right]^b &= LSTM\left(w\left[k,t\right], h\left[k,t-1\right]^b\right) \\ h\left[k,t\right] &= h\left[k,t\right]^f \oplus h\left[k,t\right]^b, \end{aligned} \tag{2}$$

where $h\left[k,t\right]^f$ and $h\left[k,t\right]^b$ represent the t-th hidden output of the forward and backward LSTM network, respectively, in the k-th layer. $h[k, t]$ is the transformation of the original input. Considering the original information will be at risk of vanishing when training a large and deep model, a transfer weight matrix $W[k]$ of the k-th layer is introduced to the constructed deep network model. The input $w\left[k+1, t\right]$ of the $k+1$-th LSTM layer is calculated by (3) instead of only

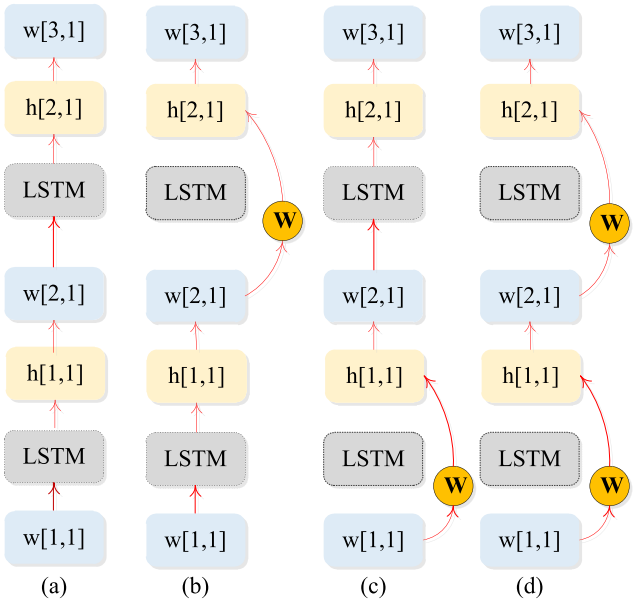


FIGURE 3. An example to explain transfer weight matrix with 2-layer LSTM model.

feeding $h[k,t]$ to $w[k+1,t]$ directly:

$$w[k+1,t] = w[k,t]\,W[k] + h[k,t]. \tag{3}$$

To explain the motivation behind transfer weight matrix, we take the final $w[3, 1]$ in Fig. 2 as an example and its four generation paths are shown in Fig. 3.

The first generation of $w[3, 1]$ (Fig. 3(a)) can be expressed as:

$$\begin{aligned} w[2,1] &= h[1,1] = LSTM\,(w[1,1], h[1,0]) \\ h[2,1] &= LSTM\,(w[2,1], h[2,0]) \\ w[3,1] &= h[2,1] \\ &= LSTM\,(LSTM\,(w[1,1], h[1,0]), h[2,0]). \end{aligned} \tag{4}$$

Based on (4), we find that the transfer weight matrixes of two layers are not functioning, indicating that $w[k,t]\,W[k]$ is close to zero. In this manner, the input of the $k+1$-th layer is determined as the output of the k-th layer only.

In contrast to the first generation, the transfer weight matrix in Fig. 3(b) works on the output of the first layer. The generation of $w[3, 1]$ can be expressed as:

$$\begin{aligned} w[2,1] &= h[1,1] \\ h[2,1] &= w[2,1]W[2] \\ w[3,1] &= h[2,1] = LSTM\,(w[1,1], h[1,0])\,W[2]. \end{aligned} \tag{5}$$

Similarly, in Fig. 3(c), the transfer weight matrix only works on the input of the first layer. Hence, the generation of $w[3, 1]$ can be expressed as:

$$\begin{aligned} w[2,1] &= h[1,1] = w[1,1]W[1] \\ h[2,1] &= LSTM\,(w[2,1], h[2,0]) \\ w[3,1] &= h[2,1] = LSTM\,(w[1,1]W[1], h[2,0]). \end{aligned} \tag{6}$$

Contrary to the former three generations, the transfer weight matrixes in Fig. 3(d) work on two layers, and the generation of $w[3, 1]$ can be expressed as:

$$\begin{aligned} w[2,1] &= h[1,1] = w[1,1]W[1] \\ h[2,1] &= w[2,1]W[2] \\ w[3,1] &= h[2,1] = w[1,1]W[1]W[2]. \end{aligned} \tag{7}$$

Based on (7), we find that the two LSTM modules do not need to work, and the input of the $k+1$-th layer is determined as the input of the k-th layer only. Given the above, the 2^K model combinations can be generated by dynamic transfer weight matrixes with the K-layer network. Meanwhile, the input for the k-th layer can come from a linear transformation of the first layer input. These characteristics not only endow the proposed model with excellent robustness but also overcome the overfitting problem when training a large and deep model.

C. FEATURE FUSION WITH MEMORY NETWORK FOR DRUG-DRUY TYPE CLASSIFICATION

The proposed multilayer bidirectional LSTM with transfer weight matrix implements the feature extraction of drug–drug interaction. Instead of taking only the last hidden output of LSTM as final features to classify the drug–drug interaction type, the attention mechanism is used to access all hidden states. In the traditional attention-based model, all operations are conducted on features, which may cause a crosstalk in information. Inspired by memory network [23], [24] and transformer by Google's recent work [25], [26], we split the deep semantic feature of input sentences into memory space for information storage and attention spaces for attentive weight calculation. Moreover, feature fusion based on attentive weights is implemented to avoid crosstalk amongst input information. The details are shown in Fig. 4.

Firstly, the t-th input of the k-layer $w[k,t]$ is decomposed into memory and attentive information via matrix transformations:

$$\begin{aligned} m[t] &= w[k,t]\,U \\ a[t] &= w[k,t]\,T, \end{aligned} \tag{8}$$

where U and T represent transformation matrixes. $m[t]$ is regarded as memory, which is used to store information in each time step; whereas $a[t]$ is rendered as attention, which is used to generate attentive weight for memory access. By splitting $w[k,t]$ into memory and attention spaces, it can effectively avoid crosstalk amongst the input information. Meanwhile, memory cell $m[t]$ can alleviate the long-distance dependence of words by storing the information of all time steps.

Secondly, the attention weight $s[t]$ is calculated based on $a[t]$:

$$s[t] = Softmax_t\left(v^T a[t]\right), \tag{9}$$

where v is an attentive coefficient, and SoftMax is a normalised function, which compresses the input value into

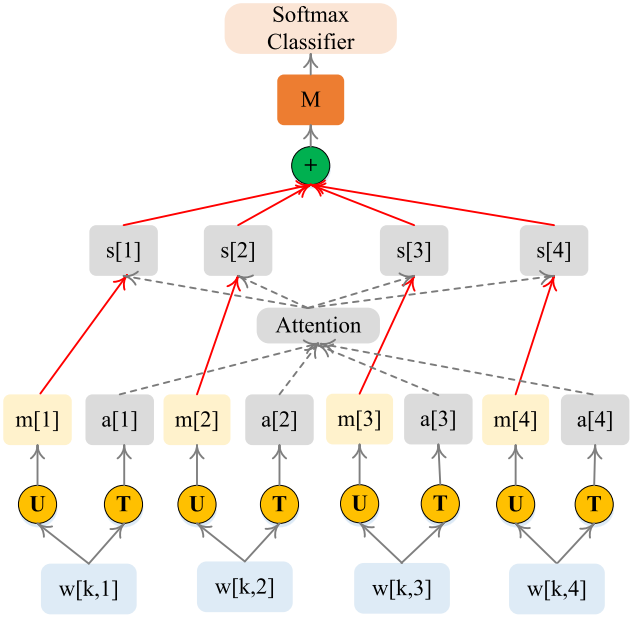


FIGURE 4. Feature fusion with memory network for drug-drug type classification.

[0,1]. Meanwhile, the final output M is the weighted sum over all memories:

$$M = \sum_{t=1}^{N} m(t)^* s(t), \tag{10}$$

where N represents the number of words in the input sentence, and M is regarded as the fusion information of a given sentence $X = [w_1, \cdots, w_t \cdots, w_N]$.

Finally, the fusion information M is fed to a SoftMax classifier for classification. Let C denote the classes of drug–drug interaction. The probability of each class X belongs to

$$o = \text{Soft max}(MV + d), \tag{11}$$

where V is a transformation matrix, and d is the bias term.

Cross entropy is taken as a loss function to train the parameters of the proposed model. Suppose Y_i is the one-hot representation of the true label, and o_i is the final output for the i-th input instance, the final cross-entropy loss can be calculated as

$$loss = -\log\left(Y_i^T o_i\right) \tag{12}$$

Meanwhile, an improved gradient descent method based on Adadelta [27] is used to update the parameters in each training step.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

We use PyTorch to implement our proposed model.[2] The related hardware configurations are as follows: Intel(R) Xeon(R) CPU E5-2650 V3@2.3 GHz, NVidia Tesla K40m GPU, 8 GB Memory. The related parameters of TM-RNN are set as follows: The embedding size is 200, the hidden units for LSTM are set to 64, and the learning rate for Adadelta is set to 0.5. In order to get a better results and reduce the randomness, 10 epochs over training datasets are needed and the tenfold cross validation is performed in the following statistics.

TABLE 1. The Number of Samples for Train and Test.

	Train		Test	
	Drug Bank	Med Line	Drug Bank	Med Line
Advice	815	7	214	7
Effect	1517	152	298	62
Mechanism	1257	62	278	21
Int	178	10	94	2
Other(before preprocessing)	22118	1547	4367	345
Other(after preprocessing)	14445	1179	2819	243

A. DDIEXTRACTION 2013

To illustrate the effectiveness of the proposed method, we conduct an analysis and experimental comparison with related methods on DDIExtraction 2013, which is composed of DrugBank and MedLine [2]. This dataset focuses on the extraction of drug–drug interactions that appear in biomedical literature. One of the main purposes of this dataset is to pursue the classification of each drug–drug interaction according to one of the following five types: Advice, Effect, Mechanism, Int and Other. Therefore, DDIExtraction is regarded as a five-label classification task.[3] The basic descriptions and examples for each type are as follows: (1) Advice: a recommendation or advice regarding the concomitant use of two drugs involved (e.g. '**UROXATRAL** should not be used in combination with other **alpha-blockers**'); (2) Effect: the effect of drug–drug interaction is described (e.g.: '**Quinolones** may enhance the effects of the oral anticoagulant, **warfarin**'); (3) Mechanism: the mechanism of interaction, which can be pharmacodynamic or pharmacokinetic (e.g. '**Grepafloxacin** is a competitive inhibitor of the metabolism of **theophylline**'); (4) Int: states that an interaction occurs (e.g. 'The interaction between **omeprazole** and **ketoconazole** has been established'); (5) Other: no relation between entities.

In our experiments, we preprocess the DDIExtraction 2013 dataset based on our early method [28]. In summary, preprocessing consists of two parts, namely, 1) using rules to filter negative instances ('Other' instances) and 2) replacing the two target drugs with special symbols, such as DRUG1 and DRUG2. Other drug entities with the symbol DRUG0 can refer to the detailed illustrations [28]. Unless otherwise stated, the following experiments are based on the preprocessed datasets. Table. I demonstrates the number of samples for training and testing. Notably, we do not

[2]The source code of the proposed model can be found in https://github.com/coddinglxf/DDI-with-rnn.

[3]https://www.cs.york.ac.uk/semeval-2013/task9/

TABLE 2. Three Statistics of Baseline and TM-RNN.(Both Models use Two-Layer Bi-LSTM).

	Baseline			TM-RNN		
	P	R	F	P	R	F
Int	88.57	32.29	47.33	82.22	38.54	**52.48**
Advice	81.87	63.93	71.79	81.44	72.15	**76.51**
Effect	64.25	69.47	66.76	69.27	71.99	**70.60**
Mechanism	79.78	71.48	75.40	74.13	78.86	**76.42**
Overall	73.57	65.15	69.11	74.11	70.82	**72.43**

TABLE 3. Comparisons Between 1-Layer TM-RNN and 2-Layer TM-RNN.

	1-layer TM-RNN			2-layer TM-RNN		
	P	R	F	P	R	F
Int	73.08	39.58	51.53	82.22	38.54	**52.48**
Advice	79.49	70.58	74.88	81.44	72.15	**76.51**
Effect	68.21	74.51	71.22	69.27	71.99	**70.60**
Mechanism	61.81	75.50	67.98	74.13	78.86	**76.42**
Overall	68.33	70.52	69.41	74.11	70.82	**72.43**

distinguish between DrugBank and MedLine in training and testing.

B. EXPERIMENTAL RESULTS OF THE PROPOSED TM-RNN

To illustrate the advantages of the proposed model, we compare our method to the baseline model, which was utilised based on the same multilayer bidirectional LSTM without the transfer weight matrix and memory network. For evaluation, precision (P), recall (R) and F-scores (F) for each class are calculated as follows:

$$\begin{aligned} P &= TP/(TP + FP) \\ R &= TP/(TP + FN) \\ F &= \left(2^*P^*R\right)/(P + R) \end{aligned} \tag{13}$$

where TP represents that a positive instance is correctly classified as a positive one, FN represents that a positive instance is misclassified as a negative one, whereas FP denotes that the negative instance is misclassified as a positive one. Table. II displays the three statistics of the baseline and TM-RNN.

Table. II shows that TM-RNN achieves better results compared with the baseline model. For individual interaction type classification, TM-RNN also achieves high F-scores, which illustrates the importance and effectiveness of the transfer weight matrix and memory network.

Meanwhile, to analyse the influences of the numbers of hidden layers on the proposed method, we also compare the results between 1- and 2-layer TM-RNN. Table. III provides the experimental results.

Table. III shows that even the 1-layer TM-RNN can obtain results that are comparable to the baseline model. However, by training a deeper model, the 2-layer TM-RNN achieves much better results and further improves the overall F-scores by 3.02. Moreover, we test the 3- and 4-layer TM-RNN

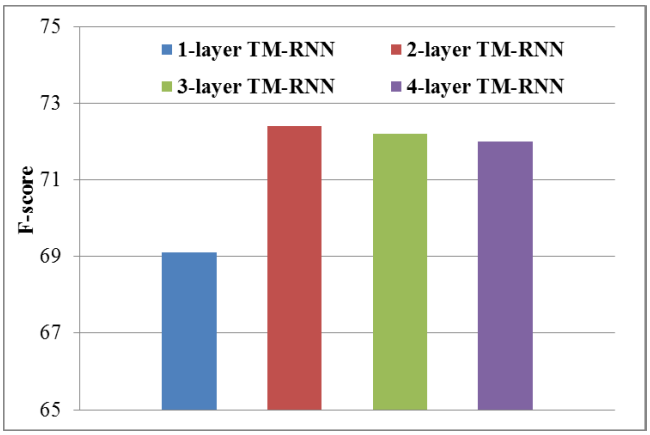


FIGURE 5. Comparisons of the overall F-score of different-layers TM-RNN.

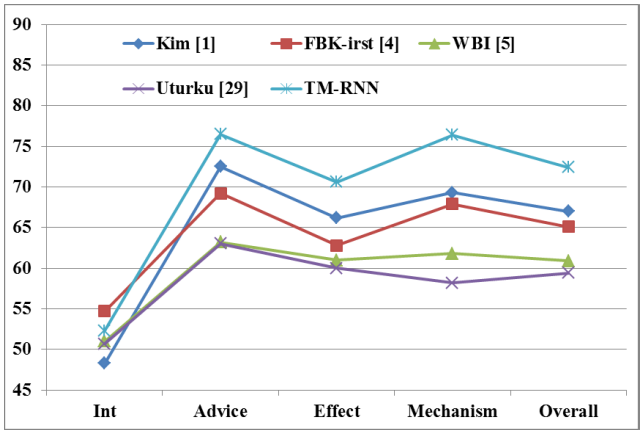


FIGURE 6. Comparison with other conventional SML-based methods.

and the F-score results are shown in Fig. 5. Fig. 5 illustrates that the results of the 3- or 4-layer TM-RNN are almost the same as the 2-layer TM-RNN. The probable explanation is that the 3- or 4-layer TM-RNN requires further data for training. Hence, the 2-layer TM-RNN seems to be an appropriate choice, considering the limited data in the current DDIExtraction 2013 dataset.

C. COMPARISON WITH OTHER METHODS

In this subsection, we compare the performance of the proposed method with other conventional SML-based and neural network methods. Fig. 6 lists the performance of Kim's linear SVM-based method [1], FBK-irst [4], WBI [8] and UTurku [29].

For these conventional SML-based methods, manually generated features (i.e. parser tree, dependency parser tree and POS) still account for a large percentage. All four models use SVM to classify the drug–drug interaction types. The distinction is mainly focused on the classification strategy. WBI [8] and UTurku [29] regard DDI extraction as a standard multi-classification problem, whereas FBK-irst [4] and Kim [1]'s systems firstly detect the interaction (binary classification) and then classify the interaction into a specific type.

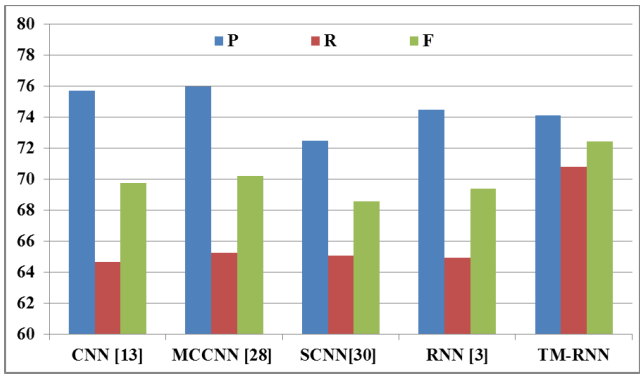


FIGURE 7. Comparisons with other neural networks based methods.

Results also indicate that FBK-irst [4] and Kim [1]'s models outperformed WBI [8] and UTurku [29]. In this study, we avoid feature engineering and classification strategy choice problems by using the neural network methods. Compared with Kim's linear kernel method [1], the proposed TM-RNN improves overall F-scores by 5.4.

We also compare the performance of the proposed method with other neural network methods, as shown in Fig. 7. Liu *et al.* [13] and Zhao *et al.* [30] firstly utilised CNN with word embedding for DDI tasks. Quqan *et al.* [28] proposed the use of multichannel word embedding to capture rich semantic information. They slightly improved the overall F-scores by 0.46 compared with Liu's work [13], which indicated that a single channel word embedding might contain enough information for a DDI task. For this reason, we focus on the structure of the model rather than external resources in the study. From Fig. 7, we can see that CNN based models can achieve better accuracy. This is mainly because CNN based models are constructed based on the word window. For positive samples, more accurate information can be extracted by max-pooling operation, however for negative sample data, the word window lacks contextual information, CNN based models are more sensitive to noise data. Considering the large number of negative samples in the DDI task, in this paper, we use LSTM to model the input sentences.

Sahu and Anand [3] reported the results of using RNN for DDI task. Our study differs, because we introduce the memory network and propose to use attention in separate spaces, namely, attention and memory spaces. Moreover, a multilayer bidirectional LSTM with transfer weight matrix is designed in our method. Hence, compared with Sahu's method [3], the proposed TM-RNN further improves the overall F-scores by 3.04. Our experiments indicate that certain intuitive observations and design from the neural network structure can help improve the results.

D. EFFECT OF PREPROCESSING

As shown in Table. I, although we have adopted the preprocessing rules, the negative sample continues to occupy a large ratio of the training set. Therefore, the DDI extraction task is an unbalanced classification problem. A robust system should be insensitive to negative samples to eliminate interference from preprocessing rules. In this section, we compare the performance of the proposed method with CNN- or RNN-based methods in the original datasets (without preprocessing rules), as shown in Table. IV (Δ indicates the gain of F-scores from preprocessing). According to Table. IV, we find that our method is less affected by negative instances compared with other models, which suggests that our model has better generalisation. Conversely, CNN seems to be more sensitive to noise instances compared with RNN-based methods.

TABLE 4. Performance Improvement After Preprocessing.

Methods	Before	After	Δ
CNN [13]	65.00	69.75	4.75
MCCN [28]	67.80	70.21	2.41
SCNN [30]	64.50	68.43	4.10
RNN [3]	67.28	69.39	2.11
TM-RNN	**70.82**	**72.43**	**1.61**

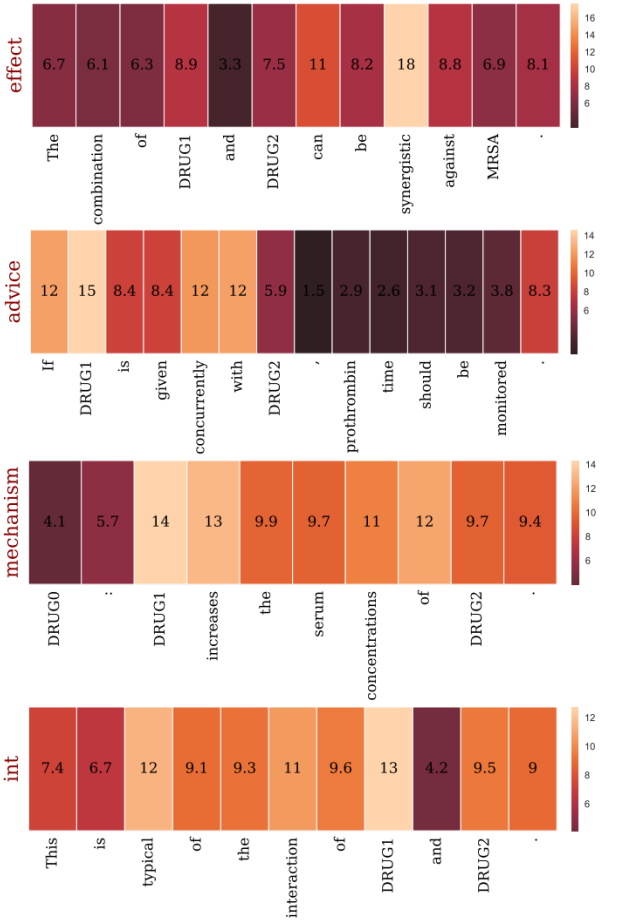


FIGURE 8. The heatmap of attentive weight for sampled data of each interaction type.

E. VISUALISATIONS OF ATTENTIVE WEIGHT

One advantage of splitting $w[k, t]$ into memory and attention spaces is that we can visualise the most important

features by tracking the attentive weight $a[t]$, as described in (8). Fig. 8 shows the visualisation of attentive weight. We found that words (i.e. **synergistic**, **increase** and **interaction**), which strongly indicate drug–drug interaction, can be correctly marked (with high weights in Fig. 8) by our proposed model. Compared with other conventional black box methods, our proposed model can measure the contribution of each word to the final SoftMax classifier. Meanwhile, the special symbol DRUG1 is always assigned with high weight in our experiments, as shown in Fig. 8 (all attentive weights are multiplied by a constant value of 100 for display convenience). An intuitive explanation is that in most cases, the words between a pair of drug entities are enough to decide the interactions between two drug entities instead of entire sentences. By assigning DRUG1 with a high weight, the proposed TM-RNN is equivalent to the extraction of boundary information and the learning of certain manual features to a certain extent.

V. CONCLUSION

In this study, we propose a multilayer bidirectional LSTM with transfer weight matrix and memory network to solve the DDI classification problem. Results highlight the following points: a multilayer bidirectional LSTM with transfer weight and attention mechanism should be considered as a powerful structure in a DDI task. The transfer weight matrix is an effective mean to prevent overfitting, whereas the memory network, which decomposes input information into memory and attentive spaces, ensures that the proposed model can completely access all stored memory. In addition, our model is less sensitive to negative samples compared with other methods, considering that many irrelevant samples exist in drug–drug relation extraction tasks. This advantage helps us discard artificial features completely (or pretreatments). Although our proposed TM-RNN has been able to achieve comparable results, a reliable and precise DDI classification system should be further improved from the following aspects: (1) Although the LSTM can encode any sentence length, an extremely long sentence and two target drug entities that are too far from each other will still introduce noise information. This aspect will hinder the performance of the TM-RNN. To solve this problem, the most straightforward way is to introduce parser tree information, because drug entities may be structurally close to each other although far away in word sequence. (2) A typical error is that the input sample will be misclassified as 'Other', which is equivalent to noise in the datasets. Punishing the 'Other' label or modifying the loss function to release the problem will be the next step in our future work.

REFERENCES

[1] K. Sun, H. Liu, L. Yeganova, and W. J. Wilbur, "Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach," *J. Biomed. Inform.*, vol. 55, no. 2015, pp. 23–30, Mar. 2015.

[2] S. I. Bedmar, P. Martínez, and M. H. Zazo, "SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 341–350.

[3] S. K. Sahu and A. Anand, "Drug-drug interaction extraction from biomedical texts using long short-term memory network," *J. Biomed. Inform.*, vol. 86, pp. 15–24, Oct. 2018.

[4] M. F. M. Chowdhury and A. Lavelli, "FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 351–355.

[5] S.-P. Choi and S.-H. Myaeng, "Simplicity is better: Revisiting single kernel PPI extraction," in *Proc. COLING*, Beijing, China, 2010, pp. 206–214.

[6] Z. Yang, N. Tang, X. Zhang, H. Lin, Y. Li, and Z. Yang, "Multiple kernel learning in protein–protein interaction extraction from biomedical literature," *Artif. Intell. Med.*, vol. 51, no. 3, pp. 163–173, Mar. 2011.

[7] L. Li, R. Guo, Z. Jiang, and D. Huang, "An approach to improve kernel-based protein–protein Interaction extraction by learning from large-scale network data," *Methods*, vol. 83, pp. 44–50, Jul. 2015.

[8] P. Thomas, M. Neves, T. Rocktäschel, and U. Leser, "WBI-DDI: Drug-drug interaction extraction using majority voting," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 628–635.

[9] I. Segura-Bedmar, P. Martínez, and C. D. Pablo-Sánchez, "Using a shallow linguistic kernel for drug–drug interaction extraction," *J. Biomed. Inform.*, vol. 44, no. 5, pp. 789–804, Apr. 2011.

[10] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "A rich feature vector for protein-protein interaction extraction from multiple corpora," in *Proc. EMNLP*, Singapore, 2009, pp. 121–130.

[11] M. Miwa, R. Sætre, Y. Miyao, and J. Tsujii, "Protein–protein interaction extraction by leveraging multiple kernels and parsers," *Int. J. Med. Inform.*, vol. 78, no. 12, pp. e39–e46, Dec. 2009.

[12] B. Cui, H. Lin, and Z. Yang, "SVM-based protein-protein interaction extraction from medline abstracts," in *Proc. BIC-TA*, Zhengzhou, China, Sep. 2007, pp. 182–185.

[13] S. Liu, B. Tang, Q. Chen, and X. Wang, "Drug-drug interaction extraction via convolutional neural networks," *Comput. Math. Methods Med.*, vol. 2016, pp. 1–8, Jan. 2016.

[14] T. Ching *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *J. Roy. Soc. Interface*, vol. 15, no. 141, pp. 1–123, 2018.

[15] S. Lim, K. Lee, and J. Kang, "Drug drug interaction extraction from the literature using a recursive neural network," *PLoS ONE*, vol. 13, no. 1, Jan. 2018, Art. no. e0190926.

[16] Y. Zhang, W. Zheng, H. Lin, J. Wang, Z. Yang, and M. Dumontier, "Drug–drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths," *Bioinformatics*, vol. 34, no. 5, pp. 828–835, Mar. 2018.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. 1, pp. 2493–2537, Mar. 2011. [Online]. Available: https://arxiv.org/abs/1103.0398

[18] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. 6, pp. 1137–1155, Jan. 2000.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Sep. 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781

[20] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1532–1543.

[21] S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou, "Distributional semantics resources for biomedical text processing," in *Proc. LBM*, 2013, pp. 39–44.

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

[23] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Dec. 2017.

[24] J. Weston, S. Chopra, and A. Bordes, "Memory networks," 2015, *arXiv:1410.3916*. [Online]. Available: https://arxiv.org/abs/1410.3916

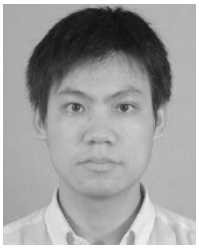
[25] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://blog.openai.com/language-unsupervised/

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2019, *arXiv:1810.04805*. [Online]. Available: https://arxiv.org/abs/1810.04805

[27] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: https://arxiv.org/abs/1212.5701

[28] C. Quan, L. Hua, X. Sun, and W. Bai, "Multichannel convolutional neural network for biological relation extraction," *BioMed. Res. Int.*, vol. 2016, no. 1, pp. 1–10, Jan. 2016.
[29] J. Björne, S. Kaewphan, and T. Salakoski, "UTurku: Drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge," in *Proc. SemEval*, Atlanta, GA, USA, 2013, pp. 651–659.
[30] Z. Zhao, Z. Yang, L. Luo, H. Lin, and J. Wang, "Drug drug interaction extraction from biomedical literature using syntax convolutional neural network," *Bioinformatics*, vol. 32, no. 22, pp. 3444–3453, Nov. 2016.

JUAN LIU received the M.S. degree from the Hefei University of Technology, China, in 2009. Her research interests include bioinformatics, natural language processing, and machine Learning.



ZHONG HUANG received the Ph.D. degree from the School of Computer and Information, Hefei University of Technology, China, in 2016. He became an Associate Professor with the School of Physics and Electronic Engineering, Anqing Normal University, in 2018. His research interests include bioinformatics, affective computing, and computer vision.

FUJI REN received the Ph.D. degree from the Faculty of Engineering, Hokkaido University, Japan, in 1991. He became a Professor with the Faculty of Engineering, University of Tokushima, in 2001. His research interests include artificial intelligence, language understanding and communication, and affective computing. He is a Fellow of the Engineering Academy of Japan, the EU Academy of Sciences, the Japan Federation of Engineering Societies, and the Institute of Electronics, Information and Communication Engineers, the President of the International Advanced Information Institute.

LEI HUA received the M.S. degree from the Hefei University of Technology, China, in 2017. Her research interests include natural language processing, bioinformatics, and machine learning.