Protein/Gene Entity Recognition and Normalization with Domain Knowledge and Local Context

Weihong Yao[1], Xuefei Li[1], Zongze Li[2(✉)], Zhe Liu[1], and Shixian Ning[1]

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian, China
weihongy@dlut.edu.cn, {lixuefei,njjnlz,ningshixian}@mail.dlut.edu.cn
[2] School of Electronic and Information Engineering, Dalian University of Technology, Dalian, China
lizongze@mail.dlut.edu.cn

Abstract. Biomedical named entity recognition and normalization aim at recognizing biomedical entity mentions from text and mapping them to their unique database entity identifiers (IDs), which are the primary task of biomedical text mining. However, name variation and entity ambiguity problems make this task challenging. In this paper, we leverage domain knowledge by a novel knowledge feature representation method to recognize more entity variants, and model important local context through a dual attention mechanism and a gating mechanism to perform entity normalization. Experimental results on the BioCreative VI Bio-ID corpus show that our proposed system achieves the new state-of-the-art performance (0.844 *F*1-score for protein/gene entity recognition and 0.408 *F*1-score for normalization).

Keywords: Entity recognition · Entity normalization · Domain knowledge · Local context

1 Introduction

The unprecedented growth in biomedical literature necessitates perpetual reformations of automated text mining to uncover the knowledge contained in the vast biomedical text. The first step of biomedical text mining is biomedical named entity recognition and normalization, which facilitates downstream applications such as relation extraction [1] and knowledge base completion [2]. However, manually annotating all of them is time-consuming. New recognition methods and tools need to be developed to support more effective extraction of biomedical entities and their identifiers (IDs).

To address these needs, the Bio-ID track in BioCreative VI focuses on accurately recognizing entities and associating them with their corresponding database IDs [3]. Two subtasks are involved in this task: (1) biomedical named entity recognition (BioNER) and (2) normalization (BioNEN), also known as disambiguation.

J.-F. Hong et al. (Eds.): CLSW 2019, LNAI 11831, pp. 371–380, 2020.
https://doi.org/10.1007/978-3-030-38189-9_39

BioNER is usually considered as a sequence labeling task. Traditional machine learning (ML)-based methods [4, 5] use the BIO (Begin, Inside, Outside) labeling scheme to tag each word for entity recognition based on one-hot represented linguistic features. However, these methods require complicated feature engineering, which is labor intensive. In recent years, deep learning techniques have been proposed to learn low-dimensional feature representations of words for entity recognition without manual feature engineering [6–8]. Among them, bidirectional long short-term memory with conditional random field model (BLSTM-CRF) exhibits promising results. In addition, Devlin et al. [9] propose a new language model framework Bidirectional Encoder Representations from Transformers (BERT), which has achieved the highest performance in entity recognition tasks.

Moreover, large-scale knowledge bases (KBs), such as UniProt [10] and NCBI gene [11], usually contain rich domain knowledge, which is quite useful for BioNER task. Therefore, how to represent knowledge and introduce it to the recognition model deserves further exploration.

Compared with BioNER, BioNEN is a more challenging task, whose purpose is to normalize each recognized entity to its database ID. Prior work uses dictionary matching and develops a set of heuristic rules [5, 8] to resolve ambiguous mentions. These methods are simple and effective, but rely heavily on the integrity of the dictionary and the design of the rules. Recently, deep learning-based methods have achieved considerable success in entity normalization task [12, 13]. These methods use neural networks to learn context representations of entity mentions, and then calculate the similarity between the candidate IDs and the context representations to determine which candidate ID is correct.

This paper aims at protein/gene named entity recognition (PNER) and normalization (PNEN). We propose a pipeline identification system, which leverages entity knowledge from biomedical KBs by a BLSTM-CRF model for PNER and captures important local context by a dual attention-based Convolutional Neural Network model for PNEN. Specifically, UniProt and NCBI gene are used as a form of domain knowledge to generate n-gram boolean knowledge features to help BLSTM-CRF model to recall more protein/gene mentions. Then, we employ a dual attention mechanism and a gating mechanism to capture important local context for mention disambiguation.

The contributions of this paper can be summarized as follows: (1) knowledge features are effectively introduced to BLSTM-CRF model for improving PNER performance, and (2) important local context is explicitly captured by our disambiguation model for promoting PNEN performance.

2 Entity Recognition

As in previous work, we treat PNER as a sequence labeling problem whose goal is to assign a label to each token in a sentence. It can be divided into two steps: (1) feature extraction; (2) BLSTM-CRF model training.

2.1 Feature Extraction

Besides word and character features, we use GENIA Tagger tool [14] to extract linguistic features, such as part of speech (POS) and chunking features, to enrich the information of each token.

Furthermore, a large amount of common sense information is difficult to reflect in the sample data, resulting in the feature representation that the neural network learns lacking the general expression of the text. This paper extracts the language model features through the pre-trained language model BERT, and indirectly introduces common sense information into the entity recognition model.

Finally, UniProt and NCBI gene are used as a form of domain knowledge for knowledge features extraction. The knowledge features proposed for PNER are represented as n-gram boolean vectors, which are called KF. Specifically, text segments based on the context of each token x_i are constructed using the pre-defined feature templates which are listed in Table 1. For a text segment that appears in the feature templates, we can generate a binary value to indicate whether the text segment is an entity in the UniProt and NCBI gene databases or not. A 7-dimensional boolean vector containing the entity boundary information is generated as KF.

Table 1. KF feature templates for each token.

Type	Template
1-gram	x_i
2-gram	$x_{i-1}x_i$, x_ix_{i+1}
3-gram	$x_{i-2}x_{i-1}x_i$, $x_ix_{i+1}x_{i+2}$
4-gram	$x_{i-3}x_{i-2}x_{i-1}x_i$, $x_ix_{i+1}x_{i+2}x_{i+3}$

2.2 BLSTM-CRF Mode

The architecture of BLSTM-CRF mainly consists of three components: an embedding layer, a BLSTM layer and a CRF layer.

Embedding Layer: This layer is used to map features to distributed representations. For a sentence with n tokens, each token is represented as a d-dimensional pre-trained word embedding. Character-level feature learned by a character-level BLSTM model is also used to address the out-of-vocabulary (OOV) problem. The POS, chunking features are all embedded into randomly initialized vectors and language model features are obtained by pre-trained BERT model. Through the embedding layer, these obtained feature vectors are concatenated as an input sequence $\boldsymbol{I} = \{I_1, I_2, \ldots, I_n\}$ to predict an output label sequence $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$.

BLSTM Layer: This layer is beneficial to have access to both past (forwards) and future (backwards) context information. We concatenate a hidden state of forward LSTM $\vec{h}_t = LSTM(I_t, \vec{h}_{t-1})$ and that of backward LSTM $\overleftarrow{h}_t = LSTM(I_t, \overleftarrow{h}_{t+1})$ as the final output $h_t = [\vec{h}_t; \overleftarrow{h}_t]$ of BLSTM at the t-th time step.

Then the output $\boldsymbol{h} = \{h_1, h_2, \ldots, h_n\} \in \mathbb{R}^{d \times n}$ of BLSTM feed to a two-layer perceptron. We take a hyperbolic tangent function as the activation function. The transformation is as follows:

$$\boldsymbol{P} = \boldsymbol{V}(\tanh(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b})) \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{(d/2) \times d}$, $\boldsymbol{V} \in \mathbb{R}^{k \times (d/2)}$, $\boldsymbol{b} \in \mathbb{R}^{(d/2) \times n}$ are the trainable parameters, and k is the number of labels.

CRF Layer: To make dependence between output tags, a linear chain CRF is added on top of the BLSTM layer. The linear chain CRF is given by:

$$P(\boldsymbol{y}|\boldsymbol{I}) = \frac{1}{Z(\boldsymbol{I})} \exp\left(\sum\nolimits_{i=1}^{n} \left(T_{y_{i-1}, y_i} + P_{i, y_i}\right)\right) \tag{2}$$

where P_{i,y_i} represents the score of the label y_i of the i-th token in the sequence, and T_{y_{i-1},y_i} represents the transition probability from the label y_{i-1} to label y_i. $Z(\boldsymbol{I})$ is a normalization factor defined as follows:

$$Z(\boldsymbol{I}) = \sum_{\mathbf{y}} \exp\left(\sum\nolimits_{i=1}^{n} \left(T_{y_{i-1}, y_i} + P_{i, y_i}\right)\right) \tag{3}$$

During the training period, our objective is to maximize the log-likelihood of the correct label path. During the prediction period, CRF uses Viterbi decoding to find the best path with the highest probability:

$$y^* = \underset{y' \in \mathbf{Y}}{\operatorname{argmax}}\, P(y'|I) \tag{4}$$

where $\boldsymbol{Y}$ denotes the set of possible label sequences for the input sequence $\boldsymbol{I}$.

2.3 Post-processing

In order to correct some errors that may exist in the prediction results and further improve the recognition performance, we use the following three heuristic rules to post-process the prediction results:

(1) Tagging consistency: the same entity mentions appeared in the same document should be tagged with the same labels. To meet this requirement, we assume that if the same entity mentions appear more than twice in the same document and the length of the mention is not less than three, these mentions should be tagged with the same label as the first entity mention.
(2) Bracket balance: entity mentions with an odd number of brackets should be corrected until the brackets are balanced.
(3) Remove uninformative terms: some mentions known for being non-informative or unwanted terms should be removed.

3 Entity Normalization

In this section, we explain how to normalize the recognized entity mentions to their standard database IDs. PNEN consists of two steps: (1) candidate generation, (2) entity disambiguation.

3.1 Candidate Generation

For each recognized entity mention, we use the following two methods to generate the corresponding candidate IDs:

Dictionary Matching: This method uses the exact string matching to find candidate IDs from a name dictionary, which is built by mapping all annotated entities that occur in the training set to the list of IDs that entities have been linked to. If this method fails to return any results, we then turn to the second method to continue retrieving candidates.

API Lookup: This method uses the APIs provided by UniProt and NCBI gene databases to search for candidate IDs. For memory saving considerations, the top 5 results returned by API lookup are taken as the candidate IDs of the entity mention.

3.2 Entity Disambiguation

Due to the entity ambiguity, many mentions may be linked to multiple candidate IDs. To solve this problem, we design a dual attention-based Convolutional Neural Network model (Att-CNN) to eliminate the ambiguity of these entities. The architecture of Att-CNN consists of four components: an embedding layer, an attention layer, a gating layer and a softmax layer, as shown in Fig. 1.

Embedding Layer: In this layer, the left local context $\{x_{i-w}, \ldots, x_{i-2}, x_{i-1}\}$ and right local context $\{x_{i+1}, x_{i+2}, \ldots, x_{i+w}\}$ of the entity mention x_i are represented as the corresponding context word embeddings $\{e_{i-w}, \ldots, e_{i-2}, e_{i-1}\}$ and $\{e_{i+1}, e_{i+2}, \ldots, e_{i+w}\}$, where $e_i \in \mathbb{R}^d$ is a d-dimensional word embedding and w is the window size of context. Then we apply shared convolution kernels to perform convolution operations over the left and right context embedding matrixes to capture semantic features $\boldsymbol{o}^l$ and $\boldsymbol{o}^r$ respectively.

Attention Layer: In this layer, a dual attention mechanism takes the candidate ID as a control signal to capture important context clues. The following equations provide details of our attention layer

$$\alpha_t = \text{softmax}(f(o_t, e_{cand})) \tag{5}$$

$$o_{att} = \sum_t \alpha_t o_t \tag{6}$$

where the similarity score between the output of embedding layer o_t at the t-th time step and the candidate entity ID embedding e_{cand} is calculated by a function

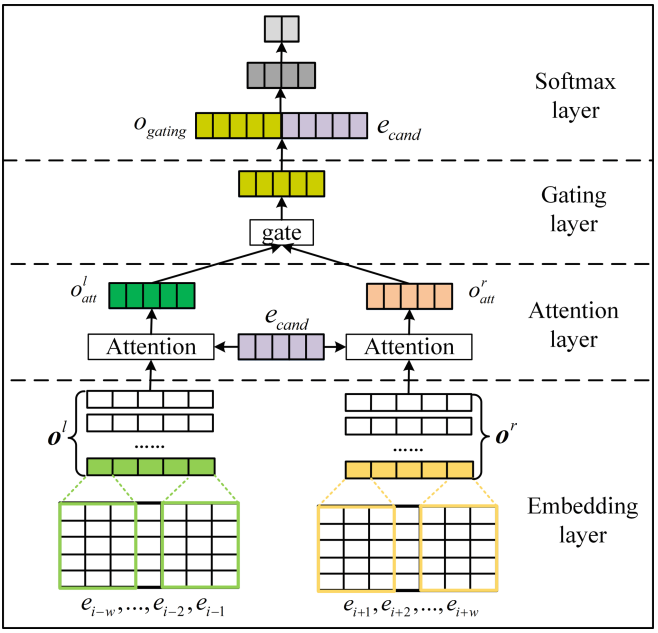

Fig. 1. Architecture of the Att-CNN model.

$f(o_t, e_{cand}) = \boldsymbol{W}_f o_t + \boldsymbol{V}_f e_{cand} + b_f$, where $\boldsymbol{W}_f, \boldsymbol{V}_f$ and b_f are the parameters that need to be trained. Then a softmax function is used to normalize the similarity score to get the attention weight α_t. The context representation o_{att} is the weighted sum of the output vectors of each time steps of the semantic features.

Gating Layer: In this layer, a gating mechanism is employed to dynamically combine left context representation o^l_{att} and right context representation o^r_{att} from a dual attention mechanism. The gate is computed as follows:

$$g = \sigma\left(\boldsymbol{W}_g o^l_{att} + \boldsymbol{V}_g o^r_{att} + b_g\right) \tag{7}$$

where $\boldsymbol{W}_g$, $\boldsymbol{V}_g$, b_g are the model parameters that need to be trained, and σ denotes an element-wise sigmoid function.

The final context representation o_{gating} is computed using a gating mechanism,

$$o_{gating} = g \odot o^l_{att} + (1 - g) \odot o^r_{att} \tag{8}$$

where $\odot$ denotes element-wise product between two vectors.

Softmax Layer: In this layer, the output of gating mechanism o_{gating} and the candidate ID embedding e_{cand} are concentrated as the final feature representation, which is then fed to a 2-layer perceptron using softmax function. The cross-entropy loss function is used as the training objective. During the test, the candidate ID which gets the highest probability will be selected as the final result for the mention.

4 Experiments

4.1 Experimental Settings

Dataset: The Bio-ID corpus released by the BioCreative VI Track 1 [3] is used to evaluate the performance of our system. The training set consists of 570 documents with 58,853 annotations for gene/protein entities, and the test set consists of 196 documents with 16,293 annotations for gene/protein entities. We train our model using the training set, and evaluate it on the test set.

Network Details: In our experiments, words are initialized with 200-dimensional pre-trained word embeddings [15]. The vector dimensions for character, POS, chunking, and KF features are 50, 25, 10, and 15 respectively. The size of the local context window w is set to 10 for the disambiguation model. RMSProp [16] is used to optimize all parameters of the BLSTM-CRF model and the Att-CNN model. Mini-batch size is set to 32 for both models.

Evaluation Metrics: During the testing, we evaluate our system by official evaluation scripts. For PNER, the scoring tool calculates the precision (P), recall (R) and $F1$-score ($F1$) at two levels: strict span match and span overlap match. The strict span match requires that span in byte level must exactly match the golden annotation, including all entities with or without IDs. Span overlap match slightly relaxes this requirement, allowing the span to completely overlap or partly override the golden annotation. For PNEN, the levels of scoring include micro-average and macro-average, which calculate P, R and $F1$ at the sentence level and document level respectively.

4.2 PNER Results

We explore the effects of linguistic features, BERT and KF on the performance of our recognition model in this experiment. We design a baseline method that uses the concatenation of the word embedding and character embedding as input. Table 2 lists the results of different combinations of these features on the test set.

Table 2. Official PNER results on the test set.

System	Strict span match			Span overlap match		
	P	R	F1	P	R	F1
Baseline	0.749	0.816	0.781	0.800	0.871	0.834
+ linguistic	0.817	0.764	0.789	0.868	0.812	0.839
+ BERT	0.802	0.809	0.805	0.778	0.878	0.825
+ KF	0.753	0.831	0.790	0.803	0.886	0.842
+ linguistic + BERT + KF	0.822	0.827	**0.825**	0.841	0.846	**0.844**
Sheng et al. [8]	0.509	0.613	0.556	0.686	0.826	0.749
Kaewphan et al. [5]	0.729	0.739	0.734	0.825	0.836	0.831

From Table 2, we can see that linguistic features improve *F*1-score by 0.5% under the overlap criterion over the baseline. The main reason is that some entity boundary errors can be revised by the linguistic information. Knowledge feature KF bring 1.5% increase in recall under the overlap criterion over the baseline, which demonstrates that the entity name knowledge of protein/gene provided by UniProt and NCBI gene KBs are effective for PNER. When linguistic features, knowledge feature KF and Language model feature BERT are added, the best performance (an improvement of 1.0% in overlap *F*1-score) is achieved. This shows that these three features are complementary.

To further demonstrate the effectiveness of our PNER method, the results of the other two top-ranked systems developed by [5, 8] are also shown in the Table 2. Sheng et al. [8] use the concatenation of word embedding and character embedding as the input of BLSTM-CRF model, but without using other external knowledge or linguistic features. Kaewphan et al. [5] propose a CRF-based method for PNER, in which complex feature engineering is employed to improve the performance. As can be seen from the results, our recognition method (BLSTM-CRF model with all the proposed features) acquires a better *F*1-score than both of them.

4.3 PNEN Results

For the protein/gene entity normalization, we consider the following variant methods to verify the validity of our disambiguation model. (1) w/o attention: This variant removes the attention mechanism from our Att-CNN model, that is, directly using the CNN to obtain the left and right context representations. (2) w/o gating: This variant removes the gating mechanism from our Att-CNN model, and directly concatenate the left context representation, the right context representation and the candidate ID embedding for classification. (3) w/o attention and gating: This variant removes both attention mechanism and the gating mechanism from our Att-CNN model, that is, directly using the CNN to obtain the left and right context representations, then concatenates them with the candidate ID embedding for classification. Table 3 presents the official protein/gene normalization results on the test set.

Table 3. Biomedical Entity Normalization results on the test set.

System	Micro-averaged			Macro-averaged		
	P	R	F1	P	R	F1
Att-CNN	0.526	0.333	**0.408**	0.615	0.376	**0.353**
w/o attention	0.481	0.329	0.391	0.566	0.379	0.343
w/o gating	0.505	0.321	0.392	0.595	0.363	0.338
w/o attention and gating	0.483	0.314	0.381	0.585	0.355	0.337
Sheng et al. [8]	0.170	0.224	0.193	–	–	–
Kaewphane et al. [5]	0.472	0.343	0.397	–	–	–

For w/o attention, when we remove the attention mechanism, the $F1$ of PNEN drops 1.7% under the micro-averaged criterion. The plausible reason is that the proposed attention mechanism allows the system to focus on finding useful information in the context that is more relevant to the candidate ID, thereby improving the quality of the output context embeddings.

For w/o gating, when we remove the gating mechanism, the performance of PNEN also drops under the micro-averaged criterion. By introducing the gating mechanism to control the update and fusion of the left and right context representations, the disambiguation model could dynamically determine how much information can be delivered to the final context semantic representation to find the balance between the left and right representations.

For w/o attention and gating, when we remove both attention mechanism and the gating mechanism, the performance of PNEN further drops. This proves that the two mechanisms are effective for PNEN.

Finally, we compare our PNEN system with the same two top-ranked systems [5, 8] as well. Sheng et al. [8] compile a contextual dictionary based on the training set and then check if the entity mention is in this contextual dictionary. If in the dictionary, they normalize the mention to the known ID that shares the most contextual words with the sequence the entity belongs to. For cases without matched IDs in the compiled dictionary, they use the same API sources as ours, to search for candidate IDs and directly assign the first ID match to ambiguous mentions. Kaewphane et al. [5] apply exact string matching to retrieve candidate IDs of protein/gene mentions based on KBs. For the ambiguous mentions with multiple candidate IDs, some heuristic rules are developed for disambiguating them and uniquely assigning an ID. Since Sheng et al. [8] and Kaewphane et al. [5] use heuristic rules rather than disambiguation models to normalize candidate IDs, their approach achieves a relatively low precision and recall on the PNEN subtask. As can be seen from Table 3, our normalization approach outperforms the above related approaches and achieves a state-of-the-art result (0.408 micro-averaged $F1$-score). We attribute this to the validity of our disambiguation model, which accurately models the important context representation through the attention and gating mechanisms.

5 Conclusion

This paper develops a pipeline identification system for protein/gene entity recognition and normalization. On the one hand, we leverage domain knowledge from biomedical KBs by a BLSTM-CRF model for PNER. The n-gram boolean knowledge features are designed and explored, meanwhile, language model feature is also introduced to realize the feature transfer of common sense information. On the other hand, we propose a dual attention-based Convolutional Neural Network model for PNEN and investigate the effect of the dual attention mechanism and gating mechanism. Experimental results on the BioCreative VI Bio-ID corpus show that our system achieves the new state-of-the-art performance on both PNER and PNEN.

In the future work, we will explore the usage of joint learning of the PNER and PNEN to bridge the gap between them.

Acknowledgments. This work was supported by the grants of the Ministry of education of Humanities and Social Science project (No. 17YJA740076) and the National Natural Science Foundation of China (No. 61772109). Comments from the audience of CLSW2019 and the reviewers are also acknowledged.

References

1. Lin, Y., Liu, Z., Sun, M.: Neural relation extraction with multi-lingual attention. Proc. Assoc. Comput. Linguist. **1**, 34–43 (2017)
2. Rudolf, K., Ondrej, B., Jan, K.: Knowledge base completion: baselines strike back. In: Proceedings of the Association for Computational Linguistics, pp. 69–74 (2017)
3. Arighi, C., et al.: Bio-ID track overview. In: Proceedings of BioCreative Workshop, pp. 482–376 (2017)
4. Sheikhshab, G., Starks, E., Karsan, A., Sarkar, A., Birol, I.: Graph-based semi-supervised gene mention tagging. In: Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pp. 27–35 (2016)
5. Kaewphan, S., Mehryary, F., Hakala, K., Salakoski, T., Ginter, F.: TurkuNLP entry for interactive Bio-ID assignment. In: Proceedings of the BioCreative VI Workshop, pp. 32–35 (2017)
6. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354 (2016)
7. Chiu, J., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguist. **4**, 357–370 (2016)
8. Sheng, E., Miller, S., Ambite, J., Natarajan, P: A neural named entity recognition approach to biological entity identification. In: Proceedings of the BioCreative VI Workshop, pp. 24–27 (2017)
9. Devlin, J., Chang, M., Lee, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Apweiler, R., et al.: UniProt: the universal protein knowledgebase. Nucl. Acids Res. **32** (suppl_1), D115–D119 (2004)
11. Edgar, R., Domrachev, M., Lash, A.: Gene expression omnibus: NCBI gene expression and hybridization array data repository. Nucl. Acids Res. **30**(1), 207–210 (2002)
12. Eshel, Y., Cohen, N., Radinsky, K., Markovitch, Y., Levy, O.: Named entity disambiguation for noisy text. arXiv preprint arXiv:1706.09147 (2017)
13. Ganea, O., Hofmann, T.: Deep joint entity disambiguation with local neural attention. arXiv preprint arXiv:1704.04920 (2017)
14. GENIA Tagger tool Homepage. https://omictools.com/genia-tagger-tool. Accessed 12 Aug 2019
15. Moen, S., Ananiadou, T.: Distributional semantics resources for biomedical text processing. In: Proceedings of the 5th International Symposium on Languages in Biology and Medicine, pp. 39–43 (2013)
16. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)