BMC Bioinformatics

RESEARCH Open Access

Enhancing biomedical named entity recognition with parallel boundary detection and category classification

Yu Wang[1*], Hanghang Tong[2], Ziye Zhu[3], Fengzhen Hou[1] and Yun Li[3]

*Correspondence:
wangyu@cpu.edu.cn

[1] School of Science, China Pharmaceutical University, Nanjing, China
[2] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA
[3] Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract

***Background*:** Named entity recognition is a fundamental task in natural language processing. Recognizing entities in biomedical text, known as the BioNER, is particularly crucial for cutting-edge applications. However, BioNER poses greater challenges compared to traditional NER due to (1) nested structures and (2) category correlations inherent in biomedical entities. Recently, various BioNER models have been developed based on region classification or large language models. Despite being successful, these models still struggle to balance handling nested structures and capturing category knowledge.

***Results*:** We present a novel parallel BioNER model, Bean, designed to address the unique properties of biomedical entities while achieving a reasonable balance between handling nested structures and incorporating category correlations. Extensive experiments on five public NER datasets, including four biomedical datasets, demonstrate that Bean achieves state-of-the-art performance.

***Conclusions*:** The proposed Bean is elaborately designed to achieve two key objectives of the BioNER task: clearly detecting entity boundaries and correctly classifying entity categories. It is the first BioNER model to handle nested structures and category correlations in parallel. We exploit head, tail, and contextualized features to efficiently detect entity boundaries via a triaffine model. To the best of our knowledge, we are the first to introduce a multi-label classification model for the BioNER task to extract entity category information without boundary guidance.

Keywords: Biomedical named entity recognition, Named entity recognition, Text mining, Natural language processing, Biomedical domain

Background

Named entity recognition (NER) is a fundamental task in natural language processing [1, 2], aimed at identifying noun phrases conveying key information in text. Although recent state-of-the-art large language models (LLMs) have prominently advanced natural language processing, fundamental tasks remain significant in specific domains. In particular, in the biomedical domain, recognizing entities from biomedical text, known as the BioNER task, is crucial for cutting-edge applications such as biological information

retrieval [3], medical knowledge base construction [4], and medical intelligent question answering [5].

Compared to named entities in the general domain, biomedical named entities present a greater challenge for recognition. Specifically, biomedical entities frequently exhibit the *entity nested structure*, where an entity often encompasses other entities while simultaneously being a component of other entities. Figure 1 illustrates this nested property with samples from the well-known biomedical GENIA corpus [6]. For instance, in the second sentence, "IL - 13 gene 5 ' flank region"(DNA) is a long entity that includes two internal entities, i.e., "IL - 13 gene"(DNA) and "IL - 13"(protein). Another distinctive property of biomedical entities is the correlation between entity categories, referred to as *entity category correlation*, which includes both co-occurrence and nested correlations. In the GENIA corpus, about 72% of the sentences show co-occurrence correlation, while over 18% exhibit nested correlation. As shown in Fig. 1, DNA and cell_line co-occur in both sentences. Additionally, DNA entities are more likely to contain protein entities but rarely contain cell_line entities. Notably, there is a notable same-category nested correlation among biomedical entities, where entities of the same category are nested within one another. Specifically, 52% of the nested pairs in the GENIA corpus belong to this mode. For example, protein-nested accounts for as high as 71%, while cell_line-nested is observed in only 2.6% of cases.

Early BioNER methods followed the paradigm of traditional NER, employing the sequence labeling model to effectively recognize flat entities [7–11]. Subsequent methods reframed BioNER as a region classification task to handle more complex scenarios involving nested biomedical entities [12, 13]. For instance, HIT [14] employs a biaffine-based head-tail detector to determine whether each token pair in the sentence is the boundary of an entity. While such region-based methods improve the handling of nested structures by prioritizing boundary detection, they often fail to fully utilize category information across the entire sentence. This major drawback severely limits their ability

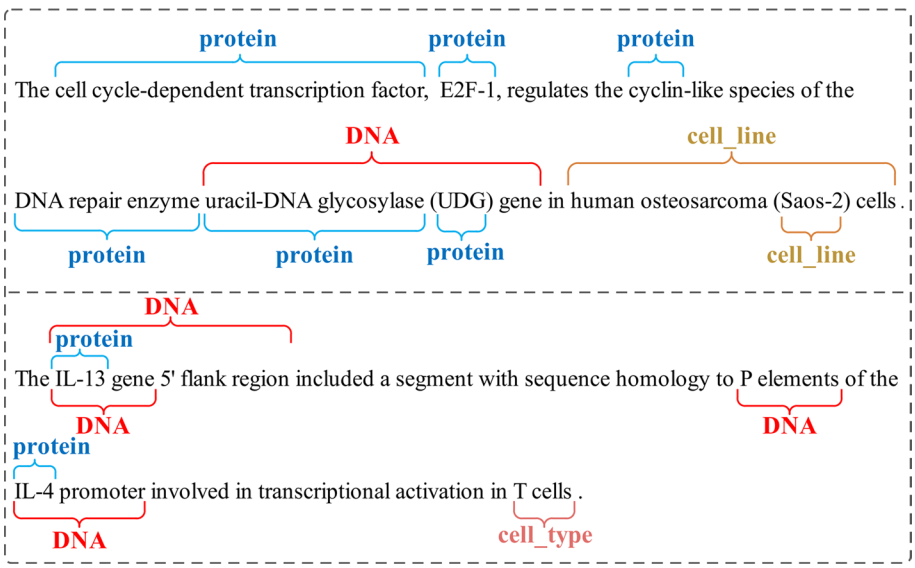

Fig. 1 Examples containing nested named entities from GENIA corpus

to manage the crucial property of entity category correlation. Recently, advancements in BioNER leveraged LLMs to better incorporate category information, adopting techniques such as instruction tuning and prompt engineering. Such as BioNER-LLaMA [15] successfully transforms BioNER task into a generation task and incorporates biomedical domain knowledge by training an instruction-following model with LLaMA as the backbone. Despite the potential of LLMs to address biomedical problems, they face constraints related to computational resources, data quality, and prompt design. Furthermore, since LLMs are primarily trained for sequence generation, LLM-based BioNER models naturally struggle with nested entities, particularly those within the same category.

In this paper, we focus on the entity nested structure and entity category correlation inherent in biomedical entities, and propose a novel parallel BioNER model named Bean (Boundary detection and category classification in parallel). Our proposed model parallelizes entity boundary detection and entity category classification, with the obvious benefit of capturing category information directly from the input sentence. To effectively handle nested structures, we design a boundary detection module consisting of a head-tail encoder and a triaffine classifier. This module carefully captures head-tail features and then integrates contextualized features to detect boundaries. To incorporate category correlations, we propose a sentence-level multi-label category classification module that combines a category-specific attention encoder, a category-correlation graph encoder, and a multi-label classifier. This module fully identifies the most distinctive segments for each entity category and models the category correlations between these segments. Notably, we specifically focus on co-occurrence and nested correlations, which significantly enhance biomedical entity recognition. Finally, a matching module performs entity recognition by combining the knowledge extracted from the boundary detection and category classification modules. Extensive experiments on three nested NER datasets (i.e., GENIA, Chilean Waiting List, ACE 2005) and two flat NER datasets (i.e., JNLPBA, NCBI Disease) reveal that our proposed Bean achieves state-of-the-art performance. To summarize, this paper makes the following contributions:

- We propose the first parallel BioNER model to handle nested structure and category correlation of biomedical entities.
- We exploit head, tail, and contextualized features to detect entity boundaries efficiently via a triaffine model.
- To our best knowledge, we are the first to introduce a multi-label classification model for the BioNER task, capable of extracting entity category information without boundary guidance.

The rest of the paper is organized as follows. "Related work" section reviews the background. "Methods" section formally defines the BioNER task and describes our model in detail. Experimental results and analysis are reported in "Results" section. "Conclusions" section concludes the paper.

Related work

Biomedical Named Entity Recognition. BioNER task aims to recognize named entities [16, 17], which are words or phrases containing the names of predefined categories like DNA, protein, and disease. Recognizing named entities within nested structures (the nested NER task) has recently emerged as an important topic in the BioNER task and benefits various natural language processing applications. Alex et al. [18] introduced three techniques that can reduce the nested NER problem to one or more sequence labeling problems. Ju et al. [12] proposed the first neural layered-based model to identify nested entities by dynamically stacking flat NER layers. Specifically, each flat NER layer is a simple sequence labeling model that contains one Bidirectional Long Short-Term Memory (BiLSTM) [19] encoder and one Conditional Random Field (CRF) [20] decoder. Wang et al. [21] proposed a nested NER model named Pyramid, which has multiple decoding layers to recognize nested entities in a bottom-up manner, from shortest to longest. During this time, a series of layered-based studies [13, 21–24] were introduced, which generally assign one label to each token in the sentence, while each assigned label can express both entity boundary and category information.

Another representative line of work for recognizing nested entities involves classifying each potential text region into one of several predefined entity categories. Zheng et al. [25] proposed a boundary-aware neural model for nested NER, which precisely localizes entities by detecting boundaries (including head tokens and tail tokens) using sequence labeling models and utilizes the boundary-relevant regions to predict entity categories. Yu et al. [26] employed a biaffine model [27] on top of a multi-layer BiLSTM to identify nested entities, where the biaffine model uses biaffine attention instead of bilinear or traditional MLP-based attention to score candidate boundaries. Yuan et al. [28] enhanced this approach by including triaffine attention and scoring, where triaffine attention learns region representations and triaffine scoring interacts with boundaries and region representations for classification. Yan et al. [29] improved region-based methods by utilizing a convolutional neural network (CNN) to model the spatial correlation between neighbor regions. Beyond these, numerous advanced methods have been proposed to refine nested NER [14, 28–34], achieving notable success in the NER task.

Multi-label Learning. In this paper, we propose a specific sentence-level multi-label classification module responsible for learning entity category information from input sentences. Multi-label learning [35] is a prevalent learning paradigm that has been successfully applied in diverse areas. In multi-label text classification, many studies have focused on sentence-level text representation learning and label correlation learning. For example, Yang et al. [36] proposed a Seq2Seq model that predicts labels sequentially, incorporating a BiLSTM with the attention mechanism [37] to encode the input text. Recent studies have indicated that learning label-specific features by capturing the interactions between tokens and labels can significantly improve classification performance. Moreover, while label-specific features are pertinent and discriminative for each class label, the correlations between labels also have a significant impact on multi-label classification [38]. For example, Ye et al. [39] employed the Transformer [37] to capture the text semantics information, and further leveraged the heterogeneous graph transformer to incorporate implicit statistical dependencies between labels. Li et al. [40] designed several regularizers to learn common and

label-specific features for multi-label classification using the correlation information from labels and instances. Especially for text classification, Ma et al. [41] leveraged a dual graph neural network to capture the label co-occurrence interactions.

Methods

This section presents our proposed model BEAN in detail. Figure 2 illustrates the overall architecture of the model, which comprises four key components: an embedding module, a boundary detection module, a category classification module, and a matching module. First, the embedding module generates the token embedding sequence for the given sentence and feeds it simultaneously to the boundary detection and category classification modules. The boundary detection module is designed to predict whether each token pair is the head-tail of an entity; while the category classification module is responsible for modeling the entity category information in the sentence. Finally, the matching module integrates boundary and category features to finalize entity recognition.

Problem statement

Given a sentence, a biomedical NER system learns to generate a triple collection specifying all entities mentioned in the sentence, uniformly handling both flat and nested entities. We use $x = \{w_1, w_2, ..., w_n\}$ to denote the input sentence with n words, and the output triple collection represented by $y = \{< e_1^h, e_1^t, e_1^c >, < e_2^h, e_2^t, e_2^c >, \ldots, < e_k^h, e_k^t, e_k^c >\}$ indicates the k named entities contained in sentence x. The triple e_i^h, e_i^t, e_i^c represents the i-th named entity e_i with head index e_i^h and tail index e_i^t, belonging to the entity category e_i^c from a predefined category set. Correspondingly, the boundary of entity e_i is defined by (e_i^h, e_i^t). The BioNER problem is formally defined as follows,

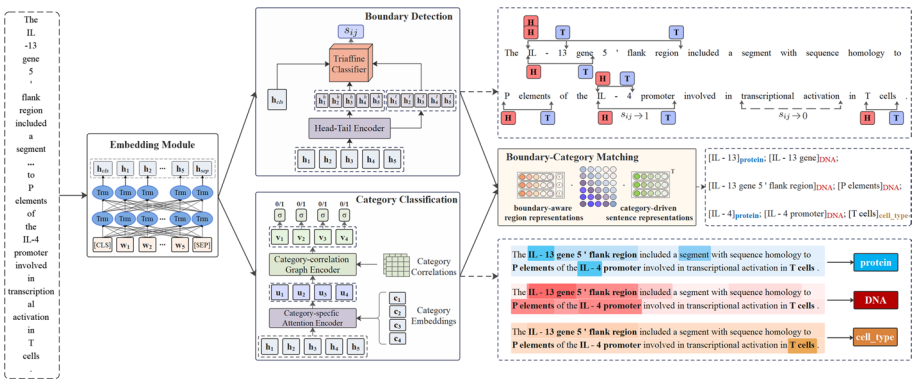

Fig. 2 An overview of the proposed BEAN model. The BEAN contains an embedding layer, a boundary detection module, a category classification module, and a matching module. The boundary detection module detects entity boundaries and generates boundary-aware region-level representations. The category classification module learns the entity category knowledge in the sentence and produces category-driven sentence-level representations. The matching module then integrates two kinds of representations to finalize entity recognition

Problem 1 Biomedical Named Entity Recognition

Given:(1) a set of sentences $X = \{x_1, x_2,..., x_N\}$, where N is the number of sentences in the corpus, (2) a set of triple collections $Y = \{y_1, y_2,..., y_N\}$, where each triple collection $y_i \in Y$ expresses the entity information for sentence x_i, (3) a set of predefined entity categories $C = \{c_1, c_2,..., c_M\}$, where M is the total number of entity categories, (4) a new sentence $x^* \notin X$;
Find: the triple collection y^* for the new sentence x^*.

Embedding module

Given an input sentence, we utilize the context-sensitive pre-trained language model BERT [42] as the backbone to obtain contextualized token representations for the input sentence. BERT, along with its domain-specific variant BioBERT, is an open-source model widely adopted in advanced studies [26, 28, 29]. To be consistent with BERT, we add the special tokens [CLS] and [SEP] to the input sentence as the first and last tokens, respectively. For a sentence with n tokens, $x = \{w_1, w_2,..., w_n\}$, the BERT model receives the combined string and computes the output $\mathbf{h}$:

$$\begin{aligned}\mathbf{h} &= \{\mathbf{h}_{cls}, \mathbf{h}_1, \ldots, \mathbf{h}_n, \mathbf{h}_{sep}\} \\ &= PLM([CLS][w_1, w_2, \ldots, w_n][SEP]),\end{aligned} \tag{1}$$

where $\mathbf{h}_{cls}$ is the contextualized sentence representation, and $\mathbf{h}_c = \{\mathbf{h}_1,..., \mathbf{h}_n\}$ denotes the sequence of contextualized token representations. We then feed both $\mathbf{h}_{cls}$ and $\mathbf{h}_c$ into the subsequent boundary detection and category classification modules simultaneously.

Boundary detection module

The boundary detection module is a pair-wise classifier that determines whether each token pair in a sentence is the boundary of an entity. As shown in Fig. 2, both "IL - 4 promoter" and "IL - 4" are entities. The boundary detection module should ideally detect token pairs [IL, promoter] and [IL, 4] as entity boundaries. This module can be described as a mapping of token pairs, along with global textual information, to boundary scores using a head-tail encoder and a triaffine classifier.

Head–tail encoder

Since an entity boundary includes a head and a tail, we aim to learn the boundary-head token representation $\mathbf{h}_k^h$ and boundary-tail token representation $\mathbf{h}_k^t$ for each token. Specifically, the head-tail encoder generates $\mathbf{h}_k^h$ and $\mathbf{h}_k^t$ based on token representation $\mathbf{h}_k$ using two separate linear layers, as described below,

$$\mathbf{h}_k^h = \text{Linear}^h(\mathbf{h}_k, \theta^h), \tag{2}$$

$$\mathbf{h}_k^t = \text{Linear}^t(\mathbf{h}_k, \theta^t), \tag{3}$$

where θ^h and θ^t are the parameters of linear layers.

Triaffine classifier

To determine whether the token pair (w_i, w_j) forms the boundary of an entity (where $i \leq j$), we feed the corresponding head-tail token representation pair $(\mathbf{h}_i^h, \mathbf{h}_j^t)$, along with the global contextualized representation $\mathbf{h}_{cls}$, into a triaffine classifier [43]. This classifier computes the score s_{ij} for the token pair (w_i, w_j) as follows,

$$s_{ij} = \begin{bmatrix} \mathbf{h}_{cls} \\ 1 \end{bmatrix}^{\mathrm{T}} (\mathbf{h}_i^h)^{\mathrm{T}} \mathbf{W}^{\text{triaffine}} \begin{bmatrix} \mathbf{h}_j^t \\ 1 \end{bmatrix}, \tag{4}$$

where $\mathbf{W}^{\text{triaffine}}$ is a $d \times d \times d$ tensor. As illustrated in Fig. 2, the score s_{ij} for the token pair corresponding to valid entity "IL - 4 promoter" (depicted by a solid line) should be high; conversely, the score s_{ij} for the non-entity "transcriptional activation" (depicted by a dashed line) should be close to 0.

Since only a few boundaries correspond to valid entities, we employ Focal Loss [44] to alleviate the class imbalance problem faced by the boundary detector during training. Focal Loss alleviates this imbalance by down-weighting easy examples (typically from the majority class) and emphasizing harder examples from the minority class, thereby encouraging the model to focus on more challenging cases. The formula to evaluate the boundary detection module is expressed as follows:

$$\begin{aligned} \mathcal{L}^B &= \sum_{ij} -\beta'_{ij}(1 - s'_{ij})^{\gamma} \log(s'_{ij}), \\ (s'_{ij}, \beta'_{ij}) &= \begin{cases} (\sigma(s_{ij}), \beta_{ij}), & \text{if } (w_i, w_j) \text{ is a boundary}; \\ (1 - \sigma(s_{ij}), 1 - \beta_{ij}), & \text{otherwise}, \end{cases} \end{aligned} \tag{5}$$

where $(1 - s'_{ij})^{\gamma}$ denotes the modulating factor and γ is the focusing parameter; β_{ij} denotes the weighting factor.

For a sentence containing k entities, the boundary detection module predicts k token pairs as entity boundaries. For the i-th predicted boundary, we denote it as (e_i^h, e_i^t), and its corresponding region $r_i = \{w_{e_i^h}, w_{e_i^h+1}, \ldots, w_{e_i^t-1}, w_{e_i^t}\}$ is a sub-string of input sentence x. Finally, we calculate the boundary-aware region representation $\mathbf{r}_i$ of boundary (e_i^h, e_i^t):

$$\mathbf{r}_i = \mathbf{h}_{e_i^h}^h \oplus \mathbf{h}_{e_i^t}^t \oplus \left[\frac{1}{e_i^t - e_i^h + 1} \sum_{t=e_i^h}^{e_i^t} \mathbf{h}_t \right]. \tag{6}$$

The obtained boundary-aware region representations $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k\}$ of predicted boundaries are then input into the matching module to determine their entity categories.

Category classification module

The category classification module is a sentence-level multi-label classifier that can predict which entity categories are contained in the sentence. As shown in Fig. 2, given a sentence containing multiple entities of different categories, the category classification module can assign multiple labels to it, i.e., protein, DNA, and cell_type categories. Specifically, a category-specific attention encoder captures the

category-specific features for each entity category, and a category-correlation graph encoder further incorporates the category correlations between them.

Category-specific attention encoder

We first design a category-guided attention mechanism to learn category-specific representations. This mechanism can explicitly extract information related to each category from the input sentence. Specifically, we initialize the category embedding $\mathbf{c}_i$ randomly and calculate the i-th category-specific representation $\mathbf{u}_i$ of the input sentence as follows,

$$\begin{aligned} \alpha_{ij} &= \frac{\exp(\mathbf{h}_j \cdot \mathbf{c}_i^T)}{\sum_j \exp(\mathbf{h}_j \cdot \mathbf{c}_i^T)}, \\ \mathbf{u}_i &= \sum_j \alpha_{ij} \cdot \mathbf{h}_j, \end{aligned} \tag{7}$$

where $\mathbf{c}_i$ is the i-th category embedding, and α_{ij} indicates the importance of the j-th token to the i-th category. It is worth mentioning that the obtained $\boldsymbol{u}_i$ is a sentence-level representation, the core of which is the semantic segments associated with the i-th category.

Category-correlation graph encoder

To further incorporate entity category correlations for the BioNER task, we construct a category-correlation graph and propagate messages between nodes using a graph neural network. Formally, the category-correlation graph is defined as $G =< V, E, R >$, where V is the set of category nodes, E is the set of edges representing statistical correlations between nodes, and R is the set of two edge types (i.e., co-occurrence and nested).

In this graph, category-specific representations obtained from the category-specific attention encoder are used as the initial node representations. We calculate the probability between all category pairs in the training set, producing two matrices: co-occurrence correlation matrix P and nested correlation matrix Q. Then, we employ an attention-based EGNN [45] to update and capture multi-dimensional edge feature relationships. In the multi-layer attention-based EGNN, each layer takes the node representations and correlation information from the previous layer (i.e., U^{l-1}, P^{l-1}, and Q^{l-1}) as inputs, and outputs enhanced category-specific representations and correlation matrices (i.e., U^l, P^l, and Q^l). The layer-wise propagation rule is as follows,

$$\begin{aligned} U^l = \mathrm{ELU}[&(\alpha^l(U^{l-1}, P^{l-1})g^l(U^{l-1})) \\ &\oplus(\beta^l(U^{l-1}, Q^{l-1})g^l(U^{l-1}))]. \end{aligned} \tag{8}$$

Here, $\oplus$ is the concatenation operation; Exponential Linear Unit (ELU) is employed as nonlinear activation; U^0 is initialized by category-specific representations $\boldsymbol{u}$; g is a transformation used by $g^l(U^{l-1}) = U^{l-1}\mathbf{W}^l$, where $\mathbf{W}^l$ is a learned weight matrix. α^l and β^l contain attention coefficients for co-occurrence correlation and nested correlation, respectively. The attention function is chosen to be the following,

$$\begin{aligned} \alpha_{ij}^{l} &= \text{DS}\left(f^{l}(U_{i}^{l-1}, U_{j}^{l-1})P_{ij}^{l-1}\right), \\ \beta_{ij}^{l} &= \text{DS}\left(f^{l}(U_{i}^{l-1}, U_{j}^{l-1})Q_{ij}^{l-1}\right), \end{aligned} \tag{9}$$

where f^l is a linear function to produce a scalar value, DS is the doubly stochastic normalization operator [45]. DS normalization is a practical method derived from the mathematical properties of doubly stochastic matrices. In multi-layer graph neural networks, where edge feature matrices are repeatedly multiplied across layers, DS normalization helps stabilize the process. The attention coefficients for two correlations will be used as new edge features for the next layer: $P^l = \alpha^l, Q^l = \beta^l$. Following these, the node representations of the final layer U^L embody the correlations between entity categories. We denoted these node representations U^L as $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_M\}$, where $\mathbf{v}_i$ is the category-driven sentence representation related to the i-th category.

Multi-label classifier

Next, we feed each category-driven sentence representation into a multi-label classifier, which consists of a two-layer MLP, to predict whether the i-th category entity appears in the sentence. The predicted result $\tilde{q}_i$ is computed as follows,

$$\tilde{q}_i = \sigma(\text{MLP}_\text{i}(\mathbf{v}_\text{i})). \tag{10}$$

We apply the multi-label cross entropy loss function to evaluate the category classification module. The objective function for each sentence is defined as

$$\mathcal{L}^C = \sum_{i}^{M} -(q_i \log(\tilde{q}_i) + (1 - q_i) \log(1 - \tilde{q}_i)), \tag{11}$$

where q_i indicates the ground truth for whether the i-th category entity appears in the sentence. For a sentence containing m entity categories, the category classification module will predict m entity categories. The corresponding m category-driven sentence representations $\{\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_m\}$ are then passed to the matching module to further determine the boundary of the entity. Note that only the category-driven sentence representations of the categories predicted by the category classification module are retained for further processing.

Matching module

For a sentence with k entities belonging to m categories,[1] we obtain k boundary-aware region representations and m category-driven sentence representations. We then construct boundary-category pairs $(\mathbf{r}_i, \mathbf{s}_j)$ through pairwise combination and feed them into a biaffine classifier [27] to estimate the boundary-category relevancy rel_{ij} of each pair:

$$rel_{ij} = (\mathbf{s}_j)^\top \mathbf{W}^1 \mathbf{r}_i + (\mathbf{s}_j \oplus \mathbf{r}_i)^\top W^2 + b, \tag{12}$$

where $\oplus$ denotes concatenation operation, $\mathbf{W}^1$ and W^2 are weight matrices, and b is the bias. A higher boundary-category relevancy rel_{ij} indicates that the i-th region is more likely to be a j-th category entity. For all boundary-category pairs, we obtain a relevancy

[1] Multiple entities might belong to the identical category.

matrix $R = (rel_{ij}) \in \mathbb{R}^{k \times m}$. We then determine whether each boundary-category pair matches using $\tilde{p}_{ij} = \sigma(rel_{ij})$. The objective function of the matching module for each sentence is defined as:

$$\mathcal{L}^M = \sum_{ij} -(p_{ij} \log(\tilde{p}_{ij}) + (1 - p_{ij}) \log(1 - \tilde{p}_{ij})), \tag{13}$$

where p_{ij} denotes the real label indicating whether the i-th region is a j-th category entity. Finally, we match the entity category e_i^c for each entity boundary (e_i^h, e_i^t), and the recognized entity is denoted as e_i^h, e_i^t, e_i^c .

Training

We define the final multi-task loss as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}^B + \lambda_2 \mathcal{L}^C + \lambda_3 \mathcal{L}^M, \tag{14}$$

where λ_1, λ_2, and λ_3 are hyper-parameters of $\mathcal{L}^B$ in Eq. (5), $\mathcal{L}^C$ in Eq. (11), and $\mathcal{L}^M$ in Eq. (13), respectively. Note that the proposed BEAN detects entity boundaries and predicts entity categories in parallel. During training, we use all ground-truth labels, ensuring that all components are trained simultaneously.

Results

In this section, we introduce the datasets, baseline methods, and implementation details. Next, we present the experimental results, discuss the parallel processing strategy, and analyze the results of ablation studies.

Datasets

To evaluate the performance of our proposed model, we conduct experiments on five widely used NER datasets. Three datasets involve nested entities: GENIA and Chilean Waiting List in the biomedical domain, and ACE 2005 in the general domain. Additionally, we experiment on two flat NER datasets in biomedical domain, i.e., JNLPBA and NCBI Disease, to further validate our approach.

- *GENIA*[2] is a biomedical nested corpus created from 2000 MEDLINE abstracts. It consists of 18,535 sentences with 56,036 entity mentions, of which over 18% are nested. In this work, we adopt the same train/dev/test split as Yan et al. [29].
- *Chilean Waiting List*[3] is a Spanish clinical corpus created from real diagnoses of the Chilean healthcare system. It is composed of 43,730 entity mentions across seven entity types, with 46.7% of entities involved in nesting. We used the public files released by the authors [46], which are already tokenized.
- *ACE 2005*[4] is a general-domain dataset containing 9189 sentences with 31,720 entities across seven categories, and about 39% entities are nested. We pre-process the ACE2005 dataset following Yan et al. [29].

[2] https://huggingface.co/datasets/Rosenberg/genia.
[3] https://zenodo.org/records/5591011.
[4] https://catalog.ldc.upenn.edu/LDC2006T06.

- *JNLPBA*[5] is derived from the GENIA corpus. Only flat, top-level entities are retained in this dataset. We pre-process and split this dataset following Zhang et al. [34].
- *NCBI Disease*[6] is a resource for disease name recognition and concept normalization. It is created from 793 PubMed abstracts, containing 6892 flat disease mentions. We pre-process and split this dataset following Zhang et al. [34].

Baseline methods

To validate the effectiveness of our proposed model for the nested NER task, we compare it with state-of-the-art nested NER methods, including prominent layered-based and region-based methods. Additionally, to further demonstrate the applicability of our model to the flat NER task, we conduct comparisons with leading models specifically designed for identifying flat entities.

Layered-based Nested NER methods. Layered-based methods typically consist of multiple layers (or levels) to handle nested structures, where each layer identifies a group of entities of a specific level or length.

- LayeredNER [12] applies CRF to nested NER. It first encodes the input sentence using stacked flat LSTM layers and then decodes it into categories by cascaded CRFs.
- Pyramid [21] is a layered-based model with multiple LSTM-CNN-based decoding layers. Pyramid recognizes nested entities in a bottom-up manner, starting from the shortest to the longest.
- MLC [24] revisits multiple LSTM-CRF modules by incorporating a stacked embedding layer that includes domain-specific word embeddings, character embeddings, and contextual word embeddings.

Region-based Nested NER methods. Region-based methods generally frame the nested NER task as a multi-class classification problem, classifying each potential region into one of several predefined categories.

- HIT [14] contains a biaffine-based head-tail detector and a BiLSTM-based token interaction tagger. It models the boundary token and captures the connection relationship between tokens within a region.
- Biaffine-NER [26] employs a biaffine model on top of a multi-layer BiLSTM, scoring candidate boundaries and predicting their categories with additional paragraph-level features.
- BERT-MRC [32] is a region-based method that leverages word embeddings with prior knowledge. It uses BERT as the backbone to encode prior knowledge by transforming different entity categories intoto question queries.
- Triaffine-NER [28] includes triaffine attention and scoring, where triaffine attention learns region representations and triaffine scoring interacts with boundaries and region representations for classification.

[5] https://huggingface.co/datasets/jnlpba/jnlpba.

[6] https://www.ncbi.nlm.nih.gov/research/bionlp/Data/disease.

- DiffusionNER [33] converts nested NER into a boundary denoising diffusion process to identify entity boundaries from noisy spans, including a boundary forward diffusion process and a boundary reverse diffusion process.
- CNN-NER [29] enhances region-based methods by exploiting the spatial correlation between neighbor regions through CNN.
- BINDER [34] is a bi-encoder framework applying contrastive learning, where the bi-encoder encodes both entity types and text regions using a Transformer.

Flat NER methods. Several advanced NER models are designed specifically for flat entities and fail to process nested entities. The following are representative biomedical language models and LLM-based NER models.

- BioBERT [47] is a biomedical-specific language model built on general-domain BERT, continually pre-trained on large-scale biomedical corpora, including PubMed abstracts and PMC full-text articles.
- PubMedBERT [48] is a biomedical language model with a BERT architecture. It starts domain-specific pretraining from scratch using both PubMed abstracts and PMC full-text articles.
- BioNER-LLaMA [15] is an LLM-based nested NER method that learns biomedical knowledge through instruction tuning. It is built on the LLaMA foundation and fine-tuned on biomedical NER corpora with instruction-following examples.

Implementation details

Consistent with state-of-the-art baselines, we employ pre-trained contextual embeddings to initialize our proposed model. For a fair performance comparison [26, 28, 29], we use BERT-large for ACE 2005 (general domain), and BioBERT-large [47] for GENIA, JNLPBA, and NCBI Disease (biomedical domain). For the Chilean Waiting List, where BERT and BioBERT are unavailable for the Spanish biomedical domain, we use the character-level language model Flair [49]. This choice is especially useful for handling misspellings and out-of-vocabulary words in domain-specific text. Our experimental setup for this dataset primarily follows Rojas et al. [24].

Additionally, the hyperparameters of our model are chosen by performing a random search, selecting the optimal configuration based on development set performance. We report the main hyperparameters used in our experiments. Each linear layer in the boundary detection module has a depth of 1 and a hidden size of 300. In the category classification module, we use 2 layers of EGNN with dropout [50] rate 0.6 to both input features and normalized attention coefficients. The MLP in the multi-label classifier has a depth of 2 and a hidden size of 300. The focusing parameter γ is set to 2, and the β_{ij} is set to 0.7. The weights λ_1, λ_2, and λ_3 are set to 0.4, 0.4, and 0.2, respectively. The initial learning rate is set to 0.008 and gradually decreases as the training step increases. We apply an early stopping strategy with a window size of 40 across all experiments. The batch size is set to 16 for all datasets at the sentence level. Our model is implemented using PyTorch and trained with the Adam optimizer. All our experiments are performed on NVIDIA RTX 4090 GPU and Intel i9-13900K CPU. We monitor training

Table 1 Main results on GENIA and ACE 2005 datasets for nested NER task

Model	LM	GENIA			ACE 2005		
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Layered-based methods							
LayeredNER	–	78.50	71.30	74.70	74.20	70.30	72.20
Pyramid	$BioB_{v1.1}$+F/ALB_L+B_L	80.31	78.33	79.31	85.30	87.40	86.34
MLC	B_L+F	80.10	75.20	77.60	–	–	–
Region-based methods							
HIT	–	78.15	74.44	76.23	78.18	75.35	76.74
Biaffine-NER	B_L	81.80	79.30	80.50	85.20	85.60	85.40
BERT-MRC	B_L	81.14	76.82	78.92	87.16	86.59	86.88
Triaffine-NER	B_L/$BioB_{v1.1}$	80.42	82.06	81.23	86.70	86.94	86.82
DiffusionNER	$BioB_L$/B_L	82.10	80.97	81.53	86.15	87.72	86.93
CNN-NER	$BioB_B$/B_L	81.52	79.17	80.33	86.26	87.56	86.91
BINDER	$BioB_L$/B_L	83.40	78.30	80.80	89.60	90.50	90.00
Our BEAN	$BioB_L$/B_L	82.38	81.04	81.71	87.02	87.66	87.34

B: BERT, BioB: BioBERT, ALB: ALBERT, F: Flair, $-_L$: large, $-_B$: base. This table does not list traditional word embeddings (e.g., Glove, word2vet)

Table 2 Main results on Chilean Waiting List (CWL) dataset for nested NER task

Model	LM	CWL		
		P(%)	R(%)	F(%)
Layered-based methods				
LayeredNER	–	75.0	72.8	73.9
Pyramid	Flair	80.10	77.20	78.60
MLC	Flair	80.60	80.50	80.50
Region-based methods				
HIT	Flair	77.43	76.85	77.14
Biaffine-NER	BERT-base	78.70	70.80	74.50
Our BEAN	Flair	80.48	80.57	80.53

performance on the development set and report final results on the test set. To provide an estimate of the computational cost, training our model requires approximately four hours for each dataset.

Main results

We employ the precision (P), recall (R), and F1-score (F) to evaluate the performance of each method. Table 1 presents experimental results for the English datasets GENIA and ACE 2005, while Table 2 reports results for the Spanish dataset Chilean Waiting List. These tables provide a comprehensive evaluation of the model for identifying nested entities across different languages and domains. As shown in Tables 1 and 2, our BEAN achieves state-of-the-art performance across all datasets. BEAN achieves an F1-score of 81.71% and a recall of 81.04% on GENIA, surpassing all baseline methods. Specifically, it outperforms region-based approaches by 0.19—5.48% in F1-score. These results demonstrate the effectiveness of our proposed entity category learning module in recognizing biomedical named entities. On the ACE 2005 dataset, BEAN achieves competitive results

Table 3 Results of entities for each category on GENIA test dataset

Category	P(%)	R(%)	F(%)
DNA	80.04	79.75	79.89
RNA	90.96	86.70	88.78
Pretein	83.85	82.94	83.39
Cell line	82.33	76.78	79.46
Cell type	79.29	78.41	78.85
Overall	82.38	81.04	81.71

Table 4 Main results on JNLPBA and NCBI Disease datasets for flat BioNER task

Model	LM	JNLPBA			NCBI disease		
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
BioBERT	–	72.68	83.21	77.59	89.04	89.69	89.36
PubMedBERT	–	–	–	79.16	–	–	88.04
BioNER-LLaMA	LLaMA-7B	–	–	–	87.40	88.60	88.00
BINDER	PubMedBERT-base	–	–	80.30	–	–	90.90
Our BEAN	BioBERT	74.97	86.81	80.46	89.71	91.06	90.38

with 87.34% F1-score, ranking second only to BINDER. A major reason for BINDER's superior performance is that it utilizes the ACE 2004 dataset as additional training data, enabling it to achieve superior performance compared to other methods on the ACE 2005. On the Chilean Waiting List, BEAN demonstrates superior performance in both recall and F1-score, while maintaining comparable precision. Notably, BEAN attains the highest recall on both the GENIA and Chilean Waiting List datasets, indicating that it rarely filters out valid entities. This improvement is likely due to the fact that our model performs boundary detection and category classification in parallel, effectively mitigating error propagation in biomedical entity recognition.

Specifically, Table 3 shows the performance of each category on GENIA. The proposed BEAN achieves the best performance in recognizing the entities of the RNA category. The reason for this might be that entities related to RNA often end with either "mRNA" or "RNA". These two words are informative indicators of RNA entities. BEAN yields 83.39% F1-score on the protein category, which accounts for more than half of the entities in GENIA.

We also conduct experiments on two biomedical NER datasets, JNLPBA and NCBI Disease, which contain only flat entities. The primary goal is to demonstrate the applicability of BEAN to flat entities and compare it with existing biomedical language models and LLM-based NER models, which can only handle non-nested entities. The experimental results are reported in Table 4. Our BEAN achieves the highest performance on the JNLPBA dataset, surpassing its strongest competitor BINDER by 0.16% in F1-score and outperforming BioBERT by 2.87%. On the NCBI Disease dataset, BEAN achieves a competitive F1-score of 90.38%, trailing BINDER, the top performer, by just 0.52%. However, our model significantly outperforms the LLM-based BioNER-LLaMA, with gains of 2.31% in precision, 2.46% in recall, and 2.38% in F1-score. Considering that

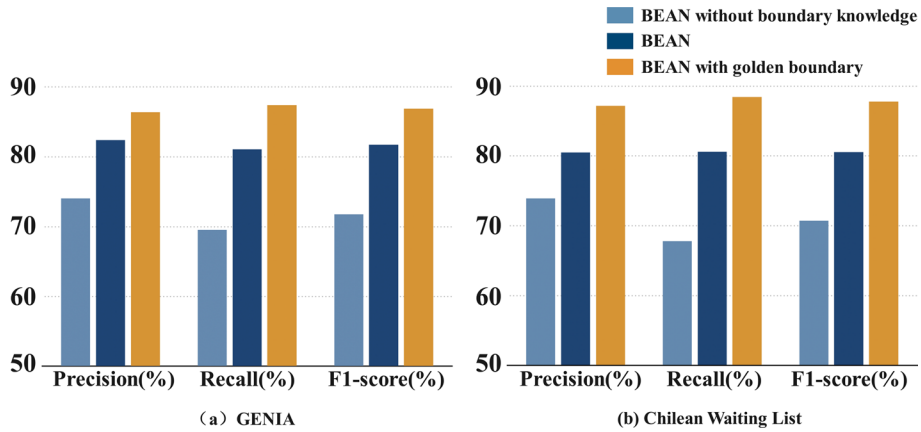

Fig. 3 Analysis of learning entity boundary knowledge

BioNER-LLaMA is built on the LLaMA foundation with 7B parameters, our model, which adopts BioBERT, underscores its efficiency. Overall, our results confirm that processing boundary detection and category classification in parallel is still an effective approach for flat NER.

Discussion

Discussion of parallel processing strategy

Our proposed model BEAN is designed to parallelize boundary detection and category classification. To evaluate the necessity of this strategy, we conduct comparative experiments on the GENIA and Chilean Waiting List datasets.

Analysis of Learning Entity Boundary Knowledge. Our model includes a boundary detection module specifically designed to predict entity boundaries in sentences. To assess the importance of learning entity boundaries, we use the category classification module to independently perform the BioNER task. Instead of using the results of the boundary detection module, we provide entity boundaries to the category classification module in two strategies: a simple enumeration strategy that lists all possible regions (denoted as "BEAN without boundary knowledge") and a golden boundary strategy that uses golden entity boundary labels (denoted as "BEAN with golden boundary"). The experimental results are presented in Fig. 3. BEAN without boundary knowledge cannot achieve satisfactory recognition performance on either dataset. One possible explanation is that the lack of explicit boundaries might lead to similar representations in many regions, thus confusing entity recognition. In contrast, BEAN with golden boundary performs exceptionally well on both datasets, significantly outperforming BEAN without boundary knowledge. These results validate the importance of explicit boundary information for effectively identifying nested entities. While BEAN with golden boundary performs slightly better, BEAN remains highly competitive.

Analysis of Learning Entity Category Knowledge. Our model uses the category classification module to assign entity categories in sentences. To assess the importance of learning entity boundary information, we conduct experiments in which the boundary detection module performs the entire BioNER task independently. For comparison, we modify the boundary detection module to predict region categories directly

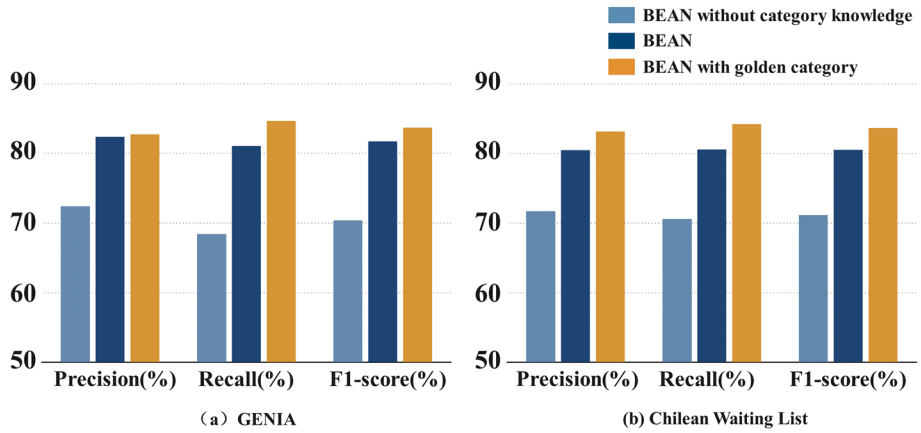

Fig. 4 Analysis of learning entity category knowledge

Table 5 Performance of the boundary detection based on different structures

Structure	GENIA	Chilean waiting list
Tagging detection	82.43	81.22
Token-wise detection	82.77	82.07
Biaffine detection	84.69	82.86
Our boundary detection	85.94	84.23

All numbers are F1-scores (%)

(denoted as "BEAN without category knowledge"). Additionally, we implement a golden category strategy, replacing the category classification module's predictions with golden entity category labels (denoted as "BEAN with golden category"). The corresponding experimental results are shown in Fig. 4. BEAN with golden category yields the best performance, notably surpassing BEAN without category knowledge. These results demonstrate that the NER model can achieve further breakthroughs if it captures more category information. Even without golden category labels, BEAN still delivers competitive performance. This corroborates the effectiveness of our proposed category classification module in learning entity category information.

Ablation study

We conduct ablation experiments to assess the contribution of key components of our proposed BEAN. The experimental results are shown in Tables 5 and 6.

Table 6 Performance of category classification for each category (CL: Cell line, CT: Cell type) on GENIA test dataset based on different structures

Structure	DNA	RNA	Protein	CL	CT	Avg
CC w/o CS	89.52	87.57	91.19	84.32	85.46	89.04
CC w/o CR	91.21	89.30	92.06	85.10	87.55	90.33
Our CC	91.82	90.05	92.80	86.67	88.79	91.21

All numbers are F1-scores (%)

Effectiveness of Boundary Detection Module. The boundary detection module in our model comprises a head-tail encoder and a triaffine classifier. To evaluate the boundary detection module, we test the following variants on the GENIA and Chilean Waiting List datasets: (1) tagging detection: predicts boundary labels with BIEO tagging strategy, then matches each token labeled "B" to the token labeled "E" to form candidate entity regions; (2) token-wise detection: predicts start and end positions with two token-wise classifiers, then forms valid regions where the end position is larger than the start position; (3) biaffine detection: retains the head-tail encoder but replaces the triaffine classifier with a biaffine classifier, which lacks global contextualized representation. From Table 5, we observe that variants with the head-tail encoder outperform tagging detection and token-wise detection. These results suggest that separately modeling boundary-head and boundary-tail knowledge for each token enhances entity boundary detection. Additionally, incorporating sentence-level textual information significantly improves performance over the biaffine detection. These findings demonstrate the crucial role of textual information in accurate boundary detection.

Effectiveness of Category Classification Module. In our model, the category classification module learns entity category knowledge by leveraging category-specific and category-correlation features. To illustrate the effectiveness of our method, we conduct a sentence-level multi-category classification task on the GENIA dataset using different network structures: (1) CC w/o CS: replaces the category-specific attention encoder with a traditional self-attention mechanism; (2) CC w/o CR: removes the category-correlation graph encoder from the category classification module. Here, CC, CS, and CR are the abbreviations for category classification, category-specific attention and category-correlation graph, respectively. Table 6 shows that our category classification module obtains the best performance compared with CC w/o CS and CC w/o CR. This suggests that both category-specific and category-correlation features are crucial for improving category classification. Additionally, CC w/o CR outperforms CC w/o CS, indicating that the improvements provided by category-specific features are even more pronounced. We plan to explore ways to better utilize category-correlation features in the future.

Conclusions

In this paper, we propose a novel BioNER model, BEAN, that parallelizes two objectives of the NER task: detecting entity boundaries and classifying entity categories. Specifically, the boundary detection module leverages the head-tail pair and textual information to predict entity boundaries, producing boundary-aware region-level representations. The category classification module learns category-specific features and integrates category-correlation features to perform classification, preserving category-driven sentence-level representations. Finally, the matching module performs entity recognition by combining these two types of representations. Extensive experiments on five public datasets (including four BioNER datasets and one general NER dataset) demonstrate that our model achieves significant improvements over state-of-the-art models.

In future work, we plan to explore more efficient training strategies. We found experimentally that the loss of our fusion module is difficult to converge in the early stages of training. An effective manner is to prioritize training the two modules responsible for

learning boundary and category knowledge, introducing the fusion module later in the process. Additionally, our model struggles with handling discontinuous entities or novel entity categories as they continuously emerge. We plan to explore these more realistic scenarios in future work.

Acknowledgements
Not applicable.

Author contributions
Conceptualization, Y.W. and FZ.H.; Methodology, Y.W., ZY.Z., and FZ.H.; Software, Y.W. and ZY.Z.; Formal analysis, Y.W. and ZY.Z.; Writing-original draft preparation, Y.W. and ZY.Z.; Writing-review and editing, HH.T. and Y.L. All authors read and approved the final manuscript.

Funding
This work was supported by the Natural Science Foundation of China (Grants Nos. 62306339, 62476137, 62406148) and the Natural Science Foundation of Jiangsu Province (Grant No. BK20240647).

Availability of data and materials
The dataset used and analyzed during this study is publicly available and can be downloaded.
GENIA: https://huggingface.co/datasets/Rosenberg/genia;
Chilean Waiting List: https://zenodo.org/records/5591011;
ACE 2005: https://catalog.ldc.upenn.edu/LDC2006T06;
JNLPBA: https://huggingface.co/datasets/jnlpba/jnlpba;
NCBI Disease: https://www.ncbi.nlm.nih.gov/research/bionlp/Data/disease.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Conflict of interest
The authors declare that they have no conflict of interest.

Received: 19 September 2024 Accepted: 14 February 2025
Published online: 25 February 2025

References

1. Li J, Sun A, Han J, Li C. A survey on deep learning for named entity recognition. IEEE Trans Knowl Data Eng. 2020;34:50–70.
2. Wang Y, Tong H, Zhu Z, Li Y. Nested named entity recognition: a survey. ACM Trans Knowl Discov Data. 2022;16(6):1–29.
3. Esteva A, Kale A, Paulus R, Hashimoto K, Yin W, Radev D, Socher R. Covid-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization. NPJ Digit Med. 2021;4(1):68.
4. Murali L, Gopakumar G, Viswanathan DM, Nedungadi P. Towards electronic health record-based medical knowledge graph construction, completion, and applications: a literature study. J Biomed Inform. 2023;143: 104403.
5. Jin Q, Yuan Z, Xiong G, Yu Q, Ying H, Tan C, Chen M, Huang S, Liu X, Yu S. Biomedical question answering: a survey of approaches and challenges. ACM Comput Surv. 2022;55(2):1–36.
6. Silla CN, Freitas AA. A survey of hierarchical classification across different application domains. Data Min Knowl Disc. 2011;22(1–2):31–72.
7. Dang TH, Le H-Q, Nguyen TM, Vu ST. D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. Bioinformatics. 2018;34:3539–46.
8. Zhou H, Ning S, Liu Z, Lang C, Liu Z, Lei B. Knowledge-enhanced biomedical named entity recognition and normalization: application to proteins and genes. BMC Bioinform. 2020;21:1–15.
9. Wang Y, Li Y, Zhu Z, Tong H, Huang Y. Adversarial learning for multi-task sequence labeling with attention mechanism. IEEE/ACM Trans Audio Speech Lang Process. 2020;28:2476–88.
10. Lee EB, Heo GE, Choi CM, Song M. MLM-based typographical error correction of unstructured medical texts for named entity recognition. BMC Bioinform. 2022;23(1):486.
11. Luo L, Wei C-H, Lai P-T, Leaman R, Chen Q, Lu Z. AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning. Bioinformatics. 2023;39(5):btad310.
12. Ju M, Miwa M, Ananiadou S. A neural layered model for nested named entity recognition. In: Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies; 2018. pp. 1446–59.
13. Fei H, Ren Y, Ji D. Dispatched attention with multi-task learning for nested mention recognition. Inf Sci. 2020;513:241–51.

14. Wang Y, Li Y, Tong H, Zhu Z. 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; 2020. pp. 6027–36.
15. Keloth VK, Hu Y, Xie Q, Peng X, Wang Y, Zheng A, Selek M, Raja K, Wei CH, Jin Q, et al. Advancing entity recognition in biomedicine via instruction tuning of large language models. Bioinformatics. 2024;40(4):btae163.
16. Xiong Y, Chen S, Tang B, Chen Q, Wang X, Yan J, Zhou Y. Improving deep learning method for biomedical named entity recognition by using entity definition information. BMC Bioinform. 2021;22:1–13.
17. Guan Z, Zhou X. A prefix and attention map discrimination fusion guided attention for biomedical named entity recognition. BMC Bioinform. 2023;24(1):42.
18. Alex B, Haddow B, Grover C. Recognising nested named entities in biomedical text. In: Biological, translational, and clinical language processing; 2007. pp. 65–72.
19. Palangi H, Deng L, Shen Y, Gao J, He X, Chen J, Song X, Ward R. Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans Audio Speech Lang Process. 2016;24(4):694–707.
20. Lafferty J, McCallum A, Pereira FC. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the international conference on machine learning; 2001.
21. Wang J, Shou L, Chen K, Chen G. Pyramid: a layered model for nested named entity recognition. In: Proceedings of the 58th annual meeting of the association for computational linguistics; 2020. pp. 5918–28.
22. Fisher J, Vlachos A. Merge and label: a novel neural network architecture for nested NER. In: Proceedings of the 57th annual meeting of the association for computational linguistics; 2019. pp. 5840–50.
23. Shibuya T, Hovy E. Nested named entity recognition via second-best sequence learning and decoding. Trans Assoc Comput Linguist. 2020;8:605–20.
24. Rojas M, Bravo-Marquez F, Dunstan J. Simple yet powerful: an overlooked architecture for nested named entity recognition. In: Proceedings of the 29th international conference on computational linguistics; 2022. pp. 210–17.
25. Zheng C, Cai Y, Xu J, Leung H-f, Xu G. 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; 2019. pp. 357–66.
26. Yu J, Bohnet B, Poesio M. Named entity recognition as dependency parsing. In: Proceedings of the 58th annual meeting of the association for computational linguistics; 2020, pp. 6470–6.
27. Dozat T, Manning CD. Deep biaffine attention for neural dependency parsing. In: Proceedings of the 5th international conference on learning representations; 2017.
28. Yuan Z, Tan C, Huang S, Huang F. Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition. In: Findings of the association for computational linguistics: ACL 2022; 2022. pp. 3174–86.
29. Yan H, Sun Y, Li X, Qiu X. An embarrassingly easy but strong baseline for nested named entity recognition. In: Proceedings of the 61st annual meeting of the association for computational linguistics (volume 2: short papers); 2023. pp. 1442–52.
30. Tan, C., Qiu, W., Chen, M., Wang, R., Huang, F.: Boundary enhanced neural span classification for nested named entity recognition. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34; 2020. pp. 9016–23.
31. Sun L, Sun Y, Ji F, Wang C. Joint learning of token context and span feature for span-based nested NER. IEEE/ACM Trans Audio Speech Lang Process. 2020;28:2720–30.
32. Li X, Feng J, Meng Y, Han Q, Wu F, Li J. A unified MRC framework for named entity recognition. In: Proceedings of the 58th annual meeting of the association for computational linguistics; 2020.
33. Shen Y, Song K, Tan X, Li D, Lu W, Zhuang Y. Diffusionner: boundary diffusion for named entity recognition. In: Proceedings of the 61st annual meeting of the association for computational linguistics; 2023.
34. Zhang S, Cheng H, Gao J, Poon H. Optimizing bi-encoder for named entity recognition via contrastive learning. In: The eleventh international conference on learning representations; 2023.
35. Zhang M-L, Wu L. Lift: multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell. 2014;37(1):107–20.
36. Yang P, Sun X, Li W, Ma S, Wu W, Wang H. SGM: sequence generation model for multi-label classification. In: Proceedings of the 27th international conference on computational linguistics; 2018, pp. 3915–26.
37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in neural information processing systems; 2017.
38. Huang J, Li G, Huang Q, Wu X. Learning label-specific features and class-dependent labels for multi-label classification. IEEE Trans Knowl Data Eng. 2016;28(12):3309–23.
39. Ye C, Zhang L, He Y, Zhou D, Wu J. Beyond text: incorporating metadata and label structure for multi-label document classification using heterogeneous graphs. In: Proceedings of the 2021 conference on empirical methods in natural language processing; 2021. pp. 3162–71.
40. Li J, Li P, Hu X, Yu K. Learning common and label-specific features for multi-label classification with correlation information. Pattern Recogn. 2022;121: 108259.
41. Ma Q, Yuan C, Zhou W, Hu S. Label-specific dual graph neural network for multi-label text classification. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing; 2021. pp. 3855–64.
42. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies; 2019. pp. 4171–86.
43. Zhang Y, Li Z, Zhang M. Efficient second-order TreeCRF for neural dependency parsing. In: Proceedings of the 58th annual meeting of the association for computational linguistics; 2020. pp. 3295–305.
44. Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision; 2017.
45. Gong L, Cheng Q. Exploiting edge features for graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. pp. 9211–9.

46. Báez P, Villena F, Rojas M, Durán M, Dunstan J. The Chilean waiting list corpus: a new resource for clinical named entity recognition in Spanish. In: Proceedings of the 3rd clinical natural language processing workshop; 2020. pp. 291–300.
47. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020;36(4):1234–40.
48. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H. Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthc. 2021;3:1–23.
49. Akbik A, Bergmann T, Blythe D, Rasul K, Schweter S, Vollgraf R. Flair: an easy-to-use framework for state-of-the-art NLP. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations); 2019. pp. 54–9.
50. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.

Publisher's Note