Bacteria Biotope Relation Extraction via Lexical Chains and Dependency Graphs

Wuti Xiong[1], **Fei Li**[2], **Ming Cheng**[3], **Hong Yu**[2], **Donghong Ji**[1]
1. School of Cyber Science and Engineering, Wuhan University, China
2. Department of Computer Science, UMass Lowell, USA
3. Department of Medical Information,
The First Affiliated Hospital of Zhengzhou University, China
woody.xwt@gmail.com, dhji@whu.edu.cn

Abstract

In this article, we describe our approach for the Bacteria Biotopes relation extraction (BB-rel) subtask in the BioNLP Shared Task 2019. This task aims to promote the development of text mining systems that extract relationships between Microorganism, Habitat and Phenotype entities. In this paper, we propose a novel approach for dependency graph construction based on lexical chains, so one dependency graph can represent one or multiple sentences. After that, we propose a neural network model which consists of the bidirectional long short-term memories and an attention graph convolution neural network to learn relation extraction features from the graph. Our approach is able to extract both intra- and inter-sentence relations, and meanwhile utilize syntax information. The results show that our approach achieved the best F1 (66.3%) in the official evaluation participated by 7 teams.[1]

1. Chronic gastritis was recorded before treatment in all patients. Treatment reduced its activity and the presence of H. pylori.

   Relation: H. pylori *Live_in* Habitat patients

2. Atypical mycobacteria causing non-pulmonary disease in Queensland.

   Relation: mycobacteria *Live_in* Queensland
   Relation: mycobacteria *Exhibits* non-pulmonary disease

3. Erythromycin resistance was associated with Campylobacter coli.

   Relation: Campylobacter coli *Exhibits* Erythromycin resistance

Figure 1: Bacteria Biotopes relation examples. The Red, green and blue words denote Microorganism entities, Habitat entities and Phenotype entities respectively.

1 Introduction

The BioNLP Shared Task 2019 (Bossy et al., 2019) is a continuation of the previous efforts organized around the BioNLP Shared Task workshop series (Kim et al., 2009, 2011; Nédellec et al., 2013; Deléger et al., 2017). It aims to facilitate development and sharing of computational tasks of biomedical text mining and solutions to them. The Bacteria Biotope (BB) task is one of the six main tasks of the BioNLP Open Shared Tasks 2019. Three teams participated in the BB task when it was first organized in 2011. INRA Bibliome (Ratkovic et al., 2011) achieved the best Fscore of 45% with the Alvis system which used dictionary mapping, ontology inference and semantic analysis for NER, and co-occurrence-based rules for detecting relations between the entities. The 2013 BB task (Bossy et al., 2013) contained three subtasks, the first one concerning recognition and normalization of bacteria and habitat entities, and the other two subtasks involving relation extraction. Four teams participated in these tasks, with the UTurku TEES system (Björne and Salakoski, 2013) achieving the first places with F-scores of 42% and 14%. Compared to the 2013 BB task, the 2016 BB task contains more subtasks and its subtask2 only concerned relation extraction. The team VERSE (Lever and Jones, 2016) achieved the best F-scores of 55.8% in the subtask2.

The Bacteria Biotopes relation extraction (BB-rel) in the BioNLP Shared Task 2019 aims to automatically extract Microorganism-Habitat or Microorganism-Phenotype relationships from biomedical literature. The BB-rel task follows the previous Bacteria Biotopes shared tasks, annotating directed binary relationships between Microorganism, Habitat and Phenotype entities. Fig-

[1]Code: https://github.com/woodyXwt/BB19-rel

Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, pages 158–167
Hong Kong, China, November 4, 2019.

ure 1 shows some examples for each relationship. In the BB-rel task, not all the relations occur between two entities with the same sentence. In the preprocessing step, we found that there exist about one fourth of all relations whose argument entities are located in different sentences. Therefore, we need to build a model that does not only consider the entity relationship within one sentence, but also beyond the sentence boundary.

A lexical chain (Morris and Hirst, 1991) is a sequence of words which are semantically-similar or related. These words are related sequentially in the text, defining the topic of the text segment that they cover and establishing associations between sentences. Following this observation, some researchers have obtained success in many NLP tasks such as word sense induction(Tao et al., 2014) , machine translation (Mascarell, 2017) and text (Stokes et al., 2004) segmentation. In the BB-rel dataset, the sentences where inter-sentence relations occur usually express the same topic or have semantic associations each other. These features usually appear as some related words which can form lexical chains. Following this observation, we propose a novel approach to build an inter-sentence dependency graph based on lexical chains.

In this paper, we propose a novel relation extraction method for the BB-rel task by incorporating dependency graphs and lexical chains into the neural network. As shown in Figure 1, inter-sentence relations are usually expressed in inter-related sentences, and these sentences may contain semantically-related words which can form lexical chains. We utilize these lexical chains and dependency graphs to build an inter-sentence dependency graph for inter-sentence relation extraction. Specifically, we utilize word embedding to find the semantic relationships of words that occur in different sentences for building reliable lexical chains. Then, we use the Stanford CoreNLP toolkit (Manning et al., 2014) to obtain sentence-level dependency and part-of-speech (POS) information, and build an inter-sentence dependency graph based on these information and lexical chains.

After that, we employ a neural network model which consists of the bidirectional long short-term memories and attention-guided graph convolutional neural networks to extract features from the inter-sentence dependency graph. The features are fed into a multi-layer perceptron (MLP) to classify the relation between an entity pair.

	Train	**Dev**
Lives_In	715	395
Exhibits	281	138
Total relatonships	996	533
Intra-sentence relationships	885	467
Inter-sentence relationships	111	66

Table 1: BB-rel data statistics on the training and development set.

Our approach has two advantages. First, it is capable of extracting both intra-sentence and inter-sentence relations by connecting the dependency graphs of different sentences via lexical chains. Second, it is able to leverage syntax information. The results in the BB-rel task demonstrate the superiority of our method. It achieves the highest F1-score, the second highest precision and recall in the official evaluation.

2 Method

In this section, we first introduce our strategy of relation candidate generation. Then, the approach for constructing lexical chains is described. After that, we will introduce how to build inter-sentence dependency graphs. Lastly, the architecture of our neural network model is described.

2.1 Relation Candidate Generation

In the BB-rel dataset, if all candidate pairs (bacteria and habitat or phenotype) that occur in the document are enlisted as candidate training examples, the positive and negative examples will become very unbalanced because most entity pairs located beyond one sentence do not have any relation. Based on our observations, most entity pairs spanning more than two sentences have no relations between them. Therefore, we consider all entity pairs that span within two sentences as the candidates to generate training examples. The statistics of our dataset are summarized in Table 1.

2.2 Lexical Chain Construction

In previous work, there are mainly three approaches for constructing lexical chains. The first one utilized *WordNet* (Hirst and St-Onge, 1997) to capture the semantic relationship between words. The second approach (Remus and Biemann, 2013)

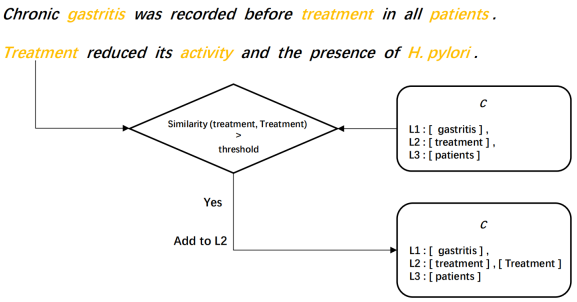


Figure 2: Process of lexical chain construction. Orange words denote nouns. C is the set of lexical chains. The similarity here refers to the cosine similarity between word vectors. We set the threshold to 0.5.

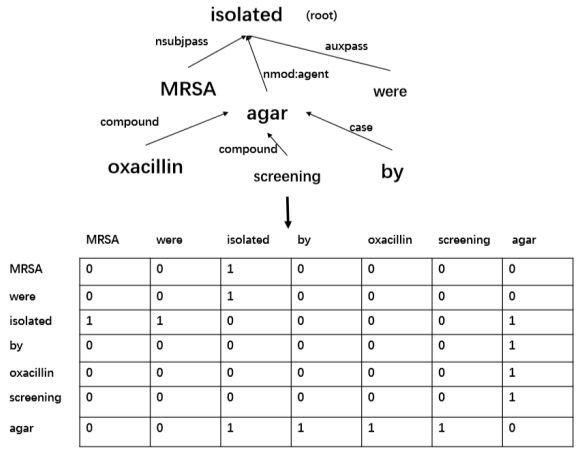
automatically extracted lexical chains using statistical methods . Another approach (Li et al., 2017) is based on semantic word vectors. In this paper, we assume that lexical relationships can be captured by calculating the similarity of their semantic vectors. To compute similarities, we use 200-dimensional pre-trained word vectors released by Pyysalo et al. (2013). Moreover, we only consider nouns for constructing the lexical chains since they usually contain relevant information.

Given a sentence, we first use the Stanford CoreNLP toolkit (Manning et al., 2014) to obtain POS tags for each word. Then we pick those words whose POS tags belonging to *N*= *(NN,NNP,NNS)* as candidates for chain construction. We take one candidate at a time and check where it should be placed. Assuming that *C* is the set of lexical chains, we add each candidate *w* to *C* according to the following steps (Figure 2):

- *Step 1*: each noun is treated as a candidate *w*. If *C* is empty, we will create a new lexical chain in *C* and add the current candidate *w* into it.

- *Step 2*: for the current candidate *w*, we traverse all the lexical chains in *C* and compute the similarity between the last word of each lexical chain and the current candidate *w*. If the similarity surpasses a predefined threshold, the current candidate *w* will be attached to the corresponding lexical chain.

- *Step 3*: if the current candidate *w* cannot be attached to any existing lexical chain, we will create a new lexical chain for it.



	MRSA	were	isolated	by	oxacillin	screening	agar
MRSA	0	0	1	0	0	0	0
were	0	0	1	0	0	0	0
isolated	1	1	0	0	0	0	1
by	0	0	0	0	0	0	1
oxacillin	0	0	0	0	0	0	1
screening	0	0	0	0	0	0	1
agar	0	0	1	1	1	1	0

Figure 3: An example of the dependency graph and its corresponding adjacent matrix. If there is a dependency relation between the node i and j in the dependency graph, the value of the element M_{ij} in the adjacent matrix is 1.

2.3 Dependency Graph Construction

In this section, we propose an approach to build an inter-sentence dependency graph by lexical chains. For an entity pair that occurs within the same sentence, we directly use their sentence dependency graph. If two entities occur in different sentences, we construct their dependency graph by lexical chains. We design two rules to build an inter-sentence graph. Here we define the following notations: C is the set of lexical chains, A and B are nouns belonging to sentence s_1 and sentence s_2, respectively.

- *Rule 1*: if A and B exist in the same chain of C, we will add an edge between A and B to build an inter-sentence dependency graph.

- *Rule 2*: if A and B do not appear in the same lexical chain, we will use the root nodes of two sentences to build the dependency inter-sentence graph.

Then we convert the dependency graph into an adjacency matrix. An example of such process is shown in Figure 3. Give a sequence $S = \{s_1, s_2, ..., s_n\}$, we considered its dependency graph as an undirected graph, which can be converted into an adjacent matrix. If there is a dependency relation between nodes i and j in the dependency graph, the element M_{ij} in the adjacent matrix is assigned with 1.

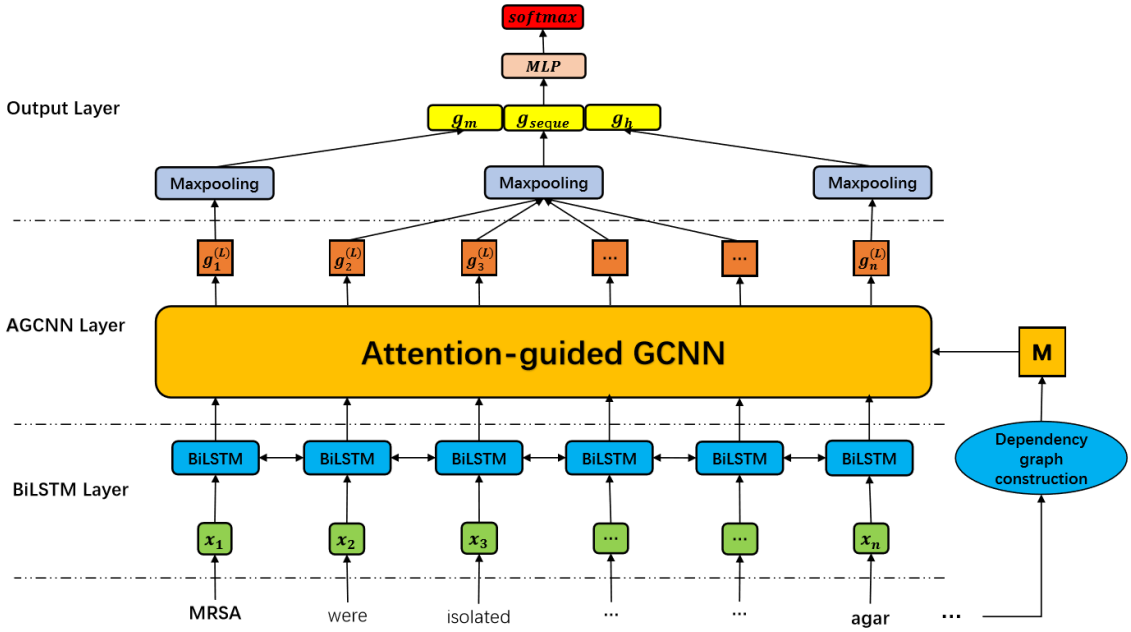


Figure 4: The architecture of our model. The input sentence is "MRSA were isolated by oxacillin screening agar" with a Microorganism entity "MRSA" and a Habitat entity "oxacillin screening agar". M denotes the adjacency matrix.

2.4 Neural Network Model

2.4.1 BiLSTM Layer

Figure 4 shows the neural network architecture of our model. It uses the words and POS tags as input. We adopt the 200-dimensional word embeddings and 20-dimensional POS tag embeddings. The final representation for the token is the concatenation x_i of the word embedding s_i and the POS tag embedding p_i. We initialize our word embeddings with the pre-trained biomedical embeddings (Pyysalo et al., 2013) and randomly initialize the POS tag embeddings.

After obtaining the word representation sequence $x = \{x_1, x_2, ..., x_n\}$, we leverage bidirectional LSTMs (Hochreiter, 1998) to encode the context information into each word. The forward and backward hidden states ($\overrightarrow{h_i}$ and $\overleftarrow{h_i}$) will be concatenated, formalized as $h_i = [\overrightarrow{h_i} \odot \overleftarrow{h_i}]$.

2.4.2 Attention-Guided GCNN Layer

We employ the attention-guided graph convolutional neural network (AGCNN) (Guo et al., 2019a) to incorporate the dependency information into word representations, which is composed of M identical blocks. Each block has three types of layers: attention-guided layer, densely connected layer, linear combination layer.

In the attention guided layer, we first update the representation of the node using a graph convolution network (GCNN) (Zhang et al., 2018). For an L-layer GCNN, we denotes the inputs in the first layer as $g_1^{(0)}, ..., g_n^{(0)}$ and the outputs in the last layer as $g_1^{(L)}, ..., g_n^{(L)}$. The $g_i^{(l)}$ denotes the output vectors of the node i in the l-th layer. The convolution operation in the l-th layer can be written as:

$$g^l = \sigma(\sum_{j=1}^{n} \tilde{M}_{ij}, W^l g^{l-1}/d_i + b^l), \qquad (1)$$

where W^l is a linear transformation, b^l is a bias term, and σ is a nonlinear function (e.g., $ReLU$). The $\tilde{M}$ can be computed by $M + I$, where $I \in \mathbb{R}^{n \times n}$ is an identity matrix and $d_i = \sum_{j=1}^{n} \tilde{M}_{ij}$ is the degree of node i in the dependency graph. Intuitively, during the graph convolution of each layer, each node gathers all the information of its neighboring nodes in the graph.

After the L-layer graph convolution operation, we transform the original dependency graph into a fully connected edge-weighted graph by constructing N (N is a hyper-parameter) attention-guided adjacency matrix. Each attention-guided adjacency matrix $\tilde{A}$ corresponds to a completely connected graph. In this paper, we use the multi-

head attention(Vaswani et al., 2017) to calculate $\tilde{A}$, which allows the model to focus on information from different representation sub-spaces. The output is computed as a weighted sum of values, where the weight is calculated by the function of the query and the corresponding key.

$$\tilde{A}^{(t)} = softmax(QW_i^Q \times (KW_i^K)^T/\sqrt{d})V, \tag{2}$$

where Q and K are both equal to the collective representation h^{l-1} at layer $l-1$ of the model. The projections are parameter matrices $W_i^Q \in \mathbb{R}^{d\times d}$ and $W_i^K \in \mathbb{R}^{d\times d}$. $\tilde{A}^{(t)}$ is the t-th attention guided adjacency matrix corresponding to the t-th head.

Following (Guo et al., 2019b), we employ the dense connection (Huang et al., 2017) into the our model to capture more structural information on the large graph. We concatenate the initial node representation $h_j^{(l)}$ and the node representations $g_j^{(1)}, ..., g_j^{(l-1)}$ produced in layer $1, ..., l-1$:

$$h_j^{(l)} = [x_j; g_j^{(1)}, ..., g_j^{(l-1)}], \tag{3}$$

Each densely connected layer has L sub-layers. The dimensions of these sub-layers d_{hidden} are decided by L and the input feature dimension d. In our model, we use $d_{hidden} = d/L$.

Then we use N separate dense connection layers to modify the computation of each layer as follows (for the t-th matrix $\tilde{A}^{(t)}$):

$$g_{t_i}^l = \rho(\sum_{j=1}^{n} \tilde{A}^{(t)} W_t^l h_i^l + b_t^l), \tag{4}$$

where $t = 1, ..., N$ and t selects the weight matrix and bias term associated with the attention guided adjacency matrix $\tilde{A}^{(t)}$. The column dimension of the weight matrix increases by d_{hidden} per sub-layer, i.e., $W_t^l \in \mathbb{R}^{d_{hidden}\times d^{(l)}}$ where $d^{(l)} = d + d_{hidden}(l-1)$.

Finally, we use linear combination layer to integrate representations from N different densely connected layers. Formally, the output of the linear combination layer is defined as:

$$g_{comb} = W_{comb} g_{out} + b_{comb}, \tag{5}$$

where g_{out} is the output by concatenating outputs from N separate densely connected layers, i.e., $g_{out} = [g^{(1)}; ...; g^{(N)}] \in \mathbb{R}^{d\times d}$. $W_{comb} \in \mathbb{R}^{d\times d}$ is a weight matrix and b_{comb} is a bias vector for the linear transformation.

2.4.3 Output Layer

We treat the BB-rel task as a classification task. $S = [s_1, ..., s_n]$ denotes a sequence, s_i is the i-th token, M_e and H_e denote Microorganism and Habitat or Phenotype entities. The entities may consist of several tokens, namely $[s_{e_1}, ..., s_{e_n}]$ and $[s_{h_1}, ..., s_{h_n}]$. The goal of the BB-rel task is to predict whether there is a "Live_in" or "Exhibits" relationship between the entities H_e and M_e.

After applying the attention-guided GCNN layer to the input word vectors, we obtain the representation for each word. The sequence representation can be obtained using the following equation:

$$g_{seque} = f(g_1, ..., g_n), \tag{6}$$

where $g_1, ..., g_n$ denotes the outputs of the the attention-guided GCNN layer and $f : R^{d\times n} \rightarrow R^d$ is a max-pooling function. Since we also observed that the entity information is often critical for BB-rel extraction, the entity representations M_e and H_e are also used, given by:

$$\begin{aligned} g_m &= f(g_{m_1}, ..., g_{m_n}), \\ g_h &= f(g_{h_1}, ..., g_{h_n}). \end{aligned} \tag{7}$$

Inspired by (Santoro et al., 2017; Lee et al., 2017), we obtained the final feature for BB-rel extraction by feeding the sequence and entity representations into a multi-layer perceptron (MLP):

$$g_{final} = MLP([g_{seque}; g_m; g_h]), \tag{8}$$

where "[]" denotes the concatenation operation. Finally, g_{final} is fed into a softmax layer to compute the probability distribution over all classes. During training, our model uses the cross-entropy loss:

$$loss(\theta) = -\sum_{j=1}^{J} logP(y_j|S_j), \tag{9}$$

where J denotes the size of the training set $S = \{(S_1, y_1), ..., (S_J, y_J)\}$ and y_j denotes the gold answer of the j-th training instance. $P(y_j|S_j)$ denotes the probability that S_j belongs to y_j, which is calculated as $P(y_j|S_j) = softmax(g_{final})$.

3 Experiments

3.1 Evaluation Metrics

We send the prediction results of our model on the test set to the task organizer for evaluation. The

Hyper-parameter	Value
Number of heads N	2
Block number M	2
Word emb size	200
POS emb size	20
LSTM hidden size	300
BiSTM layer	2
GCNN layer	2
GCNN output size	200
Dropout of GCNN	0.5
Multi-head attention head	3
Sublayers	5
d_{hidden}	300
Epoch	100
Decay rate	0.9
Learning rate	0.5
Optimizer	sgd
MLP layer	1

Table 2: Hyper-parameter setting.

Team Name	P	R	F1
Amrita_Cen	41.9	61.7	49.9
UTU	47.3	65.5	55.5
BLAIR_GMU	54.7	64.9	59.4
BOUN-ISIK	51.3	**73.1**	60.3
Yuhang_Wu	55.1	67.0	60.4
AliAI	**68.2**	62.0	64.9
Our method	62.9	70.2	**66.3**

Table 3: The official results of the BB-rel task.

performances of our model were evaluated by the standard evaluation measures: precision (P), recall (R) and F1-score (F1).

3.2 Hyper-parameter

The hyper-parameter setting is listed in Table 2. We tuned hyper-parameters based on the development set.

3.3 Official Results

The official results on the test set are shown in Table 3. There are totally 7 teams participating in the BB-rel task. Each team could submit up to 2 predictions. We report the top results for all teams. As we can see, our method achieved the highest F1 (66.3%), and the second highest precision (62.9%) and recall (70.2%).

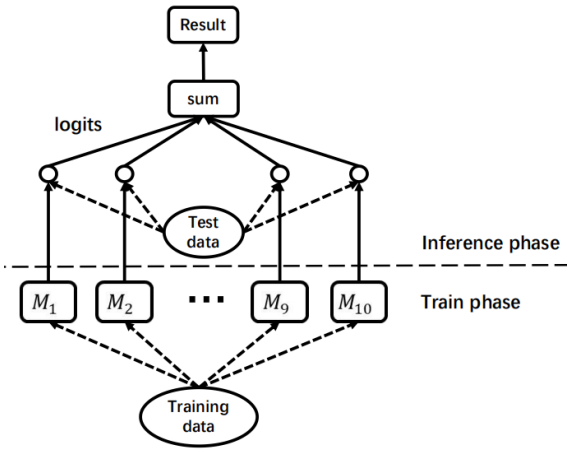


Figure 5: Ensemble training and inference.

3.4 Ensemble Training and Inference

In relation extraction tasks, the ensemble training and inference have proven to be an effective way to improve performance of the neural network model (Mehryary et al., 2016; Lim and Kang, 2018). Following previous work (Lim and Kang, 2018), we improve performance of our model using the ensemble training and inference. We sum the output probabilities (logits) of ensemble members, which are generated using the same neural network model but different weight initialization.

As shown in Figure 5, M1 to M10 are the models using the same structure and hyper-parameters. In the training phase, we independently trained each ensemble member with different initialized parameters. When inferring a relation for an easy sample, the trained ensemble members make relatively consistent predictions. When inferring for a difficult sample, the trained ensemble members may make different predictions. We incorporate the voting results of 10 ensemble members to produce final results.

To investigate the effectiveness of ensemble training and inference, we conducted the following experiment on the development set. First, we run five times of our model and average the results as the final result of the single model as shown in Table 4. Second, we run one time for the ensemble training and inference. The results show that the approach using ensemble training and inference achieved relatively balanced precision and recall, thus yielding a better F1.

Method	P	R	F1
Single	59.1	69.3	63.8
Ensemble	63.1	68.4	65.7

Table 4: Effects of ensemble training and inference.

	Relation	P	R	F1
Intra	Live_in	61.6	60.0	60.8
	Exhibits	73.4	80.6	76.8
	Total	64.8	65.2	65.0
Intra+Inter	Live_in	59.5	63.7	61.5
	Exhibits	72.8	82.4	77.3
	Total	63.1	68.4	65.7

Table 5: Results of recognizing inter- and intra-sentence relations.

3.5 Results of Recognizing Inter- and Intra-Sentence Relations

In this section, we discuss the performance of our model in Intra- and inter-sentence relation. As shown in Table 5, we obtained an F1-score of 65.0 when we only evaluated the intra-sentence relationships. When we evaluated both intra- and inter-sentence relationship, F1-score, Recall increase by 0.7% and 3.2% respectively. But Precision drops by 1.7%. We can also see from the table that the performance of "Exhibits" relation is better than the performance of the "Live_in" relation. Because most of the "Exhibits" relation happen within a sentence and have a certain pattern.

3.6 Effects of Lexical Chains

In order to verify the effectiveness of constructing inter-sentence dependency graphs by lexical chains, we also conducted related experiments on development set. The experimental results are shown in Table 6. "lexical chains" denotes the model employing the proposed method that constructs inter-sentence dependency graphs by lexical chains. "root nodes" denotes the model where the inter-sentence dependency graphs are built using root nodes. Table 6 shows the performance comparison of the "lexical chains" method and the "root nodes" method on the development set. The "lexical chains" method obtained better performance than the "root nodes" model. This demonstrates our idea is effective. The relevant sentences are usually expressed using relevant words. These relevant words found by lexical chains can be used as the associations to connect the dependency graphs of different sentences. Therefore, we can build an effective representation for an inter-sentence entity pair.

Method	P	R	F1
Root nodes	62.7	67.3	64.9
Lexical chains	63.1	68.4	65.7

Table 6: Effects of lexical chains.

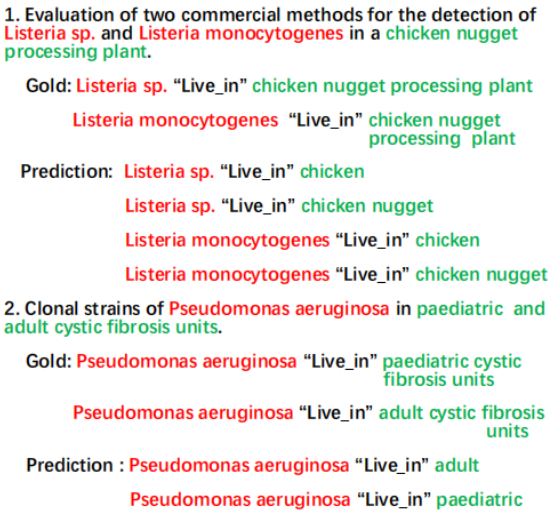


Figure 6: Examples of false positives. The Red and green words denote Microorganism and Habitat entities respectively.

3.7 Error Analysis

In this section, we manually analyzed what cases lead to false positives, since those are more critical than false negatives. Figure 6 shows some examples of false positives. The most of false positives are caused by overlapping target entities. For example, there is a "Live_in" relation between "Listeria sp." and "chicken nugget processing plant", but there is no "Live_in" relation between "Listeria sp." and "chicken" or "chicken nugget". The reason for these errors is that the model is confused by overlapping entities with similar context.

4 Related Work

In the natural language processing community, there are a number of related competitions and tasks (Wei et al., 2015; Nédellec et al., 2013; Deléger et al., 2016). Most prior work focused on extracting the relations within one sentence, and ignored the relations beyond one sentence.

In the NLP community, it has proven to be effective to combine linguistic features with neural networks for relation extraction (Zhou et al., 2015; Miwa and Bansal, 2016). Bunescu et al. (2005) demonstrated that the relationship of an entity pair can be captured along their shortest dependency path in the dependency graph because the words on the shortest dependency path concentrate the most relevant information and diminish redundant information. Following this observation, several studies (Xu et al., 2015; Liu et al., 2015) achieved outstanding performance by combining shortest dependency paths with various neural networks. As deep learning develops, some attention-based neural architectures (Zhou et al., 2016; Lin et al., 2016) have been proposed for relation classification and show the state-of-the-art performance. But with a few exceptions, almost all related work only focused on intra-sentence relation extraction, without considering the inter-sentence relations.

Recent work has explored some approaches to consider inter-sentence relations, such as Graph LSTMs (Peng et al., 2017), self-attention (Verga et al., 2018), Graph CNNs (Sahu et al., 2019). However, none of these work investigated lexical chains for inter-sentence relation extraction. In the future, we will evaluate our approach on some large-scale datasets for intra- and inter-sentence relation extraction (Yao et al., 2019).

5 Conclusion

In this paper, we describe our approach used for participating the Bacteria Biotope task at BioNLP-OST 2019. Our approach achieved very competitive performance in the official evaluation. We found that the idea using lexical chains to build inter-sentence dependency graphs is effective. Moreover, ensemble training and inference can improve the performance of our model. The attention-guided graph convolution neural network performs well in extracting Bacteria Biotope relations. However, our approach is not specific to Bacteria Biotope relation extraction, and it can be applied to other relation extraction tasks.

Acknowledgments

This work is supported by the Young Scientists Fund of the National Natural Science Foundation of China (No. 61802350), the National Natural Science Foundation of China (No. 61772378), the Major Projects of the National Social Science Foundation of China (No. 11&ZD189), the National Key Research, Development Program of China (No. 2017YFC1200500).

References

Jari Björne and Tapio Salakoski. 2013. TEES 2.1: Automated annotation scheme learning in the bionlp 2013 shared task. In *Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, August 9, 2013*, pages 16–25.

Robert Bossy, Louise Deléger, Estelle Chaix, Mouhamadou Ba, and Claire Nedellec. 2019. Bacteria Biotope at BioNLP Open Shared Tasks 2019. In *Proceedings of the BioNLP Open Shared Tasks 2019 Workshop*.

Robert Bossy, Wiktoria Golik, Zorana Ratkovic, Philippe Bessières, and Claire Nédellec. 2013. Bionlp shared task 2013 - an overview of the bacteria biotope task. In *Proceedings of the BioNLP Shared Task 2013 Workshop, Sofia, Bulgaria, August 9, 2013*, pages 161–169.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *EMNLP*, pages 724–731.

Louise Deléger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferre, Philippe Bessieres, and Claire Nedellec. 2016. Overview of the bacteria biotope task at bionlp shared task 2016. In *Proceedings of the 4th BioNLP shared task workshop*, pages 12–22.

Louise Deléger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferré, Philippe Bessières, and Claire Nédellec. 2017. Overview of the bacteria biotope task at bionlp shared task 2016. In *Bionlp Shared Task Workshop-association for Computational Linguistics*.

Zhijiang Guo, Yan Zhang, and Wei. Lu. 2019a. Attention guided graph convolutional networks for relation extraction. In *ACL*.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019b. Densely connected graph convolutional networks for graph-to-sequence learning. *TACL*, 7:297–312.

Graeme Hirst and David St-Onge. 1997. Lexical chains as representations of context for the detection and correction of malapropisms. *Lecture Notes in Physics*, 728(9):123–149.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):–.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *CVPR*, pages 2261–2269.

J D Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii. 2011. Overview of bionlp shared task 2011. In *Bionlp Shared Task Workshop*.

Jin Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'Ichi Tsujii. 2009. Overview of bionlp'09 shared task on event extraction. In *Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*, pages 188–197.

Jake Lever and Steven J. Jones. 2016. VERSE: event and relation extraction in the bionlp 2016 shared task. In *Proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, Berlin, Germany, August 13, 2016*, pages 42–49.

Liunian Li, Xiaojun Wan, Jin-ge Yao, and Siming Yan. 2017. Leveraging diverse lexical chains to construct essays for chinese college entrance examination. In *IJCNLP*.

Sangrak Lim and Jaewoo Kang. 2018. Chemical-gene relation extraction using recursive neural network. *Database*, 2018:bay060.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, pages 2124–2133.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and WANG Houfeng. 2015. A dependency-based neural network for relation classification. In *ACL*, pages 285–290.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David Mcclosky. 2014. The stanford corenlp natural language processing toolkit. In *Meeting of the Association for Computational Linguistics: System Demonstrations*.

Laura Mascarell. 2017. Lexical chains meet word embeddings in document-level statistical machine translation. In *Discourse in Machine Translation (DiscoMT)*.

Farrokh Mehryary, Jari Björne, Sampo Pyysalo, Tapio Salakoski, and Filip Ginter. 2016. Deep learning with minimal training data: Turkunlp entry in the bionlp shared task 2016. In *Proceedings of the 4th BioNLP Shared Task Workshop, BioNLP 2016, Berlin, Germany, August 13, 2016*, pages 73–81.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*, pages 1105–1116.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7.

Claire Nédellec, Robert Bossy, Jin Dong Kim, Jung Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Bionlp Shared Task Workshop*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

S. Pyysalo, F. Ginter, H. Moen, T. Salakoski, and S. Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of LBM 2013*, pages 39–44.

Zorana Ratkovic, Wiktoria Golik, Pierre Warnier, Philippe Veber, and Claire Nédellec. 2011. Bionlp 2011 task bacteria biotope - the alvis system. In *Proceedings of BioNLP Shared Task 2011 Workshop, Portland, Oregon, USA, June 24, 2011*, pages 102–111.

Steffen Remus and Chris Biemann. 2013. Three knowledge-free methods for automatic lexical chain extraction. In *NAACL*.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Tim Lillicrap. 2017. A simple neural network module for relational reasoning. In *NIPS*, pages 4967–4976.

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. 2004. Select: a lexical cohesion based news story segmentation system. *Ai Communications*, 17(17):3–12.

Qian Tao, Donghong Ji, and Congling Xia. 2014. Word sense induction using lexical chain based hypergraph model. In *Coling*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2015. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, volume 14.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*, pages 1785–1794.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *ACL*, pages 1213–1222.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*, pages 207–212.