ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240611

Empowering Biomedical Named Entity Recognition Through Multi-Tagger Collaboration

Jin Zhao[a], Jian Xie[a], Tinghui Zhu[a], Qian Guo[b], Zhixu Li[a,*] and Yanghua Xiao[a,]**

[a]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[b] Department of Computer Science and Engineering, East China University of Science and Technology

Abstract. Biomedical Named Entity Recognition (BioNER) plays a crucial role in automatically identifying specific categories of entities from biomedical texts. Currently, region-based methods have shown promising performance in BioNER. However, existing paradigms in the region-based methods suffer from inherent limitations, including the generation of negative samples, and the ignorance of token dependencies. To overcome these limitations, we propose a new paradigm, implemented as Token Cascade Tagger (TCT), which combines span identification and category classification. The TCT utilizes category information to enhance the correlation between the heads and tails of entities, effectively reducing the generation of negative samples. Additionally, we introduce a Token Dependency Tagger (TDT) that captures token dependencies within entity spans by identifying the longest span in a sentence. The TDT filters out incorrect spans and further improves the accuracy of span detection obtained from the TCT. Furthermore, we employ a multi-task learning framework to optimize both the TCT and TDT, leading to superior performance in BioNER. Extensive experiments on publicly available biomedical datasets demonstrate our method outperforms the previous state-of-the-art methods, achieving 92.44%, 92.54%, and 81.26% on NCBI-Disease, BC5CDR, and GENIA, respectively, in terms of F1 score.

1 Introduction

Biomedical Named Entity Recognition (BioNER) aims to automatically identify specific categories of entities from biomedical texts. As shown in the example in Figure 1, the sentence " *...NF-IL6 gene in U937 cells.*" contains three entities, i.e., " *NF-IL6* ", " *NF-IL6 gene* ", and " *U937 cells* ". BioNER is widely applied in bioinformatics, medical research, and clinical decision support. It aids in constructing biomedical knowledge graphs, supporting drug discovery, gene research, and expediting disease diagnosis[28, 4].

Recently, region-based methods have been proposed and have achieved promising performance on bioNER [25, 6, 23]. There are two primitive operations in region-based methods: span identification and categorization. According to the execution order of these two operations, the existing region-based methods can be formulated into two paradigms as illustrated in Figure 1. The first paradigm (referred to as P_1) first enumerates all possible candidate entity spans and then classifies spans into predefined categories [13, 16], which can be formulated as $f(h,t) \rightarrow c$. The second paradigm (referred to as P_2) first predicts the categories mentioned in the sentence and then identifies all the corresponding entity spans of these categories [19, 22], which can be expressed as $f(c) \rightarrow (h,t)$.

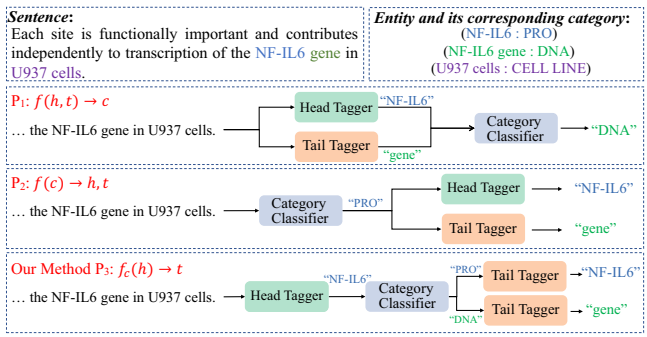

Figure 1. Paradigm comparison of region-based methods for Biomedical NER, where h and t denotes the head and tail of an entity respectively, and c denotes the category of an entity. The dotted arrow indicates the dependencies between triple elements.

Despite their success in bioNER, both paradigms exhibit inherent shortcomings. Paradigm P_1 inevitably generates a large number of negative samples due to the nature of exhaustive enumeration, hurting the recognition rate and incurring high computation costs. Additionally, it has limitations in detecting entities that exceed the predetermined maximum span length. While paradigm P_2 overcomes the shortcomings of P_1, it also suffers from two drawbacks. *Drawback 1* stems from predicting the entity's head and tail independently, leading to the generation of a substantial number of negative samples. *Drawback 2* relates to disregarding token dependencies within the entity span, resulting in numerous false positive samples.

The above analysis suggests that there exists another potential paradigm. We notice that span detection could be further divided into two more primitive operations: head index identification and tail index identification. Furthermore, these two operations could be interleaved with category identification to form a more flexible paradigm (referred to as P_3), expressed as $f_c(h) \rightarrow t$. We novelly instantiate paradigm P_3 as the token cascade tagger (TCT), which first predicts the heads(h) of all possible entities in a given sentence. Subsequently, for each head, it determines the corresponding tails (t) of the entity within the category c. By utilizing the category information as an additional signal to determine the corresponding tail relative to the head of each entity, the *Drawback 1* can be significantly alleviated, given that the category plays a crucial role in identifying entity spans, as illustrated in Example 1.

* Corresponding Author. Email: zhixuli@fudan.edu.cn.
** Corresponding Author. Email: shawyh@fudan.edu.cn.

Example 1. *In the case of the entity "IL-2R alpha gene", TCT initially predicts "IL-2R" as its head and subsequently identifies "alpha" as the corresponding tail within the "Protein" category. Likewise, if the category changes to "DNA", the corresponding tail would be "gene".*

To address *drawback 2*, we design a token dependency tagger (TDT) that captures token dependencies within entity spans by identifying the longest span among all possible entities in a sentence. Furthermore, the TDT filters out incorrect spans by determining if each entity span identified by the TCT falls within the longest span determined by the TDT, thereby improving the detection accuracy of spans obtained from the TCT. Additionally, we employ a multi-task learning framework to jointly optimize the TCT and the TDT, leading to superior performance. The main contributions of this paper are as follows:

- We introduce a novel paradigm for bioNER and implement it as a token cascade tagger. The token cascade tagger utilizes categories as supervisory signals to enforce constraints on the correlation between the head and tail of entities, effectively mitigating the generation of negative samples caused by separate predictions of the head and tail of an entity.
- We propose a token dependency tagger that identifies the longest span among all potential entities, capturing the dependencies between tokens within the entity span and effectively addressing the ignorance of token dependencies. Furthermore, we apply a multi-task learning framework that optimizes the token cascade tagger and the token dependency tagger, aiming to improve the performance of bioNER.
- We conduct extensive experiments on publicly available biomedical datasets. The experimental results show that our method outperforms the previous state-of-the-art methods, achieving 92.44%, 92.54%, and 81.26% on NCBI-Disease, BC5CDR, and GENIA, respectively, in terms of F1 score.

2 Related Works

BioNER has attracted considerable attention for its ability to improve downstream tasks. Prior studies [21, 5] employ a sequence labeling to identify biomedical entities, but these methods are prone to error propagation. To address these challenges, other studies [14, 23] employ the seq2seq model to directly generate various entities from the text. However, they may suffer from the decoding efficiency problem and exposure bias. Alternative studies [15, 24, 8] use various strategies to represent tokens and construct graphs or transition actions to represent all entities in a sentence. However, these methods suffer from spurious structure and structural ambiguity during inference.

Currently, region-based methods have achieved state-of-the-art performance and attracted much attention. Some approaches [13, 29, 16] first locate candidate entity spans from a text and then classify the candidate entity span into predefined categories. However, these approaches are subjected to maximal span lengths and lead to considerable computation costs due to their enumeration nature. On the other hand, other approaches [22, 7, 27] first enumerate the entity categories and then locate the candidate entity span. However, they detect the entity spans and categories separately and independently, which leads to error propagation.

In contrast to the existing region-based methods, we propose a new paradigm, implemented as Token Cascade Tagger, which utilizes category information to enhance the correlation between the heads and tails of entities, effectively reducing the generation of negative samples. Additionally, we introduce the Token Dependency Tagger, designed to recognize token dependencies within entity spans, thus enhancing span detection accuracy. Consequently, our method outperforms existing baseline methods, establishing a more effective means of identifying bioNER.

3 Problem Formulation

Given a sentence $S = (w_1, w_2, ..., w_n)$ with n tokens and k predefined categories $C = \{c_1, c_2, ..., c_k\}$, the task of our method is to recognize all entity spans and their corresponding categories, i.e., $E = \{(h_i, t_i, c_i)\}_{i=1}^{M}$, in S, where M is the number of entities. h_i, t_i are the head and tail index of the i-th entity consisting of several consecutive tokens, i.e., $entity.span = w_{h_i:t_i}$, where $w_{h_i:t_i} (h_i, t_i \in [1, n])$ denotes the concatenation of w_{h_i} to w_{t_i}.

4 Model Architecture

Figure 2 illustrates the overall workflow of our framework. Our framework adopts a multi-task learning approach, comprising an encoder, a token dependency tagger, and a token cascade tagger. The encoder generates a shared vector representation for each word in sentences, which is utilized by both TCT and TDT. TCT identifies the span and category of all possible entities in a sentence, while TDT identifies the longest spans among all possible entities and serves to filter out incorrect entity spans detected by TCT.

Take the sentence in Figure 2 as an example, we show how our framework identifies entities in a sentence. The token dependency tagger tags the longest spans of entities in the sentence, like "NF-IL6 gene" and "U937 cells" are tagged as entity spans. The token cascade tagger tags the head(s) in the sentence using the head tagger, such as "NF-IL6" and "U937". The category-specific tail tagger recognizes possible tail(s) under the category-specific; or returns no head, indicating that there is no entity with the given tail and category. Specifically, for the head "NF-IL6", the PRO-specific tail tagger can find a tail "NF-IL6", indicating the existence of an entity with the span "NF-IL6" and the category "PRO". while the CELL LINE-specific tail tagger fails to find a suitable tail, indicating the absence of an entity for that category. The same decoding process is applied for the head "U937". Finally, entities are recognized by combining the results obtained from both the token dependency tagger and the token cascade tagger.

4.1 Encoder

The encoder converts the input sentence into a fixed-length vector and effectively captures both the semantic and syntactic information. The vector is then passed to subsequent modules, i.e., TCT and TDT. Considering that pre-trained language models trained on general corpora are insensitive to biomedical domain knowledge, we employ multi-granularity textual features to represent sentences. To capture the multi-granularity features of the input sentence, we employ various levels of embeddings to represent the semantics of tokens, including, character embeddings, word embeddings, and contextual embeddings. For the character embeddings x^c, we apply CNN [26] to extract the character features of tokens. For the word embeddings x^w, we employ pre-trained word embeddings to encode the semantics features of tokens. For contextual embeddings x^p, we employ a pre-trained language model to represent the contextual features of tokens, and apply max pooling to produce word representations based

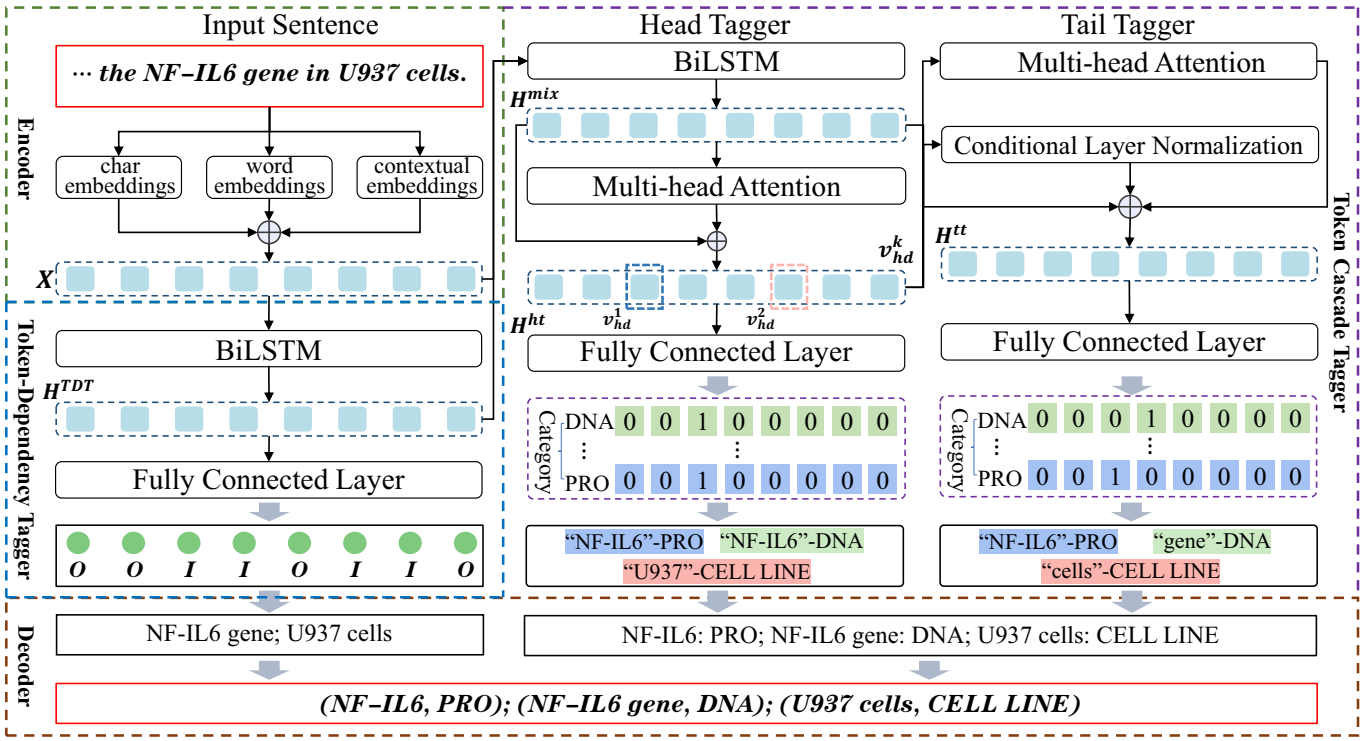

Figure 2. The architecture of our framework.

on the word piece representations. The sentence embeddings are obtained as follows,

$$X = [x_1, x_2, ..., x_n] \tag{1}$$

, where, $x_i = [x_i^w; x_i^c; x_i^p]$, $i \in [1, n]$, [;] means additive concatenation operator.

4.2 Token-Dependency Tagger

This module aims to identify the longest span of entities and to filter the spans predicted by TCT. Following the method suggested by [19], we redefine two labels, "I" and "O", and employ the sequence labeling method to identify the longest span of entities, i.e., tokens within the entity span are all assigned "I" and tokens beyond the entity span are assigned "O". As shown in Figure 2, "NF-IL6 gene" and "U937 cells" are identified as the longest span of entities.

Specifically, the sentence representation $\boldsymbol{X}$ is fed to a BiLSTM to obtain an effective sentence feature representation for token cascade tagger, which is expressed as follows,

$$\boldsymbol{H}^{TDT} = BiLSTM[\overrightarrow{\boldsymbol{X}}; \overleftarrow{\boldsymbol{X}}] \tag{2}$$

Subsequently, the sentence feature representation is passed through a fully connected layer to determine if the token is part of an entity. We calculate the probability of i-th token in the input sequence as a part of an entity:

$$p_i^{tsm} = \sigma(\boldsymbol{W}_{tsm}\boldsymbol{h}_i + \boldsymbol{b}_{tsm}) \tag{3}$$

, where $\boldsymbol{h}_i$ denotes the embeddings of i-th index in $\boldsymbol{H}^{TDT}$, i.e., $\boldsymbol{h}_i = \boldsymbol{H}^{TDT}[i]$. $\boldsymbol{W}_{tsm}$ denotes the trainable weight, $\boldsymbol{b}_{tsm}$ represents the bias and σ represents the sigmoid activation function. p_i^{tsm} indicates the score of recognizing the i-th token in the input sequence as a part of an entity.

TDT is optimized as follows:

$$\ell_{tsm} = -\sum_{i=1}^{n}[p_i^{tsm\_label} log p_i^{tsm} + (1 - p_i^{tsm\_label}) log(1 - p_i^{tsm})] \tag{4}$$

, where $p_i^{tsm\_label}$ is "I" if the i-th token is determined to be in the entity span or "O" otherwise.

4.3 Token Cascade Tagger

This module aims to identify the span of all possible entities and their corresponding categories in a sentence. The module tags the head(s) in the sentence using the head tagger, such as "NF-IL6" and "U937". The category-specific tail tagger recognizes possible tail(s) under the category-specific; or returns no head, indicating that there is no entity with the given tail and category.

Specifically, for the head "NF-IL6", the PRO-specific tail tagger can find a tail "NF-IL6", indicating the existence of an entity with the span "NF-IL6" and the category "PRO". while the CELL LINE-specific tail tagger fails to find a suitable tail, indicating the absence of an entity for that category. The same decoding process is applied for the head "U937". As shown in Figure 2, "NF-IL6": "PRO", "NF-IL6 gene": "DNA", and "U937 cells": "CELL LINE" are identified, The former denotes the entity span and the latter denotes the entity category. Each tagger is described in detail next.

4.3.1 Head tagger

The head tagger aims to recognize the head index of all possible entities from the sentence. We first apply BiLSTM to generate a sentence

representation for head tagger, denoted as follows,

$$H^{mix} = BiLSTM[\overrightarrow{\boldsymbol{H}^t}; \overleftarrow{\boldsymbol{H}^t}] \tag{5}$$

, where $\boldsymbol{H}^t = [\boldsymbol{X}; \boldsymbol{H}^{TDT}]$. Considering that different tokens in a sentence play different roles in entity recognition, we employ the multi-head attention to generate auxiliary features for the head index and tail index of an entity span, i.e., $\boldsymbol{H}^a$, $\boldsymbol{T}^a$, respectively. For more details of multi-head attention, refer to [17]. Note that in our setting, $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{H}^{mix}$, $1/\sqrt{d_k}$ is the scaling factor.

Then, we concatenate the contextual representation $\boldsymbol{H}^{mix}$ and the auxiliary features of entity's head index $\boldsymbol{H}^a$ as $\boldsymbol{U}^{head} = [\boldsymbol{H}^{mix}; \boldsymbol{H}^a]$. The score of i-th token is calculated according to the head index of an entity with a specific category $g \in G$.

$$p_i^{head} = \sigma(\boldsymbol{W}_{head}\boldsymbol{x}_i + \boldsymbol{b}_{head}) \tag{6}$$

, where p_i^{head} denotes the probability of identifying the i-th token in the input sequence as the head index of an entity. $\boldsymbol{x}_i$ is the encoded representation of the i-th token in the input sequence, i.e., $\boldsymbol{x_i} = \boldsymbol{U}^{head}[i]$, $\boldsymbol{W}_{head}$ is the trainable weight, and $\boldsymbol{b}_{head}$ denotes the bias.

The optimization of the head tagger is optimized below:

$$\ell_{head} = -\sum_{i=1}^{n}[p_i^{head\_label} log p_i^{head} + (1 - p_i^{head\_label}) log(1 - p_i^{head})] \tag{7}$$

, where $p_i^{head\_label}$ is 1 if the i-th token is determined to be the head token or 0 otherwise.

4.3.2 Tail tagger

The tail tagger aims to simultaneously recognize the tail index and the related categories with respect to the tagged head index by the head tagger. As Figure 2 shows, it is composed of a set of category-specific tail taggers for all possible categories. All tail taggers will identify the corresponding tail index for each tagged head index at the same time. In order to enhance the correlation between the head and tail index of an entity, we apply Conditional Layer Normalization (CLN) [20] to generate a normalized embedding. The core idea of CLN is to insert conditional information, so that the normalized embedding is closely related to the conditional information. In this way, the model is able to better capture the features of the entity span and improve the accuracy of entity recognition. It is expressed as $\boldsymbol{CoR}$:

$$\boldsymbol{CoR} = CLN(\boldsymbol{H}^{mix}, \boldsymbol{H}^{mix}) \tag{8}$$

We can obtain a new hidden vector for the tail tagger, which is represented as $\boldsymbol{H}^{tt} = [\boldsymbol{H}^{mix}; \boldsymbol{T}^a; \boldsymbol{CoR}]$. Then, the hidden vector is fed into the fully connected layer to judge whether the token is a tail index of an entity. We calculate the probability of i-th token in the input sequence as a tail index of an entity:

$$p_i^{tail} = \sigma(\boldsymbol{W}_{tail}(\boldsymbol{x}_i + \boldsymbol{v}_{hd}^k) + \boldsymbol{b}_{tail}) \tag{9}$$

, where $\boldsymbol{W}_{tail}$ is a trainable weight, and $\boldsymbol{b}_{tail}$ denotes the bias. p_i^{tail} represents the probability of identifying the i-th index as the tail index of an entity, and $\boldsymbol{v}_{hd}^k$ represents the embedding of the k-th index in $\boldsymbol{U}^{\boldsymbol{head}}$. For each head index detected in the head tagger, we iteratively apply the same decoding process on it. The tail tagger is optimized as follows:

$$\ell_{tail} = -\sum_{i=1}^{n}[p_i^{tail\_label} log p_i^{tail} + (1 - p_i^{tail\_label}) log(1 - p_i^{tail})] \tag{10}$$

, where $p_i^{tail\_label}$ is 1 if the i-th token is determined to be the tail token or 0 otherwise.

4.4 Training and Inference

During the training phase, we apply multi-task loss to optimize TCT and TDT simultaneously:

$$\ell_{total} = \ell_{tsm} + \ell_{head} + \ell_{tail}, \tag{11}$$

During the inference phase, given an input sentence, we initially obtain the probabilities of the head and tail index, p_i^{head} and p_i^{tail}, respectively, as predicted by TCT. A span is recognized as an entity with a specific category if ($p_i^{head} \times p_i^{tail}$) exceeds the fine-tuned threshold λ on the development set, and if and only if all the tokens within the span are tagged "I" by TDT. Otherwise, the span is not an entity. For instance, the spans, "NF-IL6" and "NF-IL6 gene", identified by TCT, are within the longest span, "NF-IL6 gene", identified by TDT, confirming the correctness of "NF-IL6" and "NF-IL6 gene". The final output format is "NF-IL6": "PRO", "NF-IL6 gene": "DNA", where "PRO" and "DNA" indicate the entity category as predicted by TCT.

5 Experiments Settings

In this section, we first present the dataset, then describe the evaluation metrics and implementation details, and finally list the baseline methods.

5.1 Datasets

We select three English corpora from the biomedical field, namely NCBI-Disease, BC5CDR, and GENIA, to evaluate the effectiveness of our method in identifying bioNER.

- NCBI-Disease is a disease corpus annotated by 14 experts specializing in the field of diseases. It consists of 6871 disease names annotated from 793 abstracts of PubMed papers.
- BC5CDR is specifically created to facilitate the identification of disease and chemical names. It consists of 1500 PubMed abstracts annotated with 4409 chemical names and 5818 disease names.
- GENIA is a specialized biomedical dataset designed specifically for nested NER. The dataset encompasses five distinct categories of entities, i.e., "PROTEIN", "CELL LINE", "DNA", "CELL TYPE", "RNA". It includes a total of 54,935 annotated entities, with 11,359 exhibiting nested structures.

To maintain experimental consistency, we follow the experimental setups outlined in KaNER [2] for the NCBI-Disease, BC5CDR and GENIA datasets. Under this configuration, we split the dataset into the training dataset, development dataset, and testing dataset with a ratio of 8.1:0.9:1. The development dataset is used to fine-tune the hyperparameters.

5.2 Evaluation Metric and Implementation Details

To ensure a comprehensive evaluation of our method, we report three metrics, i.e., Precision (P), Recall (R), and F1 scores (F1). These metrics are consistent with existing baseline methods, such as KaNER [2], UGF [23], and so on. It is important to note that entities are only considered correct if and only if their span and category match the golden entity.

Table 1. Comparison results with the baseline methods.

Model	NCBI-Disease			BC5CDR			GENIA		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
AutoNER [12]	79.42	71.98	75.52	88.96	81.00	84.80	–	–	–
DEM [13]	–	–	–	–	–	–	96.4	66.8	78.4
Seq2Seq [14]	87.65	89.21	88.42	88.43	88.93	88.70	–	–	76.44
MTM-CW [18]	85.86	86.42	86.14	89.10	88.47	88.78	70.91	76.34	73.52
LM-NER [5]	88.22	91.25	89.71	88.10	88.71	88.40	–	–	–
BENSC [16]	85.80	84.80	85.30	83.80	83.90	83.90	79.20	77.40	78.30
UGF [23]	89.32	90.59	89.95	90.58	90.86	90.72	78.87	79.60	79.23
BNER [15]	89.67	90.43	90.04	–	–	–	–	–	–
MHSA [22]	–	–	–	–	–	–	80.30	78.90	79.10
BidH [21]	87.01	88.76	87.88	89.76	90.56	90.16	73.6	78.0	75.7
LLCP [9]	88.32	89.21	88.76	89.43	91.02	90.22	78.39	78.50	78.44
BUCP [24]	90.55	91.57	91.06	91.08	91.56	91.31	78.08	78.26	78.16
BANM [29]	-	-	-	-	-	-	75.90	73.60	74.70
NRL [1]	88.07	89.17	88.61	89.47	91.19	90.32	–	–	–
AIONER [10]	–	–	89.55	–	–	89.40	–	–	–
TEDC [8]	85.23	75.17	79.88	89.16	84.96	87.01	–	–	–
KaNER [2]	90.43	92.07	91.24	91.73	90.95	91.34	79.47	78.51	78.99
Our method	**91.36**	**93.55**	**92.44**	**92.20**	**92.88**	**92.54**	**80.87**	**81.66**	**81.26**

Table 2. Comparison results with ChatGPT.

Model		NCBI-Disease			BC5CDR			GENIA		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
ChatGPT	1-shot	21.82	56.14	31.43	16.48	61.25	25.98	57.54	33.78	42.57
	3-shot	24.91	64.87	35.99	19.73	60.97	29.81	59.12	34.37	43.47
	5-shot	28.28	65.94	39.58	19.89	59.97	29.87	61.01	34.89	44.39
Our Method		**91.36**	**93.55**	**92.44**	**92.20**	**92.88**	**92.54**	**80.87**	**81.66**	**81.26**

We implement our method using PyTorch, a widely used neural network framework. For character embeddings, we employ a CNN with kernel sizes of 2, 3, and 4 to capture character-level features for each token. As for word embeddings, we utilize a pre-trained word embedding model trained on MEDLINE abstracts [3], which is applied across all datasets. For contextual embeddings, we apply $SciBERT_{base}$ for all datasets. To avoid overfitting, we apply a dropout rate of 0.5 for word embeddings and 0.3 for character embeddings. The dimensions for character embeddings, word embeddings, and contextual embeddings are set as 50, 300, and 768, respectively. AdamW is selected as the optimizer, and the learning rate is set from $1e^{-5}$ to $5e^{-5}$. Moreover, the threshold is set as 0.3.

5.3 Baseline Methods

We select four categories of methods as baseline methods for comparison, as follows. Label-based methods, which leverage label relationships for entity decoding, include AutoNER [12], MTM-CW [18], LM-NER [5], BidH [21]. Seq2seq-based methods, utilizing the seq2seq model for direct entity generation, comprise Seq2Seq [14], UGF [23], KaNER [2]. Region-based methods, focusing on span detection and classification, encompass DEM [13], MHSA [22], LLCP [9], BUCP [24]. Other-based methods, employing diverse strategies for performance enhancement, include BNER [15], NRL [1], AIONER [10], TEDC [8].

6 Experimental Results and Analysis

6.1 Comparison with Baseline Methods

We conduct an evaluation on three bioNER datasets to validate the effectiveness of our method in identifying biomedical entities. The experimental results are presented in Table 1. In comparison to state-of-the-art baseline methods, our method achieves superior performance across all three bioNER datasets. Specifically, our method outperforms KaNER [2] by 1.20%, 1.20% and outperforms UGF [23] by 2.03% absolute F1 on NCBI-Disease, BC5CDR, and GENIA datasets, respectively. This robust performance substantiates the effectiveness of our proposed method for the bioNER.

We attribute the reason to the reformulation of the relationship between span detection and entity categories as a mapping function. This reformulation significantly reduces the occurrence of erroneous samples resulting from the amalgamation of head and tail within spans. Furthermore, this reformulation employs a one-to-one decoding strategy, in which each head precisely corresponds to the tail of a category-specific entity span. The one-to-one decoding strategies operate independently, mitigating the accumulation of errors.

6.2 Comparison with ChatGPT

In order to explore the gap in performance between ChatGPT and our method, we conduct experiments on the NCBI-Disease, BC5CDR, and GENIA datasets, respectively. Since ChatGPT's capabilities are accessed via instructions, we design a specialized task instruction

Table 3. Performance on different types of entities.

Category	KaNER [2]			ChatGPT [11]			Our method		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
DNA	76.25	77.08	76.66	53.77	49.07	51.31	76.04	77.51	**76.77**
RNA	84.36	83.51	83.93	35.90	80.77	49.71	89.02	90.87	**89.94**
PROTEIN	80.21	81.23	80.72	54.39	61.40	57.68	81.45	81.34	**81.39**
CELL LINE	78.55	78.22	78.38	50.21	54.44	52.24	78.56	80.16	**79.35**
CELL TYPE	78.56	78.64	78.60	20.95	68.64	32.10	81.24	81.27	**81.25**

Table 4. Performance on span detection

Model	NCBI-Disease			BC5CDR			GENIA		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
MTM-CW [18]	87.23	88.12	87.67	90.67	90.12	90.39	72.45	77.21	74.75
UGF [23]	91.45	93.04	92.24	92.45	93.01	92.73	80.12	81.56	80.83
BidH [21]	88.76	89.56	89.16	91.56	90.78	91.17	75.12	79.98	77.47
LLCP [9]	89.89	91.34	90.61	90.78	93.01	91.88	79.87	80.56	80.21
BUCP [24]	92.14	93.06	92.60	92.78	92.90	92.84	79.12	79.89	79.50
KaNER [2]	91.56	92.89	92.22	93.67	91.58	92.61	81.23	80.78	81.00
Our method	93.24	95.45	94.33	93.56	94.12	93.84	82.45	83.13	82.79

that aligns with ChatGPT's input format. Figure 3 shows an example of instruction applied on the GENIA dataset. The instruction consists of four parts: *(1) Task Description*, marked by an orange box, prompts ChatGPT to identify all the entities in the given sentence. *(2) Options*, marked by a green box, prompts ChatGPT to generate a set of possible outputs for a given input.*(3) Few-shot Demonstrations*, marked by a purple box, provides ChatGPT with few-shot examples for reference.*(4) Output Description*, marked by a red box, specifies the format of the ChatGPT's output.

The comparative performance between ChatGPT and our method is presented in Table 2. As illustrated in the table 2, our method consistently outperforms ChatGPT across the NCBI-Disease, BC5CDR, and GENIA datasets. Several factors may contribute to this discrepancy: (1) Discrepancies in sample labeling, which may not fully correspond with ChatGPT's interpretations. For instance, entities like "H2.0" are recognized by ChatGPT but are not labeled in the provided data. (2) The entity boundaries identified by ChatGPT may not precisely match those of the manual annotations. Additionally, it is worth noting that increasing the number of reference examples does not lead to a significant improvement in ChatGPT's performance.

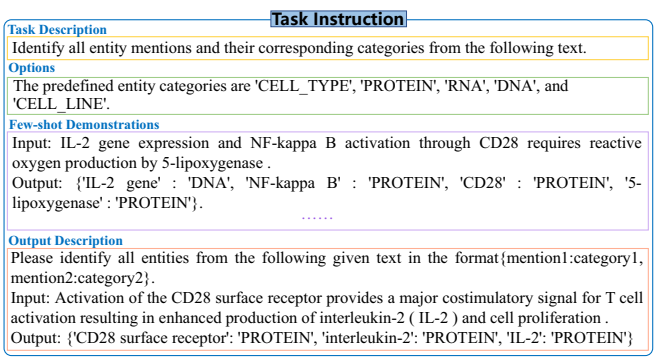

Figure 3. An example of the task instruction of ChatGPT.

6.3 *Performance of Various Categories of Entities*

To assess the recognition performance of our method across various entity categories, we conduct a comparison experiment with KaNER [2], ChatGPT [11] on the GENIA dataset. The results are shown in Table 3. Table 3 demonstrates that our method outperforms KaNER [2] and ChatGPT [11] in recognizing diverse entity categories. This superiority may stem from the entity category as a priori information that constrains the detection of entity spans, rather than as a supervised signal for span classification. Notably, our method demonstrates exceptional performance in identifying entities within the RNA category, likely due to distinctive markers in the dataset, such as suffixes like "mRNA" or "RNA". However, all methods exhibit poor performance in identifying entities in the DNA category. This can be attributed to the significant ambiguity in the dataset concerning DNA.

Table 5. Ablation study on NCBI-Disease, BC5CDR and GENIA

Model	NCBI-Disease	BC5CDR	GENIA
Our Method	92.44	92.54	81.26
w/o Context embeddings	91.08	90.96	80.06
w/o Word embeddings	90.73	90.53	79.83
w/o Char embeddings	90.49	90.24	79.71
w/o CLN	88.60	88.39	78.18
w/o Multi-head attention	88.38	88.18	78.03
w/o TDT	87.14	87.08	77.29

6.4 *Performance of span Detection*

To demonstrate the effectiveness of our model in detecting entity span, we conduct experiments on three bioNER datasets. The experimental results are shown in Table 4. Table 4 demonstrates that our method achieves superior performance in span detection, which indicates that our method is more effective than existing methods in span detection.

Table 6. Three examples for case study.

Example 1: Here we demonstrate that the [[c–myb proto-oncogene product]$_{protein}$]$_{protein}$, which is itself a [[DNA-binding]$_{protein}$]$_{protein}$ and [[transcriptional transactivator]$_{protein}$]$_{protein}$, can interact synergistically with [[Z]$_{protein}$]$_{protein}$ in activating the [[BMRF1 promoter]$_{DNA}$]$_{DNA}$ in [[Jurkat cells]$_{cell\_line}$]$_{cell\_line}$ (a [[T-cell line]$_{cell\_line}$]$_{cell_line}$) or [[Raji cells]$_{cell\_line}$]$_{cell\_line}$ (an [[EBV-positive B - cell]$_{cell\_line}$]$_{cell\_line}$), whereas the [[c-myb gene product]$_{protein}$]$_{protein}$ by itself has little effect.
Example 2: [^{2}[^{2}[^{1}[^{1}Tax^1]$_{cell_line}$^{1}]$_{cell\_line}$ expressing T cell lines^2]$_{cell\_line}$^{2}]$_{cell_line}$ contained a constitutive level of [[NF-kappa B]$_{protein}$]$_{protein}$ binding activity, detectable by mobility shift assay and uv cross-linking using a [^{2}[^{2}palindromic[^{1}[^{1}NF-kappa B^1]$_{DNA}$^{1}]$_{DNA}$*probe*^{2}]$_{DNA}$^{2}]$_{DNA}$ homologous to the [[interferon]$_{DNA}$]$_{DNA}$ beta [[PRDII site]$_{DNA}$]$_{DNA}$.
Example 3: Nuclear extracts assayed for the presence of [[ISRE binding factors]$_{DNA}$]$_{protein}$ by electrophoretic mobility shift assays show that [[ISGF3]$_{protein}$]$_{protein}$ is induced by [[IFN-alpha]$_{protein}$]$_{protein}$ within 6 h from undetectable basal levels in untreated U937 cells [[U937 cells]$_{cell\_line}$]$_{cell\_line}$.

We attribute the superior performance to the following reasons. Span detection is divided into head identification and tail identification. Initially, we predict all possible heads within a sentence. Subsequently, for each predicted head, we simultaneously identify all possible categories and the corresponding tails. Our approach not only utilizes categories to locate an entity's span but also considers the mapping of the head and tail of an entity within a particular entity category.

6.5 *Ablation Study*

To explore the contribution of various components of the method to the task, we conduct an ablation experiment on three datasets, NCBI-Disease, BC5CDR, and GENIA, respectively. The results of the ablation experiment are shown in Table 5. We observe three components that substantially contribute to the performance of the NER task, Context embeddings, CLN, and TDT. As we observed, on the NCBI-Disease, BC5CDR, and GENIA datasets, the ablation of the pre-trained language model led to a decrease in F1 scores by 1.36%, 1.58%, and 1.22%, respectively. the ablation of CLN led to a decrease in F1 scores by 1.89%, 1.85%, and 1.53%, respectively. For the ablation of TDT, the F1 scores decreased by 1.24%, 1.10, and 0.89%, respectively.

We note that TDT has a limited impact on the performance improvement of the task, probably because it reduces the recall while improving the correctness. In addition, components, such as word embeddings, char embeddings, and multi-head attention, contribute less to the performance of tasks, probably because the pre-trained language model contains the information they represent.

6.6 *Case Study*

Table 6 presents three examples of our model's predictions. The first example illustrates that our model can correctly identify all entities from a long text with about ten entities. The second example highlights that our model can correctly distinguish various entities with nested strictures. We can observe that two nested entities are correctly identified. However, our model is prone to misidentifying entity categories due to deficiencies in semantic understanding, as shown in the third example. The entity category "protein" of the entity "ISRE binding factors" is mistaken for "DNA". It is worth mentioning that our model can accurately locate the entity spans, demonstrating the effectiveness of our model in entity span detection.

7 Conclusions and Future Work

BioNER aims to identify entities in the text within the biomedical field. In this paper, we propose a multi-tagger collaboration framework, which could greatly alleviate the existing drawbacks of the region-based methods. We conduct extensive experiments on publicly available biomedical datasets. The experimental results demonstrate that our method outperforms the previous state-of-the-art methods. Specifically, the ablation study highlights the substantial contributions of three key components: Context embeddings, CLN, and TDT, to the overall performance of the method. For further work, we will explore the correlation between entity category and entity span in vector space, and design a more generalized decoding scheme to eliminate threshold constraints and further improve the performance.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62072323), Shanghai Science and Technology Innovation Action Plan (No.22511104700)

References

[1] Z. Chai, H. Jin, S. Shi, S. Zhan, L. Zhuo, Y. Yang, and Q. Lian. Noise Reduction Learning Based on XLNet-CRF for Biomedical Named Entity Recognition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1):595–605, Jan. 2023. ISSN 1557-9964. doi: 10.1109/TCBB.2022.3157630.

[2] P. Chen, J. Wang, H. Lin, Y. Zhang, and Z. Yang. Knowledge adaptive multi-way matching network for biomedical named entity recognition via machine reading comprehension. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):2101–2111, 2023. doi: 10.1109/TCBB.2022.3233856.

[3] J. P. Chiu et al. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, July 2016. ISSN 2307-387X. doi: 10.1162/tacl_a_00104.

[4] Q. Guo, Y. Guo, and J. Zhao. Hrcl: Hierarchical relation contrastive learning for low-resource relation extraction. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024. doi: 10.1109/TNNLS.2024.3386611.

[5] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[6] J. Li, H. Fei, J. Liu, S. Wu, M. Zhang, C. Teng, D. Ji, and F. Li. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973, 2022.

[7] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages

5849–5859, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.519. URL https://aclanthology.org/2020.acl-main.519.

[8] T. Liang, C. Xia, Z. Zhao, Y. Jiang, Y. Yin, and P. S. Yu. Transferring from textual entailment to biomedical named entity recognition. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(4):2577–2586, 2023. doi: 10.1109/TCBB.2023.3236477.

[9] C. Lou, S. Yang, and K. Tu. Nested named entity recognition as latent lexicalized constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6183–6198, 2022.

[10] L. Luo, C.-H. Wei, P.-T. Lai, R. Leaman, Q. Chen, and Z. Lu. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. *Bioinformatics*, 39(5):btad310, 05 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad310. URL https://doi.org/10.1093/bioinformatics/btad310.

[11] OpenAI. Chatgpt, 2022. URL https://openai.com/blog/chatgpt.

[12] J. Shang, L. Liu, X. Gu, X. Ren, T. Ren, and J. Han. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1230. URL https://aclanthology.org/D18-1230.

[13] M. G. Sohrab and M. Miwa. Deep Exhaustive Model for Nested Named Entity Recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1309.

[14] J. Straková et al. Neural Architectures for Nested NER through Linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1527.

[15] C. Sun, Z. Yang, L. Wang, Y. Zhang, H. Lin, and J. Wang. Biomedical named entity recognition using BERT in the machine reading comprehension framework. *Journal of Biomedical Informatics*, 118:103799, June 2021. ISSN 1532-0464. doi: 10.1016/j.jbi.2021.103799.

[16] C. Tan, W. Qiu, M. Chen, R. Wang, and F. Huang. Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9016–9023, Apr. 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i05.6434.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] X. Wang, Y. Zhang, X. Ren, Y. Zhang, M. Zitnik, J. Shang, C. Langlotz, and J. Han. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, 35(10):1745–1752, 2019. doi: 10.1093/bioinformatics/bty869.

[19] Y. Wang, Y. Li, H. Tong, and Z. Zhu. HIT: Nested Named Entity Recognition via Head-Tail Pair and Token Interaction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6027–6036. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.486.

[20] Y. Wang, B. Yu, H. Zhu, T. Liu, N. Yu, and L. Sun. Discontinuous named entity recognition as maximal clique discovery. In *ACL*, 2021.

[21] W. Xu, W. Li, J. Guan, and S. Zhou. Bidh: A bidirectional hierarchical model for nested named entity recognition. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4600–4604, New York, NY, USA, 2022. Association for Computing Machinery. doi: 10.1145/3511808.3557554.

[22] Y. Xu, H. Huang, C. Feng, and Y. Hu. A supervised multi-head self-attention network for nested named entity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14185–14193, 2021. doi: 10.1609/aaai.v35i16.17669.

[23] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5808–5822. Association for Computational Linguistics, Aug. 2021. doi: 10.18653/v1/2021.acl-long.451.

[24] S. Yang and K. Tu. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2403–2416, 2022.

[25] Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. Nested Named Entity Recognition: A Survey. *ACM Transactions on Knowledge Discovery from Data*, page 3522593, Mar. 2022. ISSN 1556-4681, 1556-472X. doi: 10.1145/3522593.

[26] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[27] J. Zhao, Z. Li, Y. Xiao, J. Liang, and J. Liu. Htmapper: Bidirectional head-tail mapping for nested named entity recognition. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 3433–3443, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614919. URL https://doi.org/10.1145/3583780.3614919.

[28] J. Zhao, C. Liu, J. Liang, Z. Li, and Y. Xiao. A novel cascade instruction tuning method for biomedical ner. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11701–11705, 2024. doi: 10.1109/ICASSP48485.2024.10446885.

[29] C. Zheng, Y. Cai, J. Xu, H.-f. Leung, and G. Xu. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 357–366, 2019.