BMC Bioinformatics

RESEARCH ARTICLE Open Access

Relation extraction between bacteria and biotopes from biomedical texts with attention mechanisms and domain-specific contextual representations

Amarin Jettakul, Duangdao Wichadakul and Peerapon Vateekul*

Abstract

Background: The Bacteria Biotope (BB) task is a biomedical relation extraction (RE) that aims to study the interaction between bacteria and their locations. This task is considered to pertain to fundamental knowledge in applied microbiology. Some previous investigations conducted the study by applying feature-based models; others have presented deep-learning-based models such as convolutional and recurrent neural networks used with the shortest dependency paths (SDPs). Although SDPs contain valuable and concise information, some parts of crucial information that is required to define bacterial location relationships are often neglected. Moreover, the traditional word-embedding used in previous studies may suffer from word ambiguation across linguistic contexts.

Results: Here, we present a deep learning model for biomedical RE. The model incorporates feature combinations of SDPs and full sentences with various attention mechanisms. We also used pre-trained contextual representations based on domain-specific vocabularies. To assess the model's robustness, we introduced a mean F1 score on many models using different random seeds. The experiments were conducted on the standard BB corpus in BioNLP-ST'16. Our experimental results revealed that the model performed better (in terms of both maximum and average F1 scores; 60.77% and 57.63%, respectively) compared with other existing models.

Conclusions: We demonstrated that our proposed contributions to this task can be used to extract rich lexical, syntactic, and semantic features that effectively boost the model's performance. Moreover, we analyzed the trade-off between precision and recall to choose the proper cut-off to use in real-world applications.

Keywords: Biomedical text mining, Relation extraction, Deep learning, Attention networks, Contextual word-embeddings, Domain-specific language

Background

Due to the rapid development of computational and biological technology, the biomedical literature is expanding at an exponential rate [1]. This situation leads to difficulty manually extracting the required information. In BioNLP-ST 2016, the Bacteria Biotope (BB) task [2] followed the general outline and goals of previous tasks defined in 2011 [3] and 2013 [4]. This task aims to investigate the interactions of bacteria and its biotope; habitats or geographical entity, from genetic, phylogenetic, and ecology perspectives. It involves the *Lives_in* relation, which is a mandatory relation between related arguments, the bacteria and the location where they live. Information pertaining to the habitats where bacteria live is particularly critical in applied microbiology fields such as food safety, health sciences, and waste processing [2–4]. An example relation between bacteria and their location in this task is shown in Fig. 1.

In recent years, significant efforts have focused on challenging BB tasks. Several studies have been proposed that incorporate feature-based models. TEES [5], which adopted support vector machine (SVM) with a variety of

*Correspondence: peerapon.v@chula.ac.th
Chulalongkorn University Big Data Analytics and IoT Center (CUBIC), Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

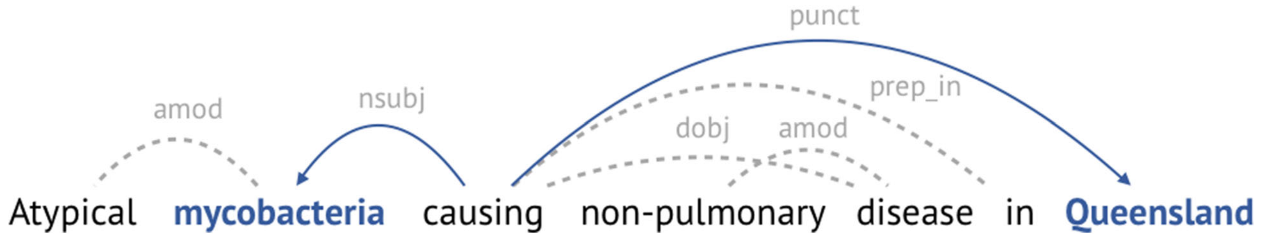

amod
nsubj
punct
prep_in
dobj
amod
Atypical mycobacteria causing non-pulmonary disease in Queensland

Fig. 1 Example of the BB relation in a BB task. Bacteria "mycobacteria" and location "Queensland" are shown in blue, bold text. The dependencies are represented by arrows; SDPs are indicated in blue

features based on shortest dependency paths (SDPs), was the best-performing system with an F1 score of 42.27% in the BioNLP-ST'13 [4]. The VERSE team [6], which placed first in BioNLP-ST'16 with an F1 score of 55.80%, utilized SVM with rich features and a minimum spanning dependency tree (MST). Feature-based models, however, heavily depend on feature engineering, which sometimes is limited by its lack of domain-specific knowledge [7].

Since 2014, deep learning (DL) methods have garnered increasing attention due to their state-of-the-art performance in several natural language processing (NLP) tasks [8]. Unlike the feature-based models, DL models demand less feature engineering because they can automatically learn useful features from training data. Examples of popular DL models that have successfully been applied for biomedical relation extraction include Convolutional Neural Networks (CNNs) [9–12] and Recurrent Neural Networks (RNNs) [13, 14].

Other than feature-based models in the BB task, several former studies using DL approaches have significantly outperformed traditional SVM approaches. For example, in BioNLP-ST'16, DUTIR [15] utilized CNN models to achieve an F1 score of 47.80%; TurkuNLP [16] used multiple long short-term memories (LSTM) with SDPs to achieve an F1 score of 52.10% and was ranked second in the competition. DET-BLSTM [17] applied bidirectional LSTM (BLSTM) with a dynamic extended tree (DET) adapted from SDPs and achieved an F1 score of 57.14%. Recently, BGRU-Attn [18] proposed bidirectional gated recurrent unit (BGRU) with attention mechanism and domain-oriented distributed word representation. Consequently, it became the state-of-the-art DL system without hand-designed features for the BB task with an F1 score of 57.42%.

Despite the success of DL in the past studies, there are still several limitations to be considered. Although SDPs have been shown to contain valuable syntactic features for relation extraction [16–21], they still may miss some important information. For example, in Fig. 1, the word "in", which should play a key role in defining the relation between the bacteria "mycobacteria" and the biotope "Queensland" is not included in SDP (represented by blue lines) because there is no dependency path between "in" and any entities. To overcome the limitation of SDPs, some studies have used sequences of full sentences to extract biomedical relations from texts [22–24]. However, it is very difficult for DL models to learn enough features from only sequences of sentences. Instead of learning from full sentences, attention networks have demonstrated success in a wide range of NLP tasks [25–31]. In addition, BGRU-Attn [18] first used the Additive attention mechanism [29] for the BB task to focus on only sections of the output from RNN instead of the entire outputs and achieved state-of-the-art performance. Other attention techniques such as Entity-Oriented attention [30] and Multi-Head attention [31] still have not been explored for this task. From the aspect of word representation, traditional word-embeddings [32, 33] only allow for single context-independent representation. This situation can lead to word sense ambiguation across various linguistic contexts [34]. Contextual representations of words [35] and sentences [36] based on language-understanding models addressed this problem and achieved state-of-the-art performance on general-purpose domain NLP tasks [35–39]. Nevertheless, [40] has shown that the word-embedding models pre-trained on a general-purpose corpus such as Wikipedia are not suitable for biomedical-domain tasks. Finally, the training process of DL approaches with many randomly initialized parameters is non-deterministic—multiple executions of the same model may not result in the same outcome. To solve this issue and provide a statistical comparison of models' performances, [41, 42] reported the mean F1 score of the same model architecture initialized with different parameter settings (random seeds). This evaluation metric indicates the average behavior of a model's performance and is more suitable for the biases and trends in real-world applications [43]. However, the mean F1 score had never been explored in prior studies of the BB task.

In this study, we propose a hybrid model between an RNN and a feed-forward neural network such as a CNN. We use the RNN to extract full-sentence features from long and complicated sentences. We also apply the CNN to capture SDP features that are shorter, more valuable, and more concise. In addition, because attention mechanisms have been proven to be helpful in the BB task [18], we incorporate several kinds of attention mechanisms—Additive attention, Entity-Oriented attention, and Multi-Head attention—into the model. Furthermore, we integrate domain-specific contextual word representation into the model to provide word-sense disambiguation. Sentence representation was also introduced to improve the full-sentence model by embedding sequence sentence information from a pre-trained language understanding model. To address the uncertainty of a single run model's performance measured by the maximum F1 score, we used the mean F1 score as an evaluation metric for comparisons of the models.

Results

We assessed the performance of our model as follows. First, we compared our model with existing models in terms of maximum and average F1 scores. Then, we evaluated the effectiveness of each contribution used by the model: feature combination between full sentences and SDP, attention mechanisms, contextual word representation, and contextual sentence representation. Here, we discuss the overall experimental results of this proposed model.

Performace comparisons with existing models

Maximum f1 score comparisons

Table 1 lists the maximum F1 score of our model compared with those of prior studies. In the BB task [2], each team evaluated the model on the test set using an online evaluation service. Most of the existing systems were based either on SVM or DL models. The SVM-based baseline [5] was a pipeline framework using SVMs on SDPs with an F1 score of 42.27%. Similarly, [6] proposed a utilized SVM with rich feature selection that yielded an F1 score of 55.80%. Compared with SVM-based models, DL-based models automatically learn feature representations from sentences and achieve state-of-the-art performance. For example, DUTIR [15] utilized a multiple-filter-widths CNN to achieve an F1 score of 47.80%. TurkuNLP [16] employed a combination of several LSTMs on the shortest dependency graphs to obtain the highest precision of 62.30% and an F1 score of 52.10%. BGRU-Attn [18] proposed a bidirectional GRU with the attention mechanism and biomedical-domain-oriented word-embedding to achieve the highest recall of 69.82% and an F1 score of 57.42%. These results reveal that our proposed model achieved the best performance in the official evaluation (i.e., the highest F1 score: 60.77%). In contrast with the previous state-of-the-art model (BGRU-Attn [18]), our model achieved more balanced precision (56.85%) and recall (65.28%). The results revealed that our model could leverage both full-sentence and SDP models along with contextual representations to capture the vital lexical and syntactic features of given sentences. Therefore, our model can combine the advantages of all contributions to achieve a good trade-off between precision and recall, which resulted in its superior performance in the BB corpus.

Mean f1 score comparisons

In this section, we compared our overall model's performance with other existing models in terms of mean F1 score. However, the source codes or the executables for all previous models except VERSE [6] were not available. In these experiments, we reimplemented two DL models: TurkuNLP [16] as a baseline for the DL model and BGRU-Attn [18] as a current state-of-the-art model. More details of the reimplementation are provided in the Additional file 1. Table 2 lists the results of our model compared with these reimplemented DL models based on mean F1 scores. For TurkuNLP [16], every hyper-parameter was strict with those provided in the original paper. We can achieve the reimplemented maximum F1 score of 51.99% compared with 52.10% that reported in the original paper and mean F1 score of 46.18%. For BGRU-Attn [18], we employed the model architecture and features based on the original paper, including domain-oriented word representations and dynamic extended trees (DET). However, the original paper did not provide some parameters of the model, such as the number of GRU's hidden dimensions, we empirically chose the best hyper-parameters by cross-validation. After several attempts, our reimplemented BGRU-Attn model achieved the maximum F1 score of 55.54% compared with 57.42% as provided in the original paper with the mean F1 score of 50.22%. In Table 2, our model achieved the highest mean F1 score of 57.63% and the lowest SD of 1.15. This finding indicates that our model is more robust to randomness and highly consistent in its performance. To provide a statistically significant comparison of our model's performance,

Table 1 Performance comparison on maximum F1 score with existing models

Methods	Team	F	R	P
SVM-based	TEES [5]	42.27	38.35	61.61
	VERSE [6]	55.80	61.50	51.00
	DUTIR [15]	47.80	39.70	60.00
	TurkuNLP [16]	52.10	44.80	**62.30**
DL-based	DET-BLSTM [17]	57.14	57.99	56.32
	BGRU-Attn [18]	57.42	**69.82**	48.76
	Our model	**60.77**	65.28	56.85

The listed results derive from the corresponding papers. F: F1 score; R: recall; P: precision. Our model used all of the proposed contributions (the results from the last row in Table 6). The highest scores are highlighted in bold

Table 2 Performance comparison on mean F1 score with existing models

Model	F1 score			
	Mean	SD	Min	Max
TurkuNLP [16]	46.18	4.64	35.07	51.99 (52.10)
BGRU-Attn [18]	50.22	2.79	43.05	55.54 (57.42)
Our model	**57.63**	**1.15**	**54.41**	**60.77**

These results derive from the model reimplementation of the existing model. Our model used all of the proposed contributions (the results from the last row in Table 6). The highest scores are highlighted in bold except for the SD. The F1 scores in parentheses represent expected maximum scores that provided by the original papers

we also performed a two-sample t-test with the hypothesis that two populations (our model and a compared model) were equal in terms of their mean F1 scores (null hypothesis H_0). The results revealed that we rejected the null hypothesis with a p-value less than 0.001 (or more than 99.9% confidence). This fact implied that our model's mean F1 score was significantly better than that of other models.

Effects analysis of each proposed strategy

In the following sections, we evaluate the effectiveness of each contribution of our proposed model: combined full-sentence and SDP models, attention mechanisms, contextual word representation, and contextual sentence representation (Tables 3, 4, 5 and 6). To overcome the variant problem in model evaluation, each experiment used the mean F1 score for model selection and evaluation.

Influence of full-sentence and sDP features

Table 3 lists the mean F1 score of 30 DL models with different random seeds. The mean F1 score obtained from the experiment indicated that the use of full-sentence and SDP models together outperformed the separated models. The data in Table 3 also demonstrate that CNN achieved better performances than BLSTM when BLSTM and CNN were separately applied to the full sentences and SDPs, respectively. This result suggests that our model effectively combines the SDP and full-sentence models to extract more valuable lexical and syntactic features. These features were generated not only from two different sequences (full sentences and SDPs) but also two different neural network structures (BLSTM and CNN).

Table 3 The effectiveness of the application of full-sentence and SDP features according to the mean F1 scores of 30 different random seeds

Model		F1 score			
Fulls	SDPs	Mean	SD	Min	Max
-	CNN	43.79	3.39	37.02	51.82
BLSTM	-	41.22	14.49	12.82	49.93
BLSTM	**CNN**	**45.96**	**2.87**	**42.09**	**52.19**

All of the highest scores are highlighted in bold except for the SD

Table 4 The effectiveness of the integrated attention mechanisms according to mean F1 scores for 30 different random seeds

Model		PE	F1 score			
Fulls	SDPs		Mean	SD	Min	Max
BLSTM	CNN	✗	45.96	2.87	42.09	52.19
BLSTM	CNN	✓	48.49	4.76	38.75	55.40
BLSTM-Attn	CNN	✓	49.02	3.62	42.03	56.51
BLSTM-EAttn	CNN	✓	50.24	3.72	43.14	55.72
BLSTM-EAttn	**MAttn**	**✓**	**53.42**	**2.51**	**46.67**	**56.70**

All of the highest scores are highlighted in bold except for the SD. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 3). Note: "PE" denotes positional encoding, "Attn" denotes the use of only Additive attention, "EAttn" denotes the use of both Additive and Entity-Oriented attentions, and "MAttn" denotes the use of Multi-Head attention

Influence of attention mechanisms

After we measured the effectiveness of the full-sentence and SDP features, we additionally explored the effects of the Additive, Entity-Oriented, and Multi-Head attention mechanisms. The attention mechanisms were applied to concentrate the most relevant input representation instead of focusing on entire sentences. Table 4 lists the productiveness of each attention mechanism integrated into our full-sentence and SDP models. According to [31], Multi-Head attention networks were first proposed with the use of PE to insert valuable locality information. Because Multi-Head attention networks were employed with PE, we applied PE to CNN in order to fairly compare the effectiveness of Multi-Head attention. The use of the Additive attention mechanism improved the mean F1 score by 0.53%. Entity-Oriented attention improved

Table 5 The effectiveness of domain-specific contextual word representation according to the mean F1 scores of 30 different random seeds

Pre-trained word model	F1 score			
	Mean	SD	Min	Max
PubMed word2vec	53.42	2.51	46.67	56.70
general-purpose ELMo	54.30	3.61	42.76	56.51
random-PubMed ELMo	53.81	3.65	38.89	57.01
specific-PubMed ELMo	**55.91**	**1.49**	**51.24**	**57.48**

All of the highest scores are highlighted in bold except for the SD. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 4). Note: "PubMed word2vec" denotes the context-free word model, "general-purpose ELMo" denotes the general-purpose contextual word model, "random-PubMed ELMo" denotes the domain-general contextual word model based on 118 million randomly selected tokens abstracts from PubMed, and "specific-PubMed ELMo" denotes the domain-specific contextual word model based on 118 million bacterial-relevant abstracts from PubMed

Table 6 The effectiveness of the contextual sentence representation by the mean F1 scores of 30 different random seeds

Sentence representation	F1 score			
	Mean	SD	Min	Max
without	55.91	1.49	51.24	57.48
with	**57.63**	**1.15**	**54.41**	**60.77**

All of the highest scores are highlighted in bold except for the SD. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 5)

the average F1 score from 49.02 to 50.24%. These results show that attention mechanisms might highlight influential words for the annotated relations and help reveal semantic relationships between each entity. This approach improved the overall performance of our model. Finally, the stacks of Multi-Head attention networks were the primary contributor to our model. The experimental results revealed that the proposed model using Multi-Head attention together with SDPs increased the mean F1 score by 3.18% compared with the proposed model using CNN. Our proposed model used stacks of Multi-Head attentions with residual connections instead of CNN.

Influence of domain-specific contextual word representation

Table 5 lists the effectiveness of our domain-specific, contextual word representation to our model after previous contributions (combined features and attention mechanisms). The contextual word representation (ELMo) was proposed to provide word sense disambiguation across various linguistic contexts and handle out-of-vocabulary (OOV) words using a character-based approach. The results in Table 5 reveal that every ELMo model outperformed the traditional word2vec model. One possible explanation for this finding is that the ELMo model uses a character-based method to handle OOV words while word2vec initializes these OOV word representations randomly. The ELMo model can also efficiently encode different types of syntactic and semantic information about words in context and therefore improve overall performance. The use of our proposed contextual word model with a domain-specific corpus (specific-PubMed ELMo) achieved the highest average F1 score of 55.91%. This score represented an improvement by 2.49%, 1.61%, and 2.10% compared with the score deriving from the use of PubMed word2vec, general-purpose ELMo, and random-PubMed ELMo, respectively. These improvements reveal the importance of taking relevant information into account when training contextual embedding vectors. We also noted that the general-purpose ELMo achieved slightly better performance compared with the random-PubMed ELMo. However, the latter was pre-trained on a biomedical-domain corpus; the size of the pre-trained corpus of the former (5.5 billion tokens) is significantly larger than that of the latter (118 million tokens), which resulted in the higher-quality word-embeddings and better semantic representations.

Influence of contextual sentence representation

In order to use sentence embeddings as fixed features from the pre-trained BERT, [36] suggested that the best-performing method involved concatenating the feature representations from the top four 768-dimensional BLSTM hidden layers of the pre-trained model. However, we found that it was better to sum up the last four 768-dimensional hidden layers into the 768-dimension sentence embedding. This situation may have been due to the small training dataset. The addition of contextual sentence representation from the fine-tuned BERT model improved the mean F1 score by 1.68% (Table 6). The results suggest that the fine-tuned BERT model could enhance the full-sentence model to encode crucial contextual representations of long and complicated sentences.

Discussion

Our proposed model can take advantage of the proposed contributions in order to construct rich syntactic and semantic feature representations. Our model significantly outperforms other existing models in terms of both mean F1 score (57.63%; SD = 1.15%) and maximum F1 score (60.77%). The mechanisms that largely support stable performance include the Multi-Head attentions and domain-specific contextual word representation, which are responsible for mean F1 score increases of 3.18% and 2.49%, respectively. A possible advantage of Multi-Head attention compared with CNN is the ability to determine the most relevant local feature representations from multiple subspaces to the BB task based on attention weights. In addition, domain-specific contextual word representation is beneficial to the proposed model for capturing contextual embeddings from a bacterial-relevant corpus. The box-and-whisker plot in Fig. 2 shows the mean F1 score distribution of the existing DL models and our final proposed model (blue boxes). The boxplot illustrates the performance of our model after incrementally adding each of the main contributions (grey boxes). The mean F1 score of each model is shown as a line. The blue boxes indicate the comparison of our final model and two reimplemented TurkuNLP [16] and BGRU-Attn [18]. The mean F1 score of our model was 57.63%, which exceeds that of the TurkuNLP and BGRU-Attn models by 11.45% and 7.41%, respectively. In other words, our proposed model generally achieves better performance in terms of both mean and maximum F1 scores. Furthermore, the inter-quartile range of our proposed model is much smaller than that of other DL models. This finding demonstrates that the performance of

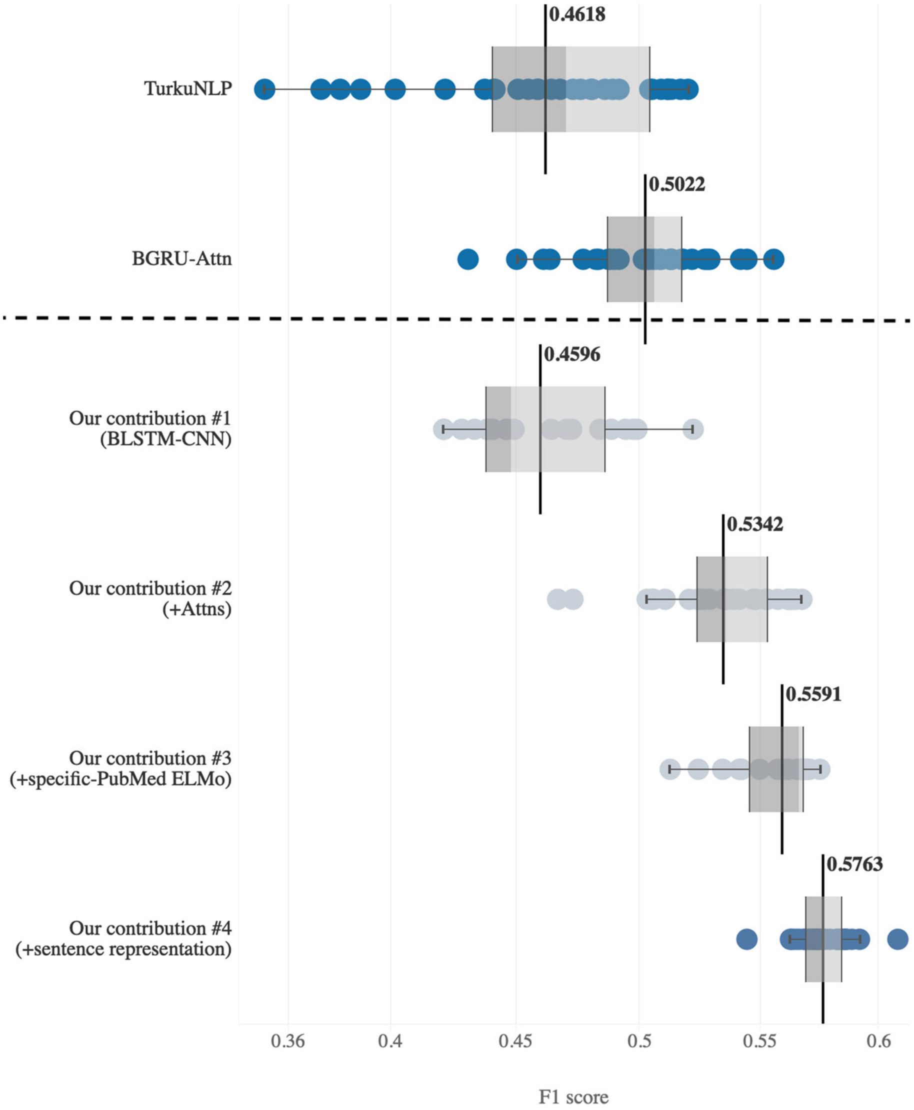


Fig. 2 Box-and-whisker plot of average F1 score distributions of the deep-learning-based relation extraction models on the BB task. The comparison between our model and existing deep-learning-based models is shown in blue; the improvement of our model after adding each of the proposed contributions is shown in grey. Note: "Attns" denotes the use of integrated attention mechanisms

our model is more robust and suitable for real-world applications.

For binary classification problems, F1 score is a common metric for evaluating an overall model's performance because it conveys both precision and recall into one coherent metric. In some applications, however, it is more important to correctly classify instances than to obtain highly convergent results (i.e., high precision). On the other hand, some other applications place more emphasis on convergence rather than correctness (high recall). We experimented with using a frequency cut-off to explore how the probabilities output by the model function as a trade-off between precision and recall. Figure 3 shows the precision-recall curve (PRC) of our proposed model. When applied to real-world scenarios, users of the model are responsible for choosing the right cut-off value for their applications. For example, in semi-automated text-mining applications for knowledge management researchers never want to miss any bacteria-biotope relations. As a result, models with a high recall will be chosen to prescreen these relations. On the other hand, automated text-mining applications for decision support systems will require more precise relations. In Fig. 3, our model with the default (0.5) cut-off value achieved an F1 score of 60.77% with balanced 56.85% recall and 65.28% precision. With a cut-off of 0.025, our model achieved the highest recall at 70.54% with 50.11% precision and an F1 score of 58.59%. With

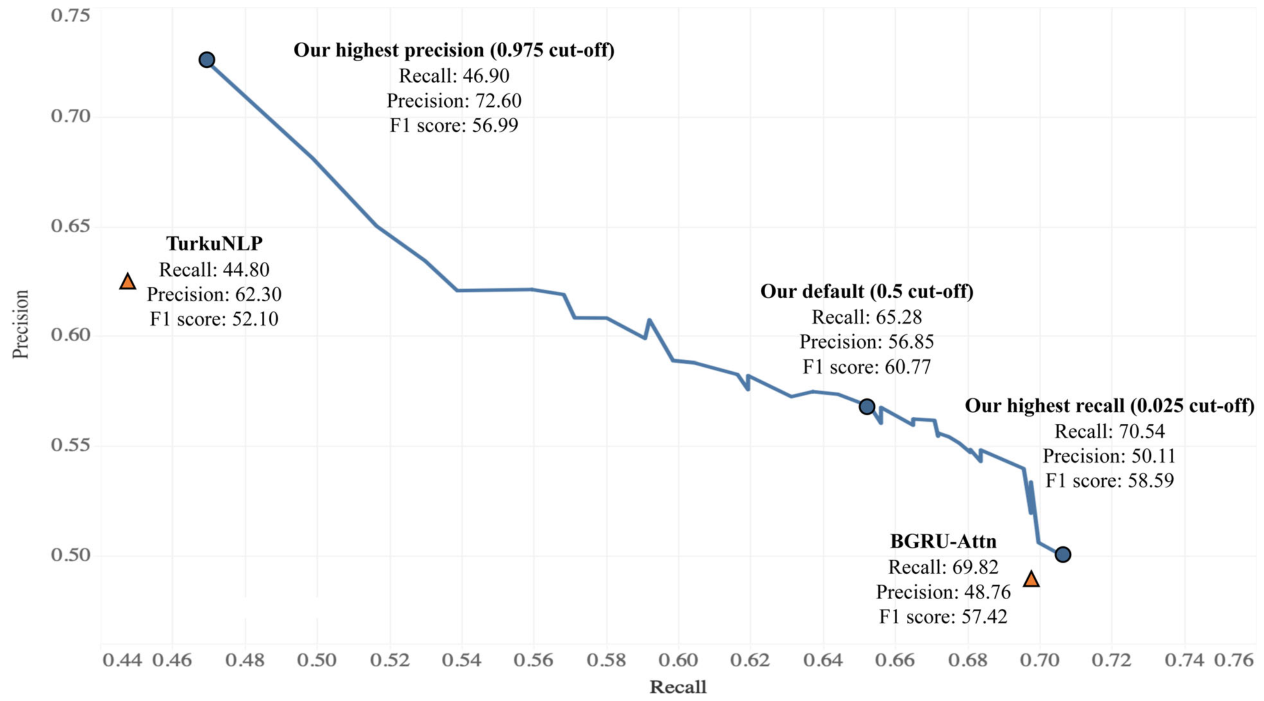


Fig. 3 The precision-recall curve for our proposed model showing the trade-off between the true positive rate and the positive predictive value for our model using different probability thresholds (cut-off values)

this cut-off value, our model outperformed the existing highest-recall model (BGRU-Attn [18]) by both 0.72% recall and 1.35% precision. Similarly, the line plot shown in Fig. 3 shows that our model with a 0.975 cut-off achieved the highest precision (72.60%), recall (46.90%) and F1 score (56.99%). This model also outperformed the existing highest-precision model (TurkuNLP [16]) by 10.30% in precision and 2.10% in recall.

To determine the factors that adversely affected the performance of our proposed model, we manually analyzed the correct and incorrect predictions from a development set compared with other existing models. We found that the proposed model could detect true negatives (TNs) better than other reimplemented models. This finding arose mainly because full-sentence features boosted the model's ability to predict an entity pair as a false relation. For example, the sentence "*Rickettsia felis* was the only **entity_1** found infecting fleas, whereas *Rickettsia bellii* was the only agent infecting ticks, but no animal or human **entity_2** was **shown** to contain rickettsial DNA.", where SDP are shown in bold, was predicted to be a false relation by our model. Other models predicted this sentence to be a true relation because of the word "shown" in the SDP. In addition, we found that false positives (FPs) were generally caused by the complex and coordinate structures of full sentences. A complicated sentence and a long distance between two entities can lead to relation classification failures. Examples of these adverse effects include the sentences "The 210 isolates with typical LPS patterns (119 Ara- clinical, 13 Ara- soil, 70 **entity_1 entity_2**, and 8 reference National Type Culture Collection strains) also exhibited similar immunoblot profiles against pooled sera from patients with melioidosis and hyperimmune mouse sera." and "Testing animal and human sera by indirect immunofluorescence assay against four rickettsia antigens (*R. rickettsii*, *R. parkeri*, *R. felis*, and *R. bellii*), some opossum, **entity_2**, horse, and human sera reacted to **entity_1** with titers at least four-fold higher than to the other three rickettsial antigens." In each of these sentences, the SDPs are highlighted in bold.

Limitations of our model

One of the most important limitations of our model is that it cannot extract inter-sentence relations between the bacteria and the biotopes. Hence, all true inter-sentence relations become false negatives. Inter-sentence relation extraction is much more challenging because it requires a more nuanced understanding of language to classify relations between entities in different sentences and clauses characterized by complex syntax [44–46]. Because the size of our BB dataset is quite small, it is very difficult for DL models to learn sufficient high-quality features for the target tasks. However, this challenging task is left for future work. Furthermore, there is a large repertoire of biomedical literature and domain resources that are freely accessible and can be used as unlabeled data for semi-supervised learning and transfer learning methods [47–49].

Application to other tasks

Since our proposed model automatically learns the features from the context of any two entities, this model architecture can be applied to other biomedical RE tasks, such as DDI extraction task. In this section, to show the model's generalization to other tasks, we evaluated our proposed model to the DDIExtraction 2013 corpus [50]. Unlike BB task [2], DDI extraction is a multi-class relation extraction task. The DDI dataset contains four DDI types: Advice, Mechanism, Effect, and Int. The detailed statistics of the DDI dataset are listed in Table 7.

To apply our proposed model to the DDI corpus, there are three steps to adjust from the proposed model to the BB corpus. First, for the pre-training corpus of contextual word representations (specific-PubMed ELMo), the word "drug" was used as a keyword, instead of the bacteria mention. Second, the DDI corpus was used to fine-tune the pre-trained contextual sentence model (BERT), instead of the BB corpus. Third, the best hyper-parameters for the DDI task were chosen using 5-fold cross-validation on the training and development data.

Table 8 lists the maximum F score (micro) of our proposed model compared with other previous models for the DDI corpus. Similar to the BB corpus, most of the existing models were based on either SVM or DL approaches. The experimental results revealed that our proposed model could achieve the highest overall F score of 80.3% and the highest recall of 83.0%. These results show that our model can combine the advantages of every contribution to achieve the highest F score in the leaderboard of both BB and DDI tasks.

Table 8 Performance comparison (maximum F score) with existing models on the DDI corpus

Methods	Team	F	R	P
SVM-based	Uturku [51]	59.4	49.9	73.2
	FBK-irst [52]	65.1	65.6	64.6
	RAIHANI [53]	71.1	68.7	73.7
DL-based	Liu-CNN [9]	69.8	64.7	75.7
	MCCNN [54]	70.2	65.3	76.0
	TEES-CNN [55]	73.5	67.6	**80.5**
	Joint AB-LSTM [13]	71.5	65.0	74.5
	Char-RNNs [42]	72.1	65.9	80.0
	Hierarchy RNN [14]	72.9	71.8	74.1
	Recursive NN [56]	73.5	69.6	77.8
	Attn-BLSTM [30]	77.3	76.8	78.4
	Our model	**80.3**	**83.0**	77.6

The listed results derive from the corresponding papers. F: F score (micro); R: recall; P: precision. The highest scores are highlighted in bold

Conclusions

We have presented a DL extraction model for the BB task based on a combination of full-sentence and SDP models that integrate various attention mechanisms. Furthermore, we introduced a pre-trained, contextual, word-embedding model based on the large bacteria-relevant corpus and fine-tuned contextual sentence representation. These embeddings encouraged the model to effectively learn high-quality feature representations from pre-trained language modeling. We evaluated our proposed model based on maximum and mean F1 scores. The experimental results demonstrated that our model effectively integrated these proposed contributions. The results showed that we could improve the performance of relation extraction to achieve the highest maximum and average F1 scores (60.77% and 57.63%, respectively). Our proposed model significantly outperformed other state-of-the-art models. Additionally, our model is more robust to real-world applications than the previous RE models. Furthermore, our model can achieve the best performance in the DDI task which can ensure the model's generalization to other tasks and strengthen our proposed contributions.

Despite our model exhibiting the best performance on the BB task, some challenges remain. In particular, inter-sentence relations between bacteria and location entities have not been taken into account by any existing deep-learning-based models; this situation is likely due to insufficient training data. In the future, we plan to develop a new approach to increase the quantity and quality of limited training data for the target task using transfer learning and semi-supervised learning methods.

Table 7 Statistics of a DDI dataset

	DDI type	Train	Dev	Test
Positive instances	Mechanism	1054	263	301
	Effect	1349	336	359
	Advice	660	164	220
	Int	139	37	95
Negative instances	False	19015	4753	4363
Total instances		22227	5553	5338

Methods

In this section, we describe the proposed DL model for extracting BB relations from the biomedical literature (Fig. 4).

Text preprocessing

We used the TEES system [5, 16] to run the pipeline of the text preprocessing steps. Tokenization and part-of-speech (POS) tagging for each word in a sentence were generated using the BLLIP parser [57] with the biomedical-domain model. The dependency grammar resulted from the BLLIP was further processed using the Stanford conversion tool [58] to obtain the Stanford dependencies (SD) graph.

We then used Dijkstra's algorithm to determine the SDPs between each pair of entities: bacteria and biotope. The SDPs represented the most relevant information and diminished noises by undirected graph (Fig. 1). An entity pair was neglected if there was no SDP between the entities. While the dependency paths only connect a single word to others within the same sentence (intra-sentence), there are some cross-sentence (inter-sentence) associations that can be very challenging in terms of the extraction task. In order to compare with other existing works [5, 15–18], only intra-sentence relations were considered.

To ensure the generalization of the models, we followed the protocol of previous studies [17, 18] that blinded the entities in a sentence. The bacteria and location mentions were replaced by "entity_1" and "entity_2" respectively. For example, as shown in Table 9, we can generate two BB relation candidates (termed "instances") from a sentence

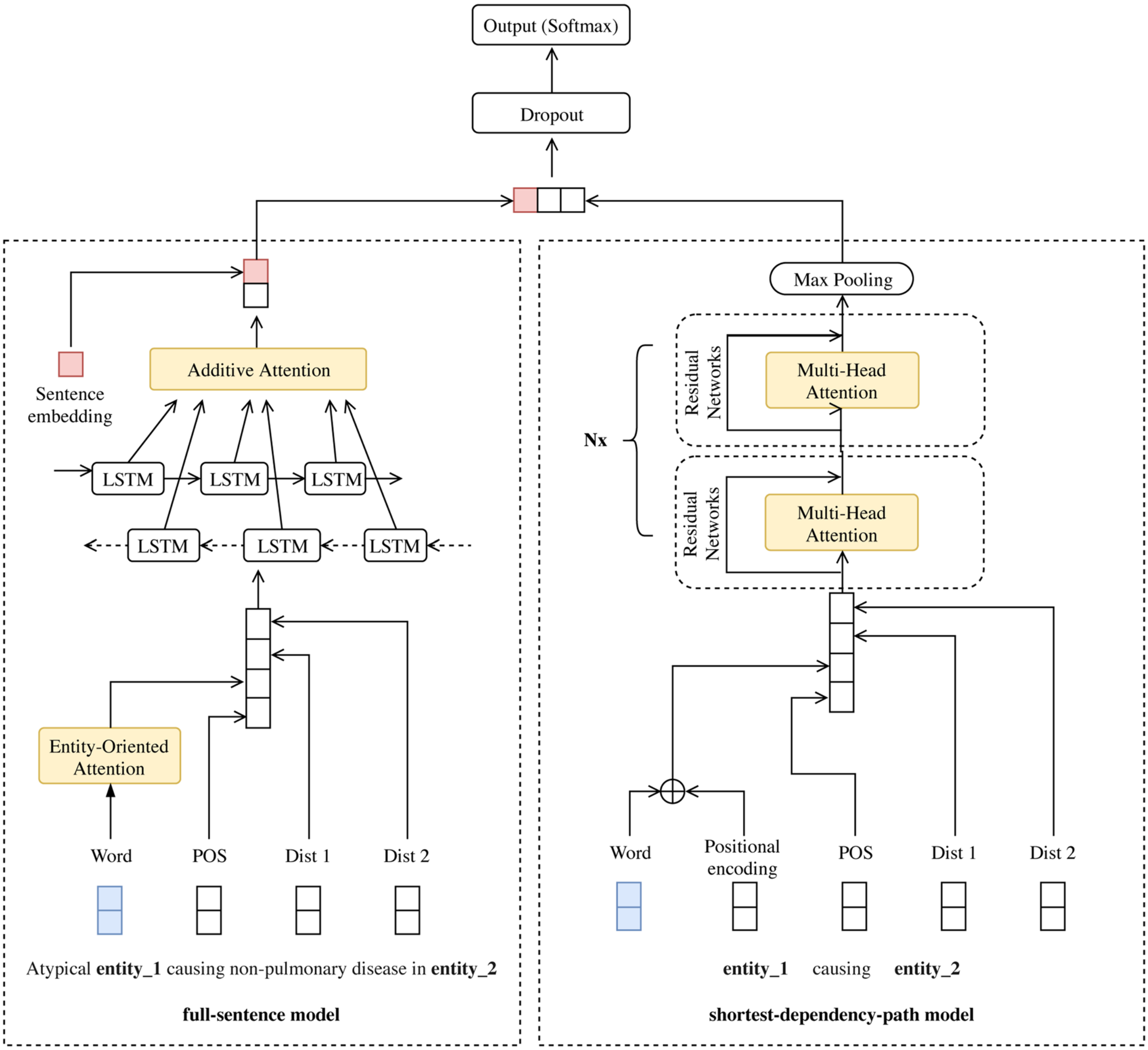


Fig. 4 The overall architecture of our proposed model with the combined full-sentence and SDP models, together with various attention mechanisms

Table 9 Bacteria-biotope relation candidates (instances) in a sentence after entity blinding

Entity pair	Bacteria-biotope relation candidate after entity binding
(***Helicobacter pylori***, *gastric*)	Long-term ***entity_1*** infection and the development of atrophic gastritis and *entity_2* cancer in Japan.
(***Helicobacter pylori***, *Japan*)	Long-term ***entity_1*** infection and the development of atrophic gastritis and gastric cancer in *entity_2*.

The bacteria and location mentions are highlighted in bold italics and italics, respectively

“Long-term ***Helicobacter pylori*** infection and the development of atrophic *gastritis* and gastric cancer in *Japan*.”, where the bacteria and location mentions are highlighted in bold italics and italics, respectively. After entity blinding, we converted all words to lowercase to simplify the searching process and improve text matching.

Input embedding representations

The input representations used in our model were divided into full-sentence and SDP features. Let $\{w_1, w_2, \ldots, w_m\}$ and $\{s_1, s_2, \ldots, s_n\}$ denote the full sentence and SDPs of a sentence that are represented by different embeddings. Each word w_i in a full sentence was represented by word vector, POS, and distance embeddings. Each word s_j in the SDP was represented by word vector, POS, and distance embeddings together with positional encoding (PE). The detailed embeddings used in our model are explained below.

For a full sentence in the RNN model, **word-embedding** was a 200-dimensional word vector, the pre-trained biomedical word-embedding model [59], built from a combination of PubMed and PMC texts using Word2Vec [32]. **Part-of-speech embedding** was initialized randomly at the beginning of the training phase.

Distance embedding [18, 60] is derived from the relative distances of the current word to the bacteria and location mentions. For example, in Fig. 1, the relative distances of the word “in” to bacteria “mycobacteria” and location “Queensland” are −4 and 1, respectively. To construct the distance embedding $D(l)$ for each relative distance, every dimension $d(l)$ of the distance embedding is initialized as in Eq. 1, where l is the relative distance and s refers to the maximum of the relative distances in the dataset. All $d(l)$ dimensions form the distance vectors $[dist_1, dist_2]$, which represent the distance embeddings $D(l)$ of the current word to the bacteria and location mentions, respectively.

$$d(l) = \tanh\left(\frac{l}{s}\right) \tag{1}$$

For SDP in the CNN model, we used **PE** [31] to inject some information about the absolute position of the words in the sentence. The PE vectors were initialized by sine and cosine functions of different frequencies; these functions embed information based on their relative position. Because PE has the same dimension as the word-embedding, we can sum these two vectors.

In summary, the overall input embedding representation for a word w_i in full sentences is $z_i = [w_i^{word}; w_i^{pos}; w_i^{dist_1}; w_i^{dist_2}]$. Similarly, for a given word s_j on the SDP the overall input embedding representation is $z_i = [w_i^{word} + w_i^{PE}; w_i^{pos}; w_i^{dist_1}; w_i^{dist_2}]$.

A dL model based on full sentences and sDPs

Full-sentence model

We employed BLSTM [61] to learn global features from full sentences. The BLSTM can be used to encode the sequential inputs both forward and backward, and it has been shown to outperform one-way LSTM in many studies [13, 60–63]. Given a full sentence of M tokens, $\{z_1, z_2, \ldots, z_M\}$, at the t-th time step, the BLSTM takes the current input representation (z_i), previous hidden state (h_{t-1}), and previous memory cell (c_{t-1}) as its inputs to generate the current hidden state (h_i) and memory cell (c_i). For BLSTM, the forward LSTM output (h_k^f) and the backward LSTM output (h_k^b) are concatenated into $h_k = h_k^f; h_k^b$.

SDP model

The multiple-filter-widths CNN model [64] was proposed for the SDP model to learn local features from SDPs. For a given SDP sequence of N tokens, $\{z_1, z_2, \ldots, z_N\}$, let $z_i \in \Re^k$ be the k-dimensional input embedding vector corresponding to the i-th word in the sequence. The CNN takes an input sequence of length N to generate the feature map (c_i) by convolutional filters and max pooling operations. Compared with LSTM, the CNN model is expected to be better at extracting high-quality features from short and concise SDPs [65].

Attention mechanisms

Attention mechanisms are motivated by how human pays visual attention to different words in a sentence. The main idea of attention mechanism is to assign attention score (alignment score), which can be either trainable [29, 31] or non-trainable parameters [66]. Each of these attention mechanisms has recently been successfully applied to biomedical relation extraction tasks [14, 18, 30]. In this work, we proposed to use a combination of three attention mechanisms—Additive for extracting sentence-level features, Entity-Oriented for extracting word-level features, and Multi-Head for extracting local features from SDPs—because each attention was proposed to focus on the different information levels. Figure 4 shows how these attention mechanisms are integrated into our proposed DL model.

Additive attention

The Additive attention focuses on sentence-level information. It was first used by [29] to improve neural machine translation and recently applied to the BB task [18]. The idea of Additive attention is to consider all LSTM hidden states with different attention weights when deriving the context vector. The context vector depends on the sequence of hidden states $\{h_1, h_2, \ldots, h_K\}$. Each hidden state contains information about the whole input sequence with a strong focus on the parts surrounding the i-th word. The context vector (c) was computed as a weighted sum of these hidden states (h_i) as in Eq. 2. The attention weight (a_i) of each hidden state (h_j) was then computed as in Eq. 3. The additive attention assigned a score (a_i) to the pair of input at position i, which was parameterized using a feed-forward network with a single hidden layer. The model was then jointly trained with other parts of the model. The attention score function is shown in Eq. 4, where v_a is the weight matrix to be learned.

$$c = \sum_{i=1}^{k} a_i h_i \tag{2}$$

$$a_i = \frac{\exp(score(h_i))}{\sum_{j=1}^{K} \exp(score(h_j))}, \quad \text{for i = 1, ..., K} \tag{3}$$

$$score(h_i) = v_a^{\top} \tanh(h_i) \tag{4}$$

Entity-Oriented attention

Based on the state-of-the-art relation extraction for Drug-Drug Interaction (DDI) task [30], Entity-Oriented attention mechanism can determine which words in the specific sentences are the most influential for the relationship between a pair of entities using a similarity score. To focus on word-level information, this attention mechanism was applied after our word-embedding layer to quantify the concentration of word-level information. Figure 5 exhibits an example of how Entity-Oriented attention weights were computed. We exploited two similarity scores (S_i^1, S_i^2) which were computed by inner product similarity of the current word-embedding vector (w_i) and the j-th entity word-embedding (e_j), $j \in \{1, 2\}$ as in Eq. 5. Then, both similarity scores were averaged and scaled by the square root of word-embedding dimensionality (m) as in Eq. 6. The attention weight (a_i) for each word was computed by a softmax function over the similarity scores of all M words in a sentence as in Eq. 7.

$$S_i^j = (w_i \cdot e_j), \quad j \in \{1, 2\} \tag{5}$$

$$S_i = \frac{S_i^1 + S_i^2}{2\sqrt{m}} \tag{6}$$

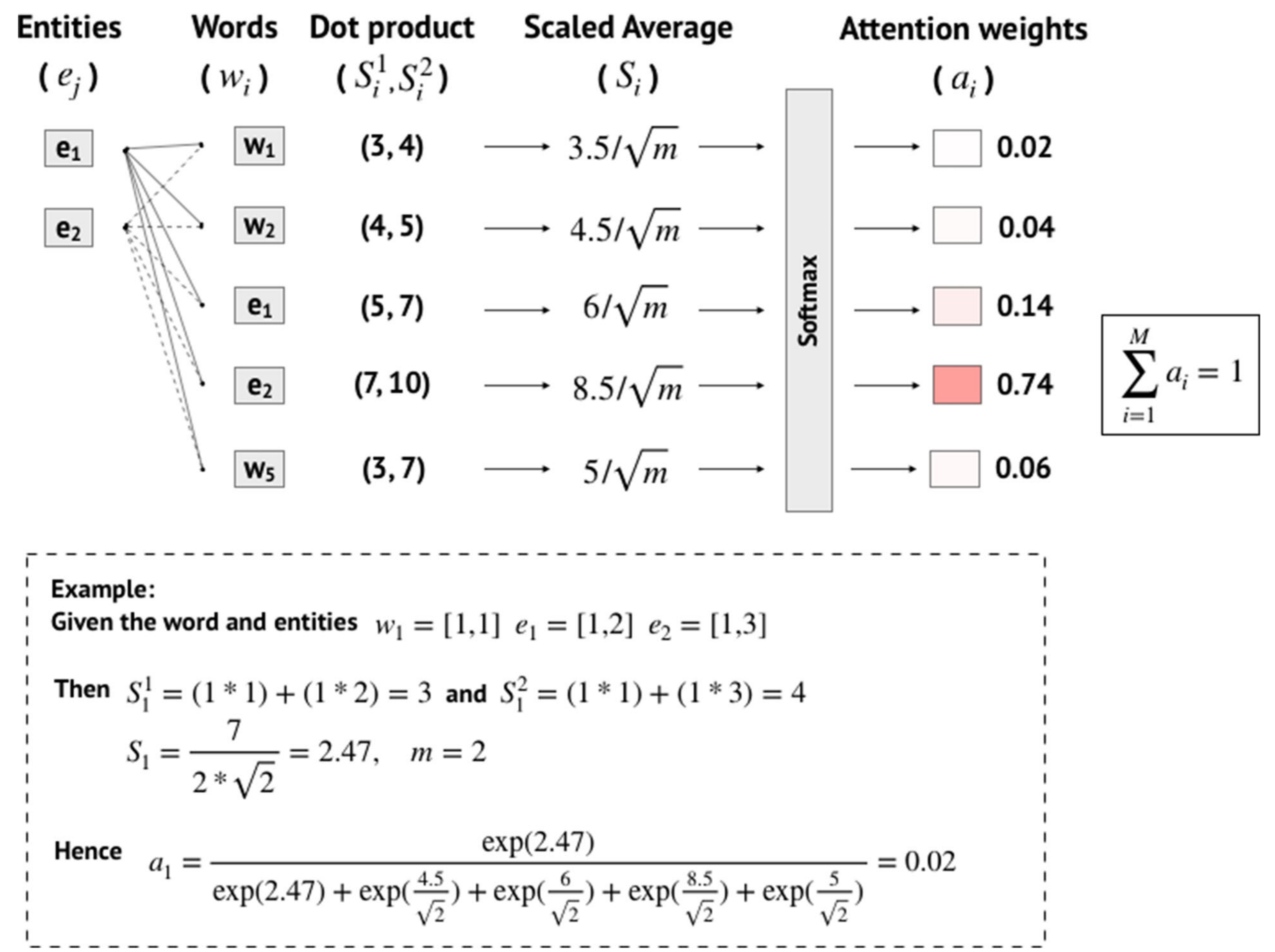
$$a_i = \frac{\exp(S_i)}{\sum_{j=1}^{M} \exp(S_j)}, \quad \text{for i = 1, \ldots, M} \tag{7}$$

Multi-Head attention

Multi-Head attention was used as the major component in Transformer model [31] for the encoder-decoder networks. The attention mechanism in Transformer model was interpreted as a way of computing the relevance of a set of values (context vector representations) based on some keys and queries. The encoder part of the model used word-embeddings in a sentence for its keys, values, and queries. The decoder part, in contrast, used the word-embeddings in a sentence for its queries and the encoder's outputs for its keys and values. Similar to [67], we employed the Multi-Head attention as the encoder to generate attention-based representation from SDP embeddings. Self-attention used in the Multi-Head attention is a mechanism to compute a representation for each word in SDP. This attention relates different positions of a single sentence to compute a representation of each word in a sentence. The self-attention purpose is to combine the interpretation of other relevant words into the current word representation.

The Multi-Head attention used multiple attention-weighted sums instead of a single attention. Figure 6 shows how we computed the Multi-Head attention features of three attention heads (h_1, h_2, h_3) based on three Scaled Dot-Product attentions, similar to [31]. For each head, we applied different learnable weights (W_q, W_k, and W_v) to the same SDP embedding (z_i) of length N to obtain query (q_i), key (k_i), and value (v_i) as in Eq. 8. More generally, these vectors (q_i, k_i, and v_i) represented the SDP in different vector spaces. In Eq. 9, the attention score was calculated based on the key and query, then scaled by the square root of word-embedding dimensionality (m). The attention weight (a_i) was computed by applying a softmax function to its corresponding attention score as in Eq. 10. The context vector (c_i) was generated by applying an element-wise multiplication of the attention weight with the value as in Eq. 11. In order to obtain each attention head feature (h_i), the context vector from each word in SDP of length N was concatenated as in Eq. 12.

A number of the attention heads exhibit behaviors that seem related to the sentence structure. The empirical results of the former study [68] showed that the Multi-Head attention worked more efficiently than the usual Single-Head attention in the context of relation extraction. Figure 7 represents how we generated two different context vectors from two attention heads based on self-attention mechanism. Each attention head can learn to encode SDP features by detecting different orders of individual words in the sentence. Hence, each attention



Example:
Given the word and entities $w_1 = [1,1]$ $e_1 = [1,2]$ $e_2 = [1,3]$

Then $S_1^1 = (1*1) + (1*2) = 3$ **and** $S_1^2 = (1*1) + (1*3) = 4$

$$S_1 = \frac{7}{2*\sqrt{2}} = 2.47, \quad m = 2$$

Hence $$a_1 = \frac{\exp(2.47)}{\exp(2.47) + \exp(\frac{4.5}{\sqrt{2}}) + \exp(\frac{6}{\sqrt{2}}) + \exp(\frac{8.5}{\sqrt{2}}) + \exp(\frac{5}{\sqrt{2}})} = 0.02$$

Fig. 5 Illustration of Entity-Oriented attention mechanism to normalize full-sentence embeddings by similarity-based attention weights

head produced the different context vector based on its self-attention weights. Similar to Transformer model, we employed a stack of Multi-Head attentions with residual connections and positional encodings, as shown in Fig. 4.

$$(q_i, k_i, v_i) = (z_i W_q^T, z_i W_k^T, z_i W_v^T) \tag{8}$$

$$score(h_i) = \frac{q_i \cdot k_i}{\sqrt{m}} \tag{9}$$

$$a_i = \frac{\exp(score(h_i))}{\sum_{j=1}^{N} \exp(score(h_j))}, \quad \text{for i = 1, ..., N} \tag{10}$$

$$c_i = \sum_{i=1}^{N} v_i a_i \tag{11}$$

$$h_i = [c_1; c_2; ...; c_N] \tag{12}$$

Contextual representations

The choice of how to represent words or sentences poses a fundamental challenge for NLP communities. There have been some advances in universal pre-trained contextual representations on a large corpus that can be plugged into a variety of NLP tasks to automatically improve their performance [35, 36]. By incorporating some contextualized information, these representations have been shown in [35–39] to alleviate the problem of ambiguation and outperform traditional context-free models [32, 33]. In this study, we propose two contextual embedding models pre-trained on a biomedical corpus of words and sentences.

Contextual word representation

The contextual word vector used in our proposed model was generated by ELMo [35]. ELMo learned word representations from the internal states of a bidirectional language model. It was shown to improve the state-of-the-art models for several challenging NLP tasks. Context-free models such as Skip-gram [32] and GloVe [33] generate a single word representation for each word in their vocabulary. For instance, the word "cold" would have the same representation in "common cold" and "cold sensation" [34]. On the other hand, contextual models will generate a representation of the word "cold" differently based on context. This representation can be easily added to our

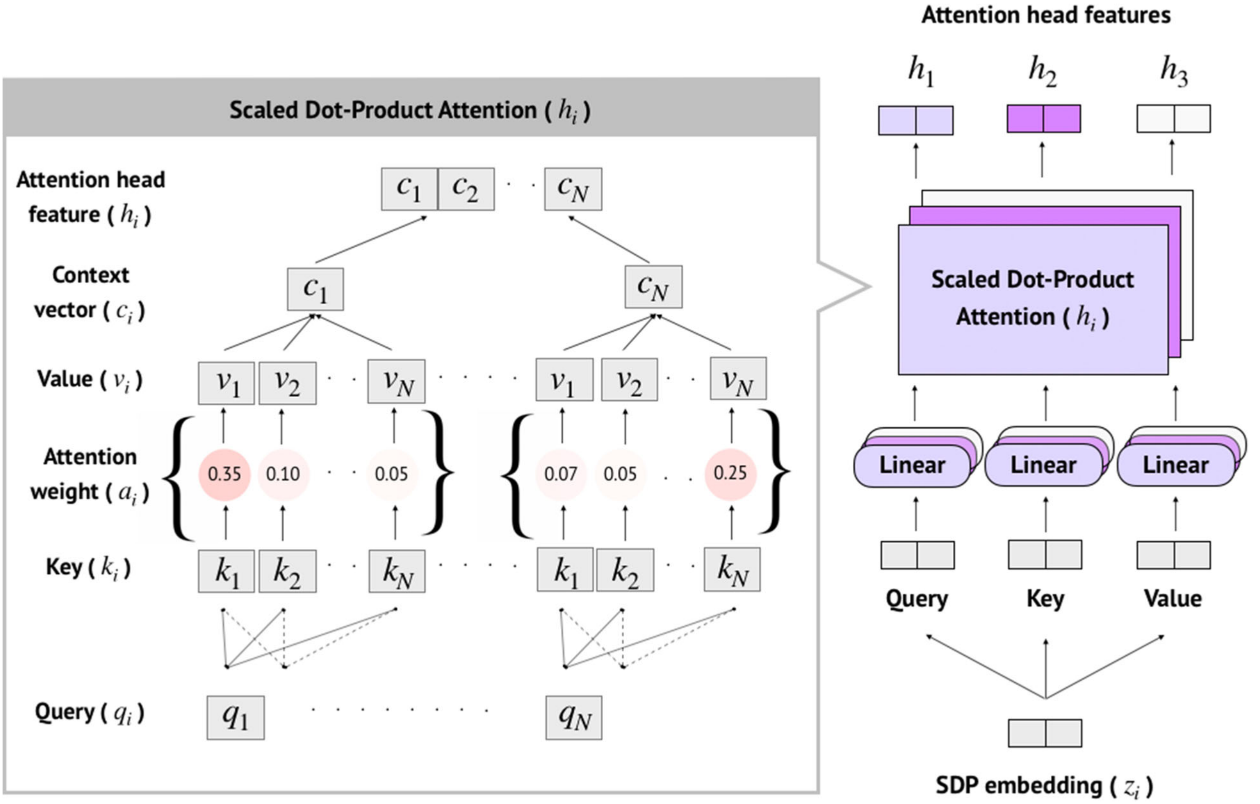


Fig. 6 Illustration of Multi-Head attention mechanism to encode SDP embeddings, which consists of three Scaled Dot-Product attentions running in parallel

proposed model by reconstituting the 200-dimensional word vectors with the new pre-trained contextual word vectors. Currently, the ELMo model, pre-trained on a large general-purpose corpus (5.5 billion tokens), is freely available to use [35]. However, [40, 69] showed that domain-irrelevant word-embedding models pre-trained on large, general-purpose collections of texts are not sufficient for biomedical-domain tasks. Therefore, we present a domain-specific, contextual, word-embedding model pre-trained on a bacterial-relevant corpus. Inspired by the relevance-based word-embedding [70], the corpus to pre-train our proposed contextual word-embedding model included relevance-based abstracts downloaded from PubMed, which contain only sentences with bacterial scientific names from the BB task (118 million tokens). To evaluate the effectiveness of our proposed domain-specific, contextual, word-embedding model, we compared it with the contextual model pre-trained on randomly selected abstracts from PubMed with the same number of tokens. All of the pre-trained models were fine-tuned with the BB dataset in order to transfer learned features from the pre-train models to our task.

Contextual sentence representation

Our contextual sentence embedding was constructed by BERT [36]. BERT represents words based on a bidirectional approach and learns relationships between sentences. Hence, BERT representation unambiguously represents both words and sentences. However, due to the limited computational resource to pre-train BERT using our biomedical corpus the available pre-trained BERT on general-purpose corpus was adopted and fine-tuned with the BB task.

Training and classification

The output layer used the softmax function [71] to classify the relationship between pairs of bacteria and biotope mentions. The softmax layer takes the output of BLSTM for full-sentence feature, the output of Multi-Head attention networks for SDP feature, and the sentence embedding from BERT as its inputs (Fig. 4). These inputs are fed into a fully connected neural network. The softmax layer's output was the categorical probability distribution over each class type (c) as in Eq. 13.

$$p(c|s) = softmax(W_0 \cdot s + b_0) \tag{13}$$

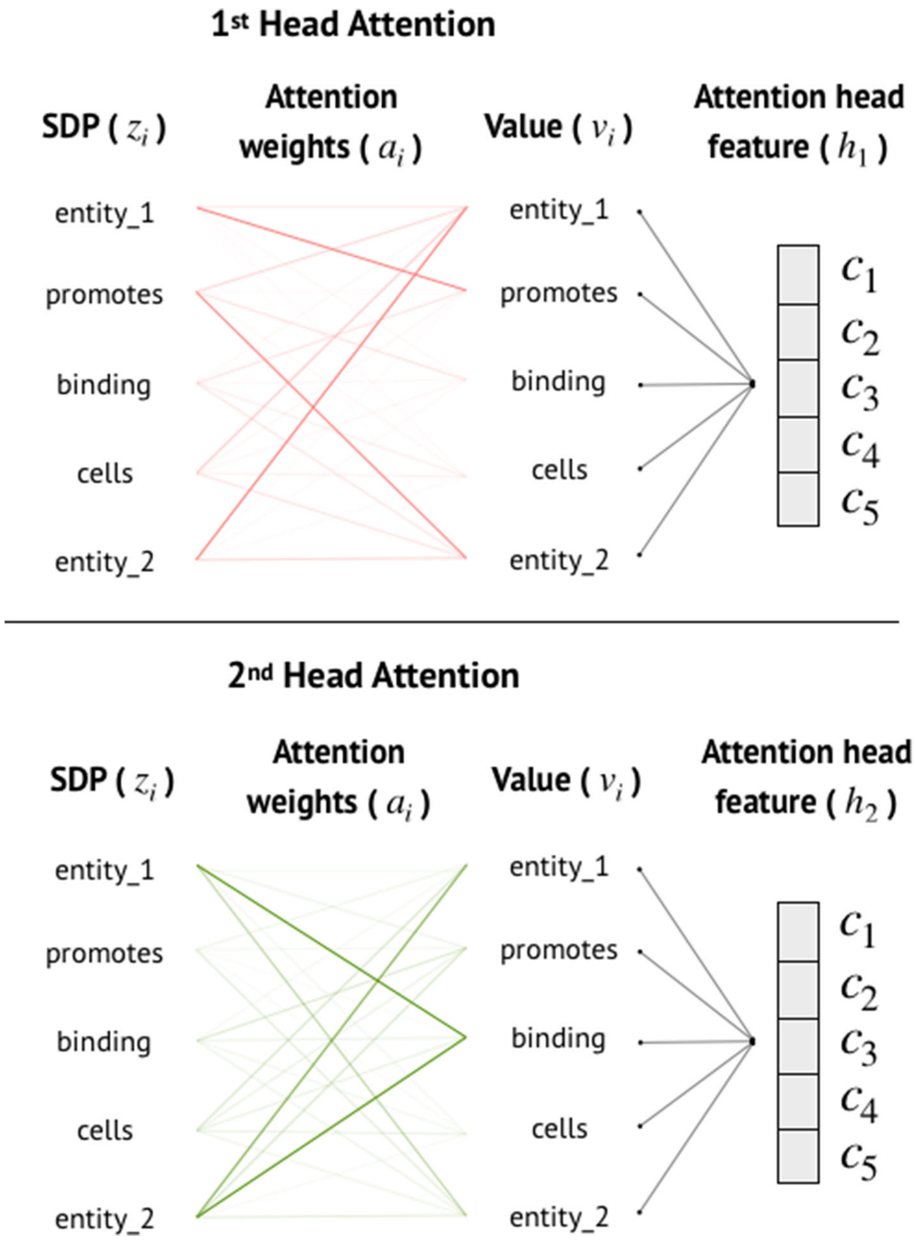


Fig. 7 An example of how each one of two attention heads in Multi-Head attention computes different context vectors based on words in SDP. The width of a line refers to an attention weight

where W_0 and b_0 are weight parameters and s is the feature representation of sentences. For the binary classification, we used the cross-entropy cost function ($J(\theta)$) as the training objective as in Eq. 14.

$$J(\theta) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{14}$$

where y is the binary indicator (0 or 1) if the class label is correct for each predicted sentence and p is the predicted probability. Additionally, we applied Adam optimization to update the network weights with respect to the cost function.

Data

Training and test datasets

The dataset provided by the BB task [2] of BioNLP-ST'16 consists of titles and abstracts from PubMed with respect to reference knowledge sources (NCBI taxonomy and OntoBiotope ontology). All entity mentions—*Bacteria*, *Habitat*, and *Geographical*—and their interactions were manually annotated from diverse-backgrounds annotators. Each bacteria-biotope pair was annotated as either a negative or positive *Lives_in* relation. The relations can be defined as inter-sentence and intra-sentence.

In our study, we also followed previous studies [5, 15–18] in simply excluding inter-sentence instances from the dataset. This procedure resulted in the removal of 107 and 64 annotated instances from the training data and development data, respectively. Table 10 lists the statistics of the preprocessed BB dataset used in our experiments.

The pre-training corpus of contextual word representations

In order to get the proposed domain-specific word-embeddings (specific-PubMed ELMo), we pre-trained ELMo on the bacterial-relevant abstracts downloaded from the PubMed database. These specific abstracts contain roughly 118 million words that use all of the bacteria names that are noted in the BB dataset as keywords. An example keyword is the bacteria mention “mycobacteria” (Fig. 1). Furthermore, we pre-trained another domain-general word-embeddings (random-PubMed ELMo) on randomly selected PubMed abstracts with a similar corpus size to evaluate the performance of the domain-specific model. To reduce the memory requirement of both pre-training models, we only used the words in the training, development, and test sets to construct the vocabularies.

Hyper-parameter setting

We used the Pytorch library [72] to implement the model and empirically tuned the hyper-parameters using 3-fold cross-validation on the training and development data. After tuning, the dimensions of the contextual word-embedding (ELMo), context-free word-embedding, POS embedding, distance embedding, and sentence embedding (BERT) were 400, 200, 100, 300, and 768, respectively. The dimension of PE was set to either 200 or 400 for the context-free or contextual word-embeddings, respectively. The hidden unit number of BLSTM and the filter number of CNN were 64. The convolutional window sizes were 3, 5, and 7. For the Multi-Head attention mechanism, we used three stacks of Multi-Head attentions with respect to the residual connections; the number of heads for each stack was 2. Before the output layer, we applied a dropout rate of 0.5 to the concatenation of full-sentence, SDP, and sentence-embedding features. The mini-batch was set to 4, and a rectified linear unit (ReLU) was used as our activation functions. We set the learning rate to 0.001 for Adam optimization with early stopping based on the development data. As a result, the epoch number varied depending on this early stopping. From our experiments, we found that the optimal epoch number would be in a range between 3 and 5. To avoid model convergence issue, we used different parameters for the model with only full-sentence features, denoted as “full-sentence” in the “Influence of full-sentence and sDP features” section. The dropout rate was set to 0.1, and the hidden unit number of LSTM was 32.

Table 10 Statistics of a preprocessed BB dataset

	Train	Dev	Test
Avg.word in SDPs	4.81	4.86	4.58
Avg.word in full sentences	27.22	29.92	28.12
Positive instances	248	173	-
Negative instances	275	332	-
Total instances	523	505	532

Evaluation metrics

For our model, the final results on the test dataset were evaluated using the online evaluation service provided by the BB task of the BioNLP-ST’16 [2]. Due to the removal of inter-sentence examples, any inter-sentence relations in the test dataset that counted against our submission were considered to be false negatives.

As discussed above, different parameter initializations (or random seeds) can affect the model’s performance, an evaluation of a single model several times tends to result in performance convergence. To alleviate this problem, we reported the mean F1 score instead of only the maximum F1 score reported by previous studies [5, 6, 15–18]. To calculate the mean F1 score, we built 30 models as suggested by [41]. These models were trained using the same architecture but with different random seeds. Then, we evaluated the F1 score of each model on the same test set using an online evaluation service. With these F1 scores, we then calculated the minimum, maximum, mean, and standard deviation (SD) to assess the robustness of the model. In this study, we used the mean F1 score as the main evaluation metric; the maximum F1 score was still used to compare with other previously used models.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s12859-019-3217-3.

Additional file 1: Reimplementation details contain hyper-parameters and training setting for reimplemented TurkuNLP and BGRU-Attn models.

Abbreviations

BB: Bacteria Biotope; BERT: Bidirectional Encoder Representations from Transformers; BGRU: Bidirectional gated recurrent unit; BioNLP-ST: BioNLP Shared Task; BLSTM: Bidirectional long short-term memory; CNNs: Convolutional neural networks; DDI: Drug-drug interaction; DDI: Drug-Drug Interactions; DL: Deep learning; ELMo: Embeddings from Language Models; MST: Minimum spanning dependency tree; NLP: Natural language processing; OOV: Out of vocabulary; PE: Positional encoding; POS: Part of speech; PRC: Precision-Recall curve; RE: Relation extraction; RNNs: Recurrent neural networks; SD: Stanford dependencies; SDPs: Shortest dependency paths; SVMs: Support vector machines

Acknowledgments

AJ thanks the scholarship from Chula Computer Engineering Graduate Scholarship for CP Alumni. All authors greatly thank Chulalongkorn University

Systems Biology Center (CUSB) and NVAITC AI Technology Center (NVAITC) at Chulalongkorn University for constructive comments. Also, we appreciate the editors and anonymous referees for their careful reading of the manuscript and valuable and constructive comments.

Authors' contributions

AJ was the major contributor in coding, performing experiments, and writing the manuscript. PV and DW gave guidance and support during the design and analysis processes. All authors have read and approved the final version of this manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets of BB task and DDI task can be downloaded at http://2016.bionlp-st.org/tasks/bb2 and https://www.cs.york.ac.uk/semeval-2013/task9, respectively. The 200-dimensional pre-trained biomedical word2vec model can be downloaded at http://bio.nlplab.org. The codes for our proposed model and the reimplemented models are available from the corresponding author on request during the peer review process. We also plan to publish the codes on GitHub in the near future.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 28 April 2019 Accepted: 12 November 2019
Published online: 03 December 2019

References

1. Cohen KB, Hunter L. Getting started in text mining. PLoS Comput Biol. 2008;4(1):0001–3. https://doi.org/10.1371/journal.pcbi.0040020.
2. Deléger L, Bossy R, Chaix E, Ba M, Ferré A, BessiÈres P, Nédellec C. Overview of the Bacteria Biotope Task at BioNLP Shared Task 2016; 2016. p. 12–22.
3. Bossy R, Jourde J, Bessières P, Van De Guchte M, Nédellec C. Bionlp shared task 2011: bacteria biotope; 2011. p. 56–64.
4. Bossy R, Golik W, Ratkovic Z, Bessières P, Nédellec C. BioNLP shared Task 2013–An Overview of the Bacteria Biotope Task; 2013. p. 153–60.
5. Björne J, Salakoski T. TEES 2.1: Automated Annotation Scheme Learning in the BioNLP 2013 Shared Task. BioNLP Shared Task 2013 Workshop. 2013;2013:16–25.
6. Lever J, Jones SJM. VERSE : Event and Relation Extraction in the BioNLP 2016 Shared Task; 2016. p. 42–9.
7. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553): 436–44. https://doi.org/10.1038/nature14539.
8. Kumar S. A Survey of Deep Learning Methods for Relation Extraction. 2017. https://doi.org/10.1016/j.jpowsour.2006.03.077.
9. Liu S, Tang B, Chen Q, Wang X. Drug-Drug Interaction Extraction via Convolutional Neural Networks. Comput Math Methods Med. 2016;2016:. https://doi.org/10.1155/2016/6918381.
10. Zhao Z, Yang Z, Lin H, Wang J, Gao S. A protein-protein interaction extraction approach based on deep neural network. Int J Data Min Bioinforma. 2016;15(2):145. https://doi.org/10.1504/IJDMB.2016.076534.
11. Zeng D, Liu K, Lai S, Zhou G, Zhao J. Relation Classification via Convolutional Deep Neural Network. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING'14); 2014. http://aclweb.org/anthology/C/C14/C14-1220.pdf.
12. Quan C, Hua L, Sun X, Bai W. Multichannel convolutional neural network for biological relation extraction. BioMed Res Int. 2016. https://doi.org/10.1155/2016/1850404.
13. Sahu SK, Anand A. Drug-drug interaction extraction from biomedical texts using long short-term memory network. J Biomed Inform. 2018;86(August):15–24. https://doi.org/10.1016/j.jbi.2018.08.005.
14. Zhang Y, Zheng W, Lin H, Wang J, Yang Z, Dumontier M. Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. Bioinformatics. 2018;34(5):828–35. https://doi.org/10.1093/bioinformatics/btx659.
15. Li H, Zhang J, Wang J, Lin H, Yang Z. DUTIR in BioNLP-ST 2016 : Utilizing Convolutional Network and Distributed Representation to Extract Complicate Relations. 201693–100.
16. Mehryary F, Björne J, Pyysalo S, Salakoski T, Ginter F. Deep Learning with Minimal Training Data: TurkuNLP Entry in the BioNLP Shared Task 2016; 2016. p. 73.
17. Li L, Zheng J, Wan J, Huang D, Lin X. Biomedical event extraction via Long Short Term Memory networks along dynamic extended tree. In: Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016; 2017. p. 739–42. https://doi.org/10.1109/BIBM.2016.7822612.
18. Li L, Wan J, Zheng J, Wang J. Biomedical event extraction based on GRU integrating attention mechanism. BMC Bioinformatics. 2018. https://doi.org/10.1186/s12859-018-2275-2.
19. Liu Y, Wei F, Li S, Ji H, Zhou M, Wang H. A Dependency-Based Neural Network for Relation Classification. 2015. https://doi.org/10.3115/v1/P15-2047.
20. Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. 2016. https://doi.org/10.18653/v1/P16-1105.
21. Yan X, Mou L, Li G, Chen Y, Peng H, Jin Z. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Path. 2015.
22. Zheng S, Hao Y, Lu D, Bao H, Xu J, Hao H, Xu B. Joint entity and relation extraction based on a hybrid neural network. Neurocomputing. 2017;257: 59–66. https://doi.org/10.1016/j.neucom.2016.12.075.
23. Vu NT, Adel H, Gupta P, Schütze H. Combining Recurrent and Convolutional Neural Networks for Relation Classification. 2016. https://doi.org/10.18653/v1/N16-1065.
24. Zhang Y, Lin H, Yang Z, Wang J, Zhang S, Sun Y, Yang L. A hybrid model based on neural networks for biomedical relation extraction. J Biomed Inform. 2018;81(March):83–92. https://doi.org/10.1016/j.jbi.2018.03.011.
25. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); 2016. https://doi.org/10.18653/v1/P16-2034.
26. Zhang Y, Er MJ, Venkatesan R, Wang N, Pratama M. Sentiment classification using Comprehensive Attention Recurrent models. In: Proceedings of the International Joint Conference on Neural Networks; 2016. https://doi.org/10.1109/IJCNN.2016.7727384.
27. Zhao Z, Wu Y. Attention-based convolutional neural networks for sentence classification. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH; 2016. https://doi.org/10.21437/Interspeech.2016-354.
28. Long Y, Qin L, Xiang R, Li M, Huang C-R. A Cognition Based Attention Model for Sentiment Analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; 2017. https://doi.org/10.18653/v1/D17-1048.
29. Luong M-T, Pham H, Manning CD. Effective Approaches to Attention-based Neural Machine Translation. 2015. https://doi.org/10.18653/v1/D15-1166.
30. Zheng W, Lin H, Luo L, Zhao Z, Li Z, Zhang Y, Yang Z, Wang J. An attention-based effective neural model for drug-drug interactions extraction. BMC Bioinformatics. 2017;18(1):1–11. https://doi.org/10.1186/s12859-017-1855-x.
31. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention Is All You Need. (Nips). 2017. https://doi.org/10.1017/S0140525X16001837.
32. Mikolov T, Chen K, Corrado G, Dean J. Distributed Respresentation of Words and Pharses and their Compositionality. CrossRef Listing of Deleted DOIs. 2000. https://doi.org/10.1162/jmlr.2003.3.4-5.951.
33. Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation; 2014. p. 1532–43. https://doi.org/10.3115/v1/D14-1162.
34. Stevenson M, Guo Y. Disambiguation in the biomedical domain: The role of ambiguity type. J Biomed Inform. 2010;43(6):972–81. https://doi.org/10.1016/j.jbi.2010.08.009.

35. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations. 2018. https://doi.org/10.18653/v1/N18-1202.
36. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018. https://arxiv.org/abs/1810.04805.
37. Zhu C, Zeng M, Huang X. Sdnet: Contextualized attention-based deep network for conversational question answering. CoRR. 2018;abs/1812.03593:.
38. Radford A. Improving language understanding by generative pre-training; 2018.
39. Salant S, Berant J. Contextualized word representations for reading comprehension. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 554–9. https://doi.org/10.18653/v1/N18-2088. https://www.aclweb.org/anthology/N18-2088.
40. Wang Y, Liu S, Afzal N, Rastegar-Mojarad M, Wang L, Shen F, Kingsbury P, Liu H. A comparison of word embeddings for the biomedical natural language processing. J Biomed Inform. 2018;87:12–20. https://doi.org/10.1016/j.jbi.2018.09.008.
41. Colas C, Sigaud O, Oudeyer P-Y. How Many Random Seeds? Stat Power Anal Deep Reinforcement Learn Exp. 20181–20.
42. Kavuluru R, Rios A, Tran T. Extracting drug-drug interactions with word and character-level recurrent neural networks. In: 2017 IEEE International Conference on Healthcare Informatics (ICHI); 2017. p. 5–12. https://doi.org/10.1109/ICHI.2017.15.
43. Huang CC, Lu Z. Community challenges in biomedical text mining over 10 years: Success, failure and the future. Brief Bioinform. 2016;17(1): 132–44. https://doi.org/10.1093/bib/bbv024.
44. Singhal A, Simmons M, Lu Z. Text Mining Genotype-Phenotype Relationships from Biomedical Literature for Database Curation and Precision Medicine. PLoS Comput Biol. 2016;12(11):1–19. https://doi.org/10.1371/journal.pcbi.1005017.
45. Singhal A, Simmons M, Lu Z. Text mining for precision medicine: Automating disease-mutation relationship extraction from biomedical literature. J Am Med Inform Assoc. 2016;23(4):766–72. https://doi.org/10.1093/jamia/ocw041.
46. Tsafnat G, Lin F, Choong MK. Translational Biomedical Informatics. Encycl Syst Biol. 2013;January 2015:2275–8.
47. Ruder S. Neural transfer learning for natural language processing, PhD thesis. Galway: National University of Ireland; 2019.
48. Peters M, Neumann M, Zettlemoyer L, Yih W-t. Dissecting contextual word embeddings: Architecture and representation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics; 2018. p. 1499–509. https://www.aclweb.org/anthology/D18-1179.
49. Clark K, Luong M-T, Manning CD, Le Q. Semi-supervised sequence modeling with cross-view training. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics; 2018. p. 1914–25. https://www.aclweb.org/anthology/D18-1217.
50. Segura-Bedmar I, Martinez P, Herrero-Zazo M, Martínez P, Herrero Zazo M, Martinez P, Herrero-Zazo M, Martínez P, Herrero Zazo M. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013); 2013. https://doi.org/10.1.1.310.783.
51. Björne J, Kaewphan S, Salakoski T. UTurku : Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013); 2013. p. 651–9.
52. Chowdhury FM, Lavelli A. FBK-irst : A Multi-Phase Kernel Based Approach for Drug-Drug Interaction Detection and Classification that Exploits Linguistic Information. In: Seventh International Workshop on Semantic Evaluation; 2013.
53. Raihani A, Laachfoubi N. Extracting drug-drug interactions from biomedical text using a feature-based kernel approach. J Theor Appl Inform Technol. 2016.
54. Quan C, Hua L, Sun X, Bai W. Multichannel convolutional neural network for biological relation extraction. BioMed Res Int. 2016;2016:. https://doi.org/10.1155/2016/1850404.
55. Björne J, Salakoski T. Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing. 20181–11. https://doi.org/10.18653/v1/w18-2311.
56. Lim S, Lee K, Kang J. Drug drug interaction extraction from the literature using a recursive neural network. PLoS ONE. 2018;13(1):1–17. https://doi.org/10.1371/journal.pone.0190926.
57. Charniak E, Johnson M. Coarse-to-fine n-best parsing and maxent discriminative reranking. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05. Stroudsburg: Association for Computational Linguistics; 2005. p. 173–180. https://doi.org/10.3115/1219840.1219862.
58. De Marneffe M-C, MacCartney B, Manning CD. Generating typed dependency parses from phrase structure parses; 2006. p. 449–54. https://doi.org/10.1.1.74.3875.
59. Pyysalo S, Ginter F, Moen H, Salakoski T, Sophia A. Distributional Semantics Resources for Biomedical Text Processing. Lang Biol Med. 2013. https://doi.org/10.1074/jbc.M700492200.
60. Zhang S, Zheng D, Hu X, Yang M. 2015 Bidirectional Long Short-Term Memory Networks for Relation Classification. 201573–78. https://doi.org/10.18653/v1/P16-2034.
61. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Trans Signal Process. 1997;45:2673–81.
62. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging. 2015. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000274.
63. Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. 2016. https://doi.org/10.18653/v1/P16-1101.
64. Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL; 2014. p. 1746–51. http://aclweb.org/anthology/D/D14/D14-1181.pdf.
65. Yin W, Kann K, Yu M, Schütze H. Comparative study of cnn and rnn for natural language processing. CoRR. 2017;abs/1702.01923:.
66. Graves A, Wayne G, Danihelka I. Neural turing machines. arXiv preprint arXiv:1410.5401. 2014.
67. Ambartsoumian A, Popowich F. Self-attention: A better building block for sentiment analysis neural network classifiers. In: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. Brussels: Association for Computational Linguistics; 2018. p. 130–9. https://doi.org/10.18653/v1/W18-6219.
68. Li L, Guo Y, Qian S, Zhou A. An End-to-End Entity and Relation Extraction Network with Multi-head Attention: 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018, Changsha, China, October 19–21, 2018, Proceedings; 2018, pp. 136–46.
69. Jiang Z, Li L, Huang D, Jin L. Training word embeddings for deep learning in biomedical text mining tasks. In: Proceedings - 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015; 2015. https://doi.org/10.1109/BIBM.2015.7359756.
70. Zamani H, Croft WB. Relevance-based Word Embedding. 2017.
71. Bridle JS. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In: Soulié FF, Hérault J, editors. Neurocomputing. Berlin, Heidelberg: Springer; 1990. p. 227–36.
72. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A. Automatic differentiation in pytorch. In: NIPS-W; 2017.

Publisher's Note