IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON HUMAN-CENTERED SMART SYSTEMS AND TECHNOLOGIES

Received April 30, 2018, accepted May 29, 2018, date of publication June 11, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2845840

Leveraging Biomedical Resources in Bi-LSTM for Drug-Drug Interaction Extraction

BO XU[1], XIUFENG SHI[1], ZHEHUAN ZHAO[1], AND WEI ZHENG[2,3]

[1]Key Laboratory for Ubiquitous Network and Service Software of Liaoning, School of Software, Dalian University of Technology, Dalian 116024, China
[2]College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[3]College of Software, Dalian JiaoTong University, Dalian 116028, China

Corresponding author: Zhehuan Zhao (z.zhao@dlut.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61502071 and Grant 61572094 and in part by the Fundamental Research Funds for the Central Universities under Grant DUT18RC(3)004.

ABSTRACT The discovery of drug–drug interaction (DDI) is not only critical in understanding the mechanism of medicine, but also aids in preventing medical error and controlling healthcare costs. When physicians or pharmacists prescribe multiple drugs simultaneously to a single patient, DDI can be a crucial piece of information to keep the patient from experiencing adverse reactions or any other potential physical harm. Therefore, it is necessary to extract DDI for human healthcare and medicinal safety. Researchers have studied this using the literature mining methods. Recently, biomedical resources have been applied successfully in literature mining tasks, such as machine reading. Because biomedical resources contain a large amount of valuable information, we attempt to leverage this resource to provide professional knowledge in the procedure of DDI extraction. We propose a new bidirectional long–short-term memory (LSTM) network-based method, namely, biomedical resource LSTM (BR-LSTM), which combines biomedical resource with lexical information and entity position information together to extract DDI from the biomedical literature. We conducted experiments on the SemEval 2013 task 9.2 data set to evaluate our method. BR-LSTM outperforms the other state-of-the-art methods and achieves a competitive F-score of 0.7115.

INDEX TERMS Concept embedding, drug drug interaction, drug safety, human healthcare, long short-term memory.

I. INTRODUCTION

Administering multiple drugs on a single patient simultaneously has become a considerably common phenomenon these days. In the process of the prescription, fully understanding interactions between drugs, such as adverse or advise, positive or negative, can prevent patients from drug adverse reactions and side effects. As FDA Adverse Events Reporting System (FAERS) reported, there are 10,830,881 adverse events in last ten years (2008-2017) including 1,109,868 death reports and there is an upper trend of occurrence of adverse events report. So the study on drug-drug interaction (DDI) is imperative and urgent for human healthcare and medicine safety.

Traditionally, medical practitioners primarily employ two manners to learn about drug-drug interaction: reading tremendous biomedical papers to acquire drug-drug interactions between the lines or querying a human-maintained biomedical database to search for drug-drug interactions. Obviously, physicians would be easily bewildered in huge biomedical literature. Sinking into the ocean of rapidly-updated papers is laborious, painful, and inefficient. Querying a biomedical database seems to be feasible, but in the consideration of the scale of the biomedical literature, it requires a lot of manpower and resources to maintain a professional database manually. It is too difficult to keep the pace with the publication of papers for these biomedical databases. Both two of them are not idealistic methods to detect DDIs for doctors.

Nowadays, there has been a surge of great interest in how to extract DDIs. Researchers perceive DDI extraction to be a natural language processing task currently. As an illustration, in the following sentence:

> Because of its primary CNS effect, caution should be used when **EQUETROTM** is taken with other centrally acting drugs and **alcohol**.

DDI extraction methods attempt to predict the interaction type between two recognized entities (highlighted in

bold text). The ultimate goal of DDI extraction task is to extract DDI from biomedical literature automatically, precisely, and efficiently.

A lot of related work has been done via two kinds of methods: traditional machine learning model based methods and deep learning model based methods.

Most traditional machine learning models based methods use support vector machine based models. They utilize manually engineered features and are heavily dependent on external lexical or syntactical resources such as part-of-speech tags and shallow parsing. Existing support vector machine based methods can be classified into two categories: non-linear SVM models and linear SVM models. A typical non-linear SVM based method is FBK-irst [1]. It uses two support vector machine models with non-linear kernels. FBK-irst is a two-stage DDI extraction method. As a preliminary step, FBK-irst extracts positive DDIs from datasets with a binary SVM classifier. Contextual and shallow linguistic features are used in this step. The next step is to classify the extracted DDIs into different categories. Kim *et al.* [2] is a typical linear SVM model based method. It is equipped with a rich set of lexical and syntactic features. Kim *et al.* is also a two-stage method.

Deep learning model based methods use various types of deep learning network models such as convolutional neural networks (CNN) and recurrent neural networks (RNN) to detect DDI. These models learn features automatically rather than construct features manually. Deep learning model based methods also can be classified into two categories: CNN based methods and RNN based methods. Syntax Convolutional Neural Network (SCNN) is a CNN based method which utilizes a novel word embedding, syntax word embedding, to employ syntactic information in a sentence. Then a combination of embedding-based convolutional features and traditional features are fed to a Softmax classifier to extract DDIs. Another illustration of CNN based method is DCNN [3]. DCNN is a a dependency-based convolutional neural network for DDI extraction. DCNN performs convolution operation on adjacent words in dependency parsing trees besides word sequences of candidate DDI instances. In this way, long distance dependencies between words in the candidate DDI instance are incorporated into DCNN. Joint AB-LSTM [4] is a bidirectional LSTM based model. LSTM [5], [6] is a variant of RNN model. Joint AB-LSTM applied max pooling and attentive pooling to returned hidden state sequences from two separate Bi-LSTM models. Then the concatenation of the two pooling results is fed into a Softmax classifier to predict DDIs.

Recently, biomedical resource has been proved to be a helpful information source in many other studies such as medical semantic similarities measurement [7], [8] and patient similarities measurement [9]. To our knowledge, existing methods do not attach enough importance to biomedical resource during the procedure of DDI extraction yet. So we try to take advantage of biomedical resources to improve the performance of DDI extraction.

In this study, we propose a biomedical resource long short term memory (BR-LSTM) based DDI extraction method. In addition to normal lexical information and position information, biomedical domain-specific knowledge is utilized in BR-LSTM. All parts of information is condensed into a joint embedding feature as input of a BR-LSTM. The experimental results show that BR-LSTM is prior to other state-of-the-art methods.

The rest of this paper is organized as follows. Section II presents the details of the proposed method. In Section III, we describe experimental settings and results. Finally, our conclusion is presented in Section IV.

II. METHOD

The BR-LSTM method for DDI extraction is a three-step procedure:

1) In *preprocessing step*, three preprocessing operations are conducted: negative instance filtering, drug blinding, and tokenization.
2) In *embedding generation step*, biomedical resources are encoded into embeddings (low-dimensional vectors) which are learned by medical concept embedding method. The biomedical resource concept embeddings are joint with word embeddings and entity offset embeddings as input of our model.
3) In *Bi-LSTM network step*, we feed joint embeddings into a Bi-LSTM network to predict types of DDI.

Fig. 1 illustrates the basic pipeline of the BR-LSTM method.

A. PREPROCESSING

In preprocessing step of the BR-LSTM, three preprocessing operations are conducted: negative instance filtering, drug blinding, and tokenization.

1) NEGATIVE INSTANCE FILTERING

In the training dataset of DDI extraction task, the ratio of the positive instances to the negative instances is 1:5.91. Imbalanced datasets have been proved to exert a strong bad influence on the performance of the model [10]. And previous methods [4], [11] have verified that negative instance filtering is an effective operation to rectify the imbalanced condition.

We follow the same under-sampling policies of the SCNN. Under-sampling policies of the SCNN can be summarized into two rules:

Rule 1: If two drugs in a drug pair refer to the same drug, this pair will be removed under the assumption that the same drug can not interact with itself. There are two cases to be considered: the first case is two drug names are the same and the second case is one drug is the abbreviation of the other drug.

Rule 2: If two drugs in a drug pair are in coordinate relations, this pair will be removed since it is prone to be a false positive [12].

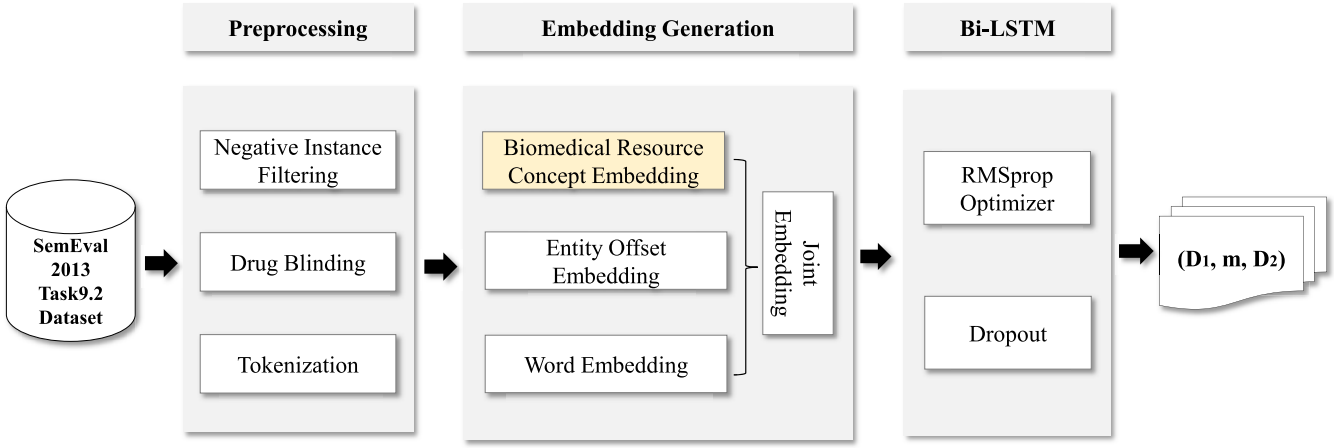

FIGURE 1. The basic pipeline of the BR-LSTM method.

2) DRUG BLINDING

Drug blinding preprocessing method replaces concrete drug entity names with special tokens. It has shown to be effective in overcoming overfitting problem [13], [14]. So we also take drug blinding in our preprocessing step. We replace two drug entity names in a drug pair with token **DURG1** and token **DRUG2**, and replace other drug entity names in the same sentence with token **DRUGN**.

3) TOKENIZATION

After the drug blinding phase, we tokenize sentences in datasets with GENIA Tagger [15]. Tokenization converts a sentence into a sequence of tokens. During the tokenization procedure, we also replace numbers with the token **NUM** and remove all punctuations and some selected meaningless stop words.

B. EMBEDDING GENERATION

We embed our processed data to a set of joint embeddings that consist of four embedding fragments: a word embedding, two entity offset embeddings, and a concept embedding.

1) WORD EMBEDDING

The word embedding is a distributed representation of a token. Compared with sparse one-hot representation, dense word embeddings are extremely storage-saving and memory-saving. Word embeddings compress lexical information under the assumption that one word can be represented by its adjacent context words. Word embedding has become a common information source in natural language processing models. It has been applied in a lot of natural language processing tasks such as sentiment analysis [16] and machine translation [17] successfully. We utilize word embeddings to represent lexical context information in sentences. Our word embeddings were induced on a combination of PubMed and PMC texts with texts extracted from a recent English Wikipedia dump using the Word2vec [18], [19] tool.

2) ENTITY OFFSET EMBEDDING

We use entity offset embeddings to indicate the position of target drug entities [11]. Firstly, we generate the two offsets for each token by calculating the distances between current token and two drug entities, respectively. Then, the generated offsets are mapped to the real value vectors using corresponding look-up table.

3) BIOMEDICAL RESOURCE CONCEPT EMBEDDING

DDI extraction is an interdisciplinary research subject that covers biomedical science, computer science, and linguistics. Due to the interdisciplinary nature of DDI extraction, the background knowledge is important for this task. So we attempt to take advantage of biomedical resources to enhance the performance of DDI extraction. In our method, the biomedical resources information is encoded into embeddings (low-dimensional vectors) which are learned by medical concept embedding method [7]. Concepts are biomedical entities from structured medical ontologies such as Unified Medical Language System (UMLS) or the International Classification of Diseases (ICD-9, ICD-10). Concept embedding maps each concept to a low-dimensional vector. The training process of the biomedical resource concept embedding of our method is presented in Fig. 2. Two corpora are used to train the concept embedding: MedTrack that is a collection of clinical patient records and OHSUMED that is a collection of MEDLINE medical journal abstracts. First, token lists from two corpora are converted into sequences of UMLS concepts by the Metamap [20]. Then UMLS concept sequences are input into the Skip-gram algorithm to train the concept embedding. These concept embeddings are pre-trained for our BR-LSTM model.

In the BR-LSTM, sentences from datasets are converted to a collection of concept sequences by the Metamap. After the conversion, concept sequences are mapped to corresponding concept embedding vectors by a pre-trained concept embedding loop-up table.

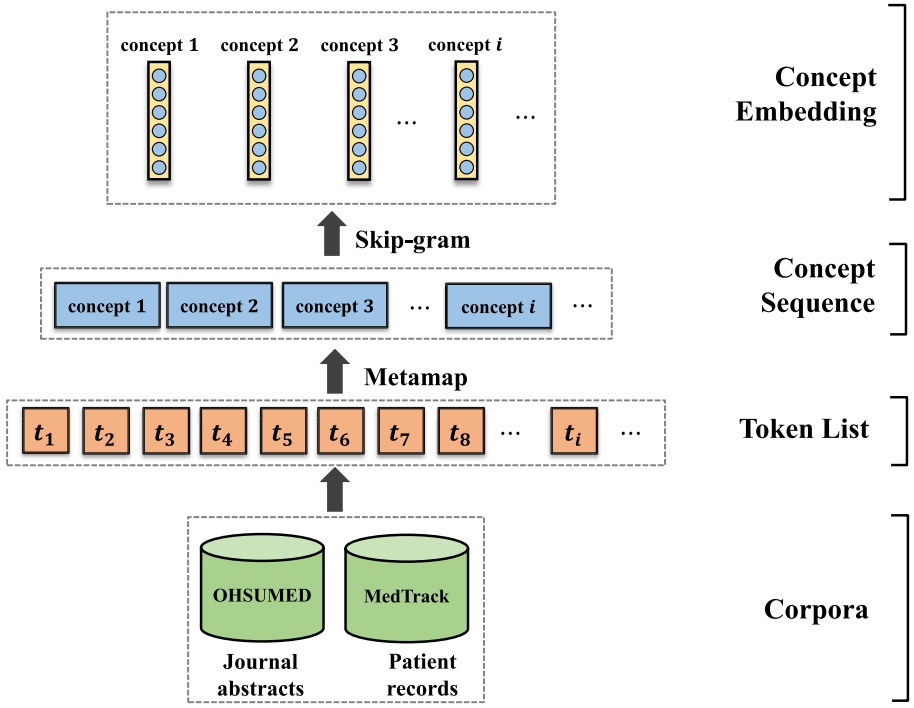

FIGURE 2. The training process of the concept embedding.

4) JOINT EMBEDDING

With regarding to the token t_i, the final joint embedding consists of word embedding vector $emb^i_w \in \mathbb{R}^{d_w}$, DRUG1 offset embedding vecotr $emb^i_{o1} \in \mathbb{R}^{d_o}$, DRUG2 offset embedding vector $emb^i_{o2} \in \mathbb{R}^{d_o}$ and concept embedding vector $emb^i_c \in \mathbb{R}^{d_c}$. The joint embedding of t_i is defined as:

$$x_i = [emb^i_w, emb^i_{o1}, emb^i_{o2}, emb^i_c] \in \mathbb{R}^{(d_w+2\times d_o+d_c)}. \quad (1)$$

C. BIDIRECTIONAL LSTM NETWORK

The architecture of the bidirectional long short-term memory network which is used in BR-LSTM is presented in Fig. 3. Joint embedding vector sequences of sentences are fed into a bidirectional long short-term memory network (Bi-LSTM) to extract a sentence embedding vector. At every time-step, an embedding vector is input until the embedding vector sequence of a sentence is exhausted. LSTM network generates two states at every single time-step: a cell state that is transferred into the next time-step and a hidden state that is the output vector of the time-step.

h_{t-1} and C_{t-1} are the hidden state and the cell state of the previous time-step $t-1$, h_t and C_t are the hidden state and the cell state of the current time-step t. h_t and C_t are defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (2)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (3)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \quad (4)$$
$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t, \quad (5)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (6)$$
$$h_t = o_t \circ \tanh(C_t). \quad (7)$$

Weights (W_f, W_i, W_C, and W_o)and biases (b_f, b_i, b_C, and b_o) are all trainable parameters. The hidden state at the last time-step is the output of the whole LSTM network.

In a bidirectional LSTM network, joint embedding sequences (forward) and corresponding reverse version (backward) are fed into two LSTMs, respectively. Then, we obtain the forward output hidden state ($\overrightarrow{h}_n$) and the backward output hidden state ($\overleftarrow{h}_1$), respectively. The concatenation of two hidden states is the final output of the Bi-LSTM network:

$$h_n = [\overleftarrow{h}_1, \overrightarrow{h}_n]. \quad (8)$$

After the process of the Bi-LSTM network, our sentences are encoded into sentence embeddings h_n.

The last portion of BR-LSTM is to make classification of DDI types using a Softmax classifier which computes a normalized probability score for every DDI type. $p(y|\mathcal{X}) \in \mathbb{R}^{|M|}$ is the probability distribution of DDI types. $\mathcal{X}$ is a sequence of joint embeddings that is generated from a sentence and $|M|$ is the count of DDI types. $p(y|\mathcal{X})$ is defined as:

$$p(y|\mathcal{X}) = Softmax(W_s h_n + b_s). \quad (9)$$

Weight W_s and bias b_s are also trainable parameters.

III. EXPERIMENTS

We implemented our model with Python and Keras deep learning programming API [21] running on the top of

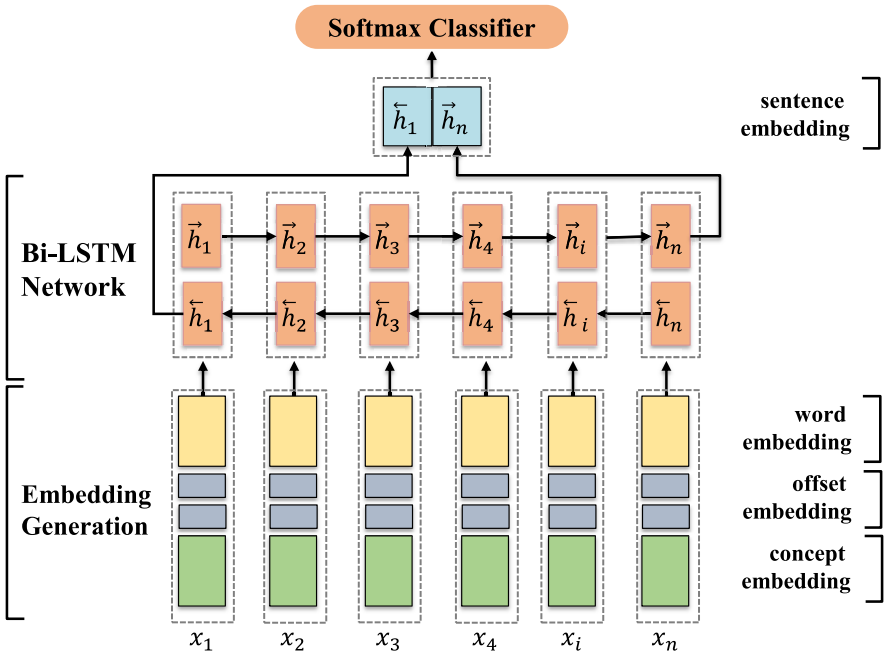

FIGURE 3. The architecture of the bidirectional long short-term memory network which is used in BR-LSTM.

TensorFlow [22] backend. Keras has a lot of built-in neural network models, so we can construct our model with Keras easily and efficiently. We trained our model on a powerful NVIDIA TITAN Xp GPU (12GB).

A. DATA AND EVALUATION METRICS

We evaluated our model on datasets of SemEval 2013 task9.2 [23]. The datasets consist of 27, 792 drug pairs and 5, 716 drug pairs in the training set and the test set respectively, the count ratio of drug pairs is 4.86 : 1. Four positive DDI types (*advise*, *mechanism*, *int* and *effect*) and a negative DDI type (*negative*) were assigned to drug pairs. Details of the count of every single interaction type before and after the negative instance filtering are showed in Table 1. We used official evaluation metrics, precision (P), recall (R), and F-score (F), to test the performance of our method. Let $M = \{\text{false, mechanism, effect, advise, int}\}$ denotes the DDI label set, the precision and recall of each $m \in M$ are calculated by

$$P_m = \frac{\#\text{ DDI is } m \text{ and is classified as } m}{\#\text{ Classified as } m}, \quad (10)$$

$$R_m = \frac{\#\text{ DDI is } m \text{ and is classified as } m}{\#\text{ DDI is } m}. \quad (11)$$

Then the overall precision, recall and F-score are defined as:

$$P = \frac{1}{|M|}\sum_{m\in M} P_m, \quad (12)$$

$$R = \frac{1}{|M|}\sum_{m\in M} R_m, \quad (13)$$

$$F = \frac{1}{|M|}\frac{2PR}{P+R}. \quad (14)$$

TABLE 1. Class distribution before and after the negative instance filtering phase.

Class	Training Set		Test Set	
	Before	After	Before	After
negative	23772	8987	4737	2049
advise	826	814	221	221
effect	1687	1592	360	357
mechanism	1319	1260	302	301
int	188	188	96	92
ratio	1:5.91	1:2.33	1:4.83	1:2.11
Total	27792	12841	5716	3020

B. HYPER-PARAMETERS

We used a pre-trained Word2vec word embedding [24] and a pre-trained concept embedding. Their dimensions are both set to 200. And we used the dynamic word embedding strategy i.e. word vectors are updated during the training process, whereas the concept embedding is static. DRUG1 offset embedding and DRUG2 offset embedding shared the same embedding look-up table whose dimension is 10. Offset embedding vectors were also dynamic.

The hidden state dimension was 200 in our Bi-LSTM layer. To overcome the overfitting problem, we used the dropout technique in our model with a drop rate value 0.5. We also used several L1-L2 regularizers [25] inside our method, the weight decay value was 0.006.

We trained our model with a RMSprop optimizer [26], the learning rate we used was 0.001, other hyper-parameters of optimizers were all default values. Our training batch size

was 128 and the zero-masking strategy was used to train on variance-length sequences.

The full set of hyper-parameters is listed in Table 2.

TABLE 2. Hyper-parameter list.

Name	Value
word embedding dimension	200
concept embedding dimension	200
offset embedding dimension	10
hidden state dimension	200
weight decay	0.006
dropout probability	0.5
batch size	128
RMSprop-learning rate	0.001
RMSprop-ρ (rho)	0.9
RMSprop-ϵ (epsilon)	1e-08
RMSprop-learning rate decay	0

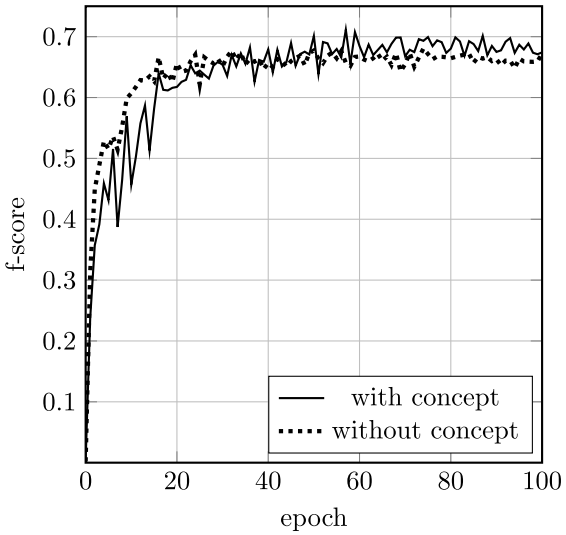

FIGURE 4. F-score comparison of with and without concept embedding.

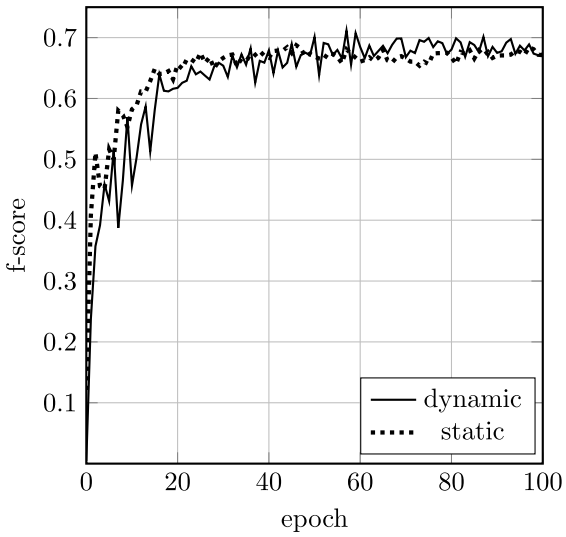

FIGURE 5. F-score comparison of dynamic and static embedding.

C. EXPERIMENTAL RESULT

In the experiment, our method outperformed other state-of-the-art methods. Table 3 shows the prediction confusion matrix for each DDI type and Table 4 shows the evaluation results of each DDI type. As can be seen, ''int'' DDI type only achieves an F-score of 0.4336 which is much lower than that of other three DDI types. It obtains a recall rate of 0.3229 that causes the lowest F-score. The reason is that addition to ''negative'' type, ''int'' DDIs also be misclassified as ''effect'' types while most of other three types of DDIs are only misclassified as ''negative'' type, which is shown in Table 3. The ''mechanism'' type achieves the highest F-score, this is probably because ''mechanism'' describes a pharmacokinetic mechanism. Pharmacokinetic means that the effects of one drug are changed by the presence of another drug at its site of action, so it is relatively not easy to confuse. Evaluation metrics comparison with other state-of-the-art methods are listed in the Table 5. The highest values of every column are highlighted in bold text. Our methods achieved the highest F-score of 0.7115 and the highest recall of 0.7079 in the comparison.

1) PERFORMANCE COMPARISON

The first four methods in Table 5 are all support vector based methods. Support machine vector machine based models always utilize high-level handcrafted features. Kim *et el.* and FBK-irst are top-2 support vector machine based methods. Kim *et al.* uses a linear SVM classifier with a rich set of lexical and syntactic features to extract DDI. FBK-irst uses a SVM classifier with a hybrid kernel. FBK-irst utilizes syntax tree and dependency tree features. They are all highly dependent on the external lexical tools such as part-of-speech tagger, chunk tagger and dependency parse to analyze text and construct features from the analysis. The performance of external lexical tools will influence the performance of DDI extraction models directly. Their structures of features are designed by researchers. Designing features manually is laborious and time-consuming. In the other hand, human-selected features may perform poor on generalization. Compared with FBK-irst and Kim *et al.*, BR-LSTM learns high-level features automatically and depends on no external lexical analyzer. However, BR-LSTM achieves an F-score of 0.7115 which is higher than that of FBK-irst and Kim *et al.* by 0.1015 and 0.1205, respectively.

Remaining methods in Table 5 are all deep learning based methods. They can be categorized into two types: convolutional neural network based methods and bidirectional long short-term memory network based methods. SCNN [11] is a syntax convolutional neural network based DDI extraction method. In SCNN, a novel word embedding, syntax embedding, is proposed to employ syntactical information of a sentence. SCNN also employs many traditional features

TABLE 3. Classification confusion matrix.

True label	Predicted label					Total
	Negative	Mechanism	Effect	Advise	Int	
negative	4531	43	115	34	14	4737
mechanism	65	231	2	4	0	302
effect	80	3	272	5	0	360
advise	47	4	9	159	2	221
int	24	1	40	0	31	96
Total	4747	282	438	202	47	5716

TABLE 4. Detailed evaluation metrics of BR-LSTM.

	TP	FP	FN	Total	P	R	F
mechanism	231	51	71	302	0.8191	0.7649	0.7911
effect	272	166	88	360	0.621	0.7556	0.6817
advise	159	43	62	221	0.7871	0.7195	0.7518
int	31	16	65	96	0.6596	0.3229	0.4336
classification	693	276	286	979	0.7152	0.7079	0.7115

TABLE 5. Performance comparison between methods.

Methods	P	R	F
UTurku [2]	0.732	0.499	0.594
WBI-DDI [4]	0.642	0.579	0.609
FBK-irst [1]	0.646	0.656	0.651
Kim *et al.* [3]	—	—	0.670
SCNN[1] [5]	0.691	0.651	0.670
SCNN[2] [5]	0.725	0.651	0.686
Joint AB-LSTM [7]	0.7447	0.6496	0.6939
DCNN [6]	**0.7722**	0.6435	0.7019
BR-LSTM	0.7152	**0.7079**	**0.7115**

to improve the performance of DDI extraction that makes the extraction process more complicated. Compared with SCNN, BR-LSTM extracts no traditional feature but achieves a higher F-score (0.7115 vs. 0.686).

Joint AB-LSTM [4] is a bidirectional long short-term memory based DDI extraction method. It merges two separate bidirectional long short-term memory network to generate a sentence embedding. Two bidirectional long short-term memory network utilize the max-pooling and attentive pooling methods to generate sentence embeddings. BR-LSTM uses only one bidirectional long short-term memory network to extract DDI. Compared with Joint AB-LSTM, BR-LSTM achieves higher F-score (0.7115 vs. 0.6939) with a simpler architecture. The reason may be that BR-LSTM employs the external biomedical resource which is ignored by Joint AB-LSTM. The domain knowledge will provide useful information for DDI extraction.

2) FEATURE ANALYSIS

Biomedical resources play a key role in BR-LSTM. To evaluate the effectiveness of biomedical resource concept embedding, we comepared the performances of our method with and without concept embedding in Fig. 4. The method with biomedical resource fluctuates more violently but gains a higher value. The method without concept embedding stagnates too early and its performance is poorer from a long-term perspective. The F-score of the method without the biomedical resource is 0.6805 and the F-score is improved by 0.031 with the utilization of the biomedical resource.

Whether to update word embeddings during the training phase is another important factor of the performance. Training word embeddings and model parameters at the same time can make the model fit the dataset more quickly. If we keep the word embeddings static, the lexical information remains constant at the state of pre-trained time. We may be mislead by the original word embedding corpora. The drawback of the dynamic word embedding is that it will increase the number of learnable parameters and pose risk of overfitting. Updating word embedding during the procedure of training also needs huge computation power. We compare the F-score of dynamic and static embedding in Fig. 5. Fig. 5 indicates that dynamic embedding can improve the performance notably. Due to more parameters of dynamic word embedding method, its optimization speed is slower than that of the static one. But after a while, the dynamic word embedding method can surpass the static word embedding method. The F-score of the method using a static embedding is 0.687 and the F-score is improved by 0.0245 with the dynamic embedding.

IV. CONCLUSIONS

In this study, we proposed a novel methods namely BR-LSTM to extract DDI from biomedical literature automatically. The biomedical resources are encoded into concept embeddings. These concept embeddings are jointed with word embeddings and entity offset embeddings as the input of BR-LSTM. The experimental result shows that our method gains the highest evaluation metrics values.

ACKNOWLEDGMENT

The authors would like to thank Prof. Zeng You of the School of Software, Dalian University of Technology, Dalian, China, for assistance with the revision of this paper and comments of their method. His comments greatly improved the manuscript.

REFERENCES

[1] 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. 2nd Joint Conf. Lexical Comput. Semantics, 7th Int. Workshop Semantic Eval.*, Atlanta, GA, USA, Jun. 2013, pp. 351–355.

[2] S. Kim, 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, pp. 23–30, Jun. 2015.

[3] S. Liu, K. Chen, Q. Chen, and B. Tang, "Dependency-based convolutional neural network for drug-drug interaction extraction," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2016, pp. 1074–1080.

[4] S. K. Sahu and A. Anand. (2017). "Drug-drug interaction extraction from biomedical text using long short term memory network." [Online]. Available: https://arxiv.org/abs/1701.08303

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

[6] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, 2000.

[7] L. De Vine, G. Zuccon, B. Koopman, L. Sitbon, and P. Bruza, "Medical semantic similarity with a neural language model," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 1819–1822.

[8] Y. Choi, C. Y.-I. Chiu, and D. Sontag, "Learning low-dimensional representations of medical concepts," *AMIA Summits Transl. Sci. Process.*, vol. 2016, pp. 41–50, Jul. 2016.

[9] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang, "Measuring patient similarities via a deep architecture with medical concept embedding," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 749–758.

[10] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Inf. Sci.*, vol. 250, pp. 113–141, Nov. 2013.

[11] 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, 2016.

[12] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "Lessons learnt from the DDIExtraction-2013 shared task," *J. Biomed. Inform.*, vol. 51, pp. 152–164, Oct. 2014.

[13] S. Liu, B. Tang, Q. Chen, and X. Wang, "Drug-drug interaction extraction via convolutional neural networks," *Comput. Math. Methods Med.*, vol. 2016, 2016, Art. no. 6918381.

[14] M. Rastegar-Mojarad, R. D. Boyce, and R. Prasad, "UWM-TRIADS: Classifying drug-drug interactions with two-stage SVM and post-processing," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics, 7th Int. Workshop Semantic Eval.*, Atlanta, GA, USA, 2013, pp. 667–674.

[15] Y. Tsuruoka. (2006). *Genia Tagger: Part-of-Speech Tagging, Shallow Parsing, and Named Entity Recognition for Biomedical Text*. [Online]. Available: www-tsujii.is.su-tokyo.ac.jp/GENIA/tagger

[16] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[17] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. (2013). "Distributed representations of words and phrases and their compositionality." [Online]. Available: https://arxiv.org/abs/1310.4546

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

[20] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: Historical perspective and recent advances," *J. Amer. Med. Inform. Assoc.*, vol. 17, no. 3, pp. 229–236, 2010.

[21] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://keras.io

[22] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[23] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics, 7th Int. Workshop Semantic Eval. (SEM/SemEval)*, vol. 2, 2013, pp. 341–350.

[24] C. Nédellec *et al.*, "Overview of BioNLP shared task 2013," in *Proc. BioNLP Shared Task Workshop*, 2013, pp. 1–7.

[25] A. Y. Ng, "Feature selection, $L1$ vs. $L2$ regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 78.

[26] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[27] 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. 2nd Joint Conf. Lexical Comput. Semantics, 7th Int. Workshop Semantic Eval.*, Atlanta, GA, USA, 2013, pp. 651–659.

[28] P. Thomas, M. Neves, T. Rocktäschel, and U. Leser, "WBI-DDI: Drug-drug interaction extraction using majority voting," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics, 7th Int. Workshop Semantic Eval.*, Atlanta, GA, USA, 2013, pp. 628–635.

BO XU received the B.Sc. and Ph.D. degrees from the Dalian University of Technology, China, in 2007 and 2014, respectively. She is currently a Lecturer with the School of Software, Dalian University of Technology. Her current research interests include social computing, data mining, information retrieval, and natural language processing.

XIUFENG SHI received the B.S. degree in software engineering from the North University of China, Taiyuan, China, in 2016. He is currently pursuing the master's degree with the School of Software, Dalian University of Technology, China. His research interests include text mining for biomedical literatures and information extraction from huge biomedical resources.

ZHEHUAN ZHAO received the Ph.D. degree from the Dalian University of Technology, China, in 2017. He is currently a Lecturer with the Software College, Dalian University of Technology. His research interest includes text mining for biomedical literatures, information extraction from huge biomedical resources, and knowledge graph construction.

WEI ZHENG received the B.Sc. degree in computer science and technology from the Dalian University of Technology, Dalian, China, in 2003, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. Her research interests include text mining and machine learning.