TOE: A Grid-Tagging Discontinuous NER Model Enhanced by Embedding Tag/Word Relations and More Fine-Grained Tags

Jiang Liu, Donghong Ji, Jingye Li, Dongdong Xie, Chong Teng, Liang Zhao, *Senior Member, IEEE*, and Fei Li

***Abstract*—So far, discontinuous named entity recognition (NER) has received increasing research attention and many related methods have surged such as hypergraph-based methods, span-based methods, and sequence-to-sequence (Seq2Seq) methods, etc. However, these methods more or less suffer from some problems such as decoding ambiguity and efficiency, which limit their performance. Recently, grid-tagging methods, which benefit from the flexible design of tagging systems and model architectures, have shown superiority to adapt for various information extraction tasks. In this paper, we follow the line of such methods and propose a competitive grid-tagging model for discontinuous NER. We call our model TOE because we incorporate two kinds of Tag-Oriented Enhancement mechanisms into a state-of-the-art (SOTA) grid-tagging model that casts the NER problem into word-word relationship prediction. First, we design a Tag Representation Embedding Module (TREM) to force our model to consider not only word-word relationships but also word-tag and tag-tag relationships. Concretely, we construct tag representations and embed them into TREM, so that TREM can treat tag and word representations as queries/keys/values and utilize self-attention to model their relationships. On the other hand, motivated by the Next-Neighboring-Word (NNW) and Tail-Head-Word (THW) tags in the SOTA model, we add two new symmetric tags, namely Previous-Neighboring-Word (PNW) and Head-Tail-Word (HTW), to model more fine-grained word-word relationships and alleviate error propagation from tag prediction. In the experiments of three benchmark datasets, namely CADEC, ShARe13 and ShARe14, our TOE model pushes the SOTA results by about 0.83%, 0.05% and 0.66% in F1, demonstrating its effectiveness.**

***Index Terms*—Discontinuous named entity recognition, grid-tagging, tagging-oriented enhancement.**

Manuscript received 14 March 2022; revised 24 July 2022, 6 September 2022, and 6 October 2022; accepted 12 October 2022. Date of publication 10 November 2022; date of current version 2 December 2022. The work of Liang Zhao was supported in part by the Center for Artificial Intelligence under Grant C4AI-USP, in part by Sao Paulo Research Foundation (FAPESP) under Grant 2019/07665-4, in part by IBM Corporation, and in part by the China Branch of BRICS Institute of Future Networks. This work was supported in part by the National Natural Science Foundation of China under Grant 62176187, in part by the National Key Research and Development Program of China under Grant 2017YFC1200500, in part by the Research Foundation of Ministry of Education of China under Grant 18JZD015, in part by the Youth Fund for Humanities and Social Science Research of Ministry of Education of China under Grant 22YJCZH064, in part by the General Project of Natural Science Foundation of Hubei Province under Grant 2021CFB385, and in part by the Fundamental Research Funds for the Central Universities which is the research result of the Independent Scientific Research Project (humanities and social sciences) of Wuhan University. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sakriani Sakti. *(Corresponding author: Fei Li.)*

Jiang Liu, Donghong Ji, Jingye Li, Dongdong Xie, Chong Teng, and Fei Li are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: liujiang@whu.edu.cn; dhji@whu.edu.cn; theodorelee@whu.edu.cn; xie.dongdong@whu.edu.cn; tengchong@whu.edu.cn; foxlf823@gmail.com).

Liang Zhao is with the University of São Paulo, São Paulo 05508-220, Brazil (e-mail: zhao@usp.br).

Digital Object Identifier 10.1109/TASLP.2022.3221009

I. Introduction

NAMED entity recognition (NER) is a fundamental task for natural language processing (NLP), which is able to facilitate many other NLP tasks (e.g., question answering [1], entity relationship extraction [2]). NER has been extensively studied and researchers have come up with numerous effective methods [3], [4], [5], [6], [7]. Previously, most methods [8], [9], [10], [11], [12], [13] treat it as a sequential marking problem, in which each token is assigned with a tag representing its entity type. Their basic assumption is that entity mentions should be short text spans [14] and should not overlap with each other. Although this assumption is valid in most cases, it is not always true, especially in clinical corpora. [15]. As shown in Fig. 1, the two entities consist of several discontinuous segments and some segments are overlapped. Therefore, it is necessary to design models that can recognize both flat entities and discontinuous entities.

To achieve this goal, some recent studies have developed some models for discontinuous NER, which can be roughly divided into the following categories: **1)** Sequence-tagging-based methods [16] extend the BIO tag scheme to more complex tag schemes such as BIOHD, but such ad hoc design is not flexible enough to handle all the situations. **2)** Hypergraph-based methods represent all entity segments as graph nodes and learn to combine these nodes with individual classifiers, but such methods suffer from the false structure and structural ambiguity in the prediction process. **3)** Seq2Seq-based methods [17], [18] generate various entities directly, which unfortunately may suffer from decoding efficiency issues and certain common pitfalls of the Seq2Seq architecture, such as exposure bias. **4)** Span-based methods [19] list all possible spans and classify them according to the level of spans. However, these methods are limited by the maximum span length and result in considerable computational complexity due to span enumeration.

Recently, grid-tagging-based methods achieve promising performance for discontinuous NER. Wang et al. (2021) [20] predict the entity boundaries and entity word relationships respectively through two grids and then decode the whole entities from the

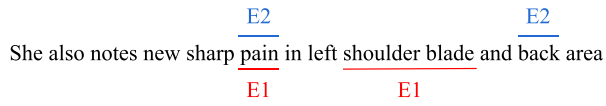

Fig. 1. An example to show two discontinuous entities in a sentence of the clinical corpus [15].

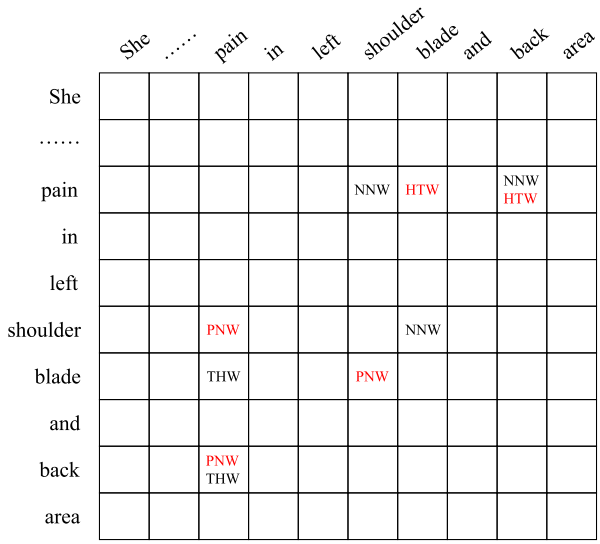

Fig. 2. An example to show our extended tag system. The red tags are new and the black ones are used in the SOTA model [21]. Columns and rows represent the subjects and objects of the relationships, where a subject indicates the tail of the relationship arc and an object indicates the head. For more details refer to Section III-E.

entity segment graph through maximal clique discovery. The latest state-of-the-art (SOTA) method, proposed by Li et al. 2022 [21], is also based on grid tagging. It transforms discontinuous NER into the word-word relationship recognition problem, and utilizes one grid to include all word-word relationships.

In this paper, we follow the line of grid tagging for discontinuous NER and propose two tag-oriented enhancements to optimize the SOTA model. First, since the SOTA model only pays attention to the relationships between words, we embed tag representations into the model in order to model the relationships between each pair of tags and the relationships between tags and words. These relationships are also important because entity mentions can be correctly identified only if all the tags have been predicted. Second, we add two tags, namely `Previous-Neighboring-Word` (**PNW**) and `Head-Tail-Word` (**HTW**), to its tag system (including only two tags, `Next-Neighboring-Word` (**NNW**) and `Tail-Head-Word` (**THW**)), in order to model more fine-grained word-word relationships. As shown in Fig. 2, when the model fails to recognize the HTW relationship between "pain" and "blad", their THW relationship can still be recognized to compensate for such error.

To implement our above idea, we build a grid-tagging model with Tag-Oriented Enhancement (**TOE**). The main framework of the model is shown in Fig. 3. First, word representations are generated by BERT [22] and BiLSTM [23]. Then, we construct a tag representation embedding module (**TREM**) to embed tag representations into our model. TREM employs a convolution layer to represent each tag as a two-dimensional (2D) table, because the tag in our task denotes the relationship of each word pair. Then TREM uses self-attention to model the relationships between each pair of tags and the relationships between tags and words. TREM can run multiple times to iteratively mix tag representations with word representations. Finally, a co-predictor predicts the four word-word relationships that we defined in this paper, namely PNW, HTW, NNW and THW, using the word and tag representations jointly. After that, discontinuous entities can be decoded out from these tags.

We conduct experiments on three datasets, namely CADEC [5], ShARe13 [2] and ShARe14 [24]. Results show that our model achieves the best performance on all datasets and outperforms eight baselines including the SOTA model. Our contributions can be summarized as:

- We propose a novel tag representation embedding module (TREM) to inject tag features and model the relationships across tags and words.
- We extend the tag system of the SOTA model to model more fine-grained word-word relationships and reduce error propagation.
- Our model achieves the SOTA performance on 3 benchmark datasets. We conduct substantial experiments on 3 datasets to analyze and understand our model.[1]

II. Related Work

We summarize the related methods for discontinuous NER in the following categories.

Sequence-tagging-based methods: In the field of NLP, NER is usually considered as a sequence tagging problem [25], [26], [27]. Based on well-designed features, CRF based models have achieved leading performance [28], [29], [30]. Recently, neural network models have been used for feature representation [31], [32]. In addition, upper and lower cultural lexical representations such as ELMo [33], Flair [34] and BERT [22] have also achieved great success. For NER, the end-to-end bidirectional LSTM-CRF model [23], [35], [36] is a representative architecture. These models can only recognize regularly named entities. Tang et al. (2018) [16] extended the BIO tagging scheme to BIOHD to solve the problem of discontinuous mention. Even so, there is still the problem of decoding ambiguity.

Hypergraph-based methods: Lu and Roth (2015) [37] first proposed a model based on hypergraph method to solve NER, and expressed possible references by exponential method. Subsequent studies [14], [38], [39] also developed and improved the method. For example, Muis and Lu (2018) [14] used this method to deal with discontinuous NER, and Wang and Lu (2018) [39] used a deep neural network to strengthen the hypergraph model.

Seq2Seq-based methods: Gillick et al. (2015) [40] was the first to use the Seq2Seq model to solve NER. The model takes the original sentence as the input and takes the head and tail position, span length and entity type of all entities as the output.

[1]The code is publicly available at https://github.com/solkx/TOE.git

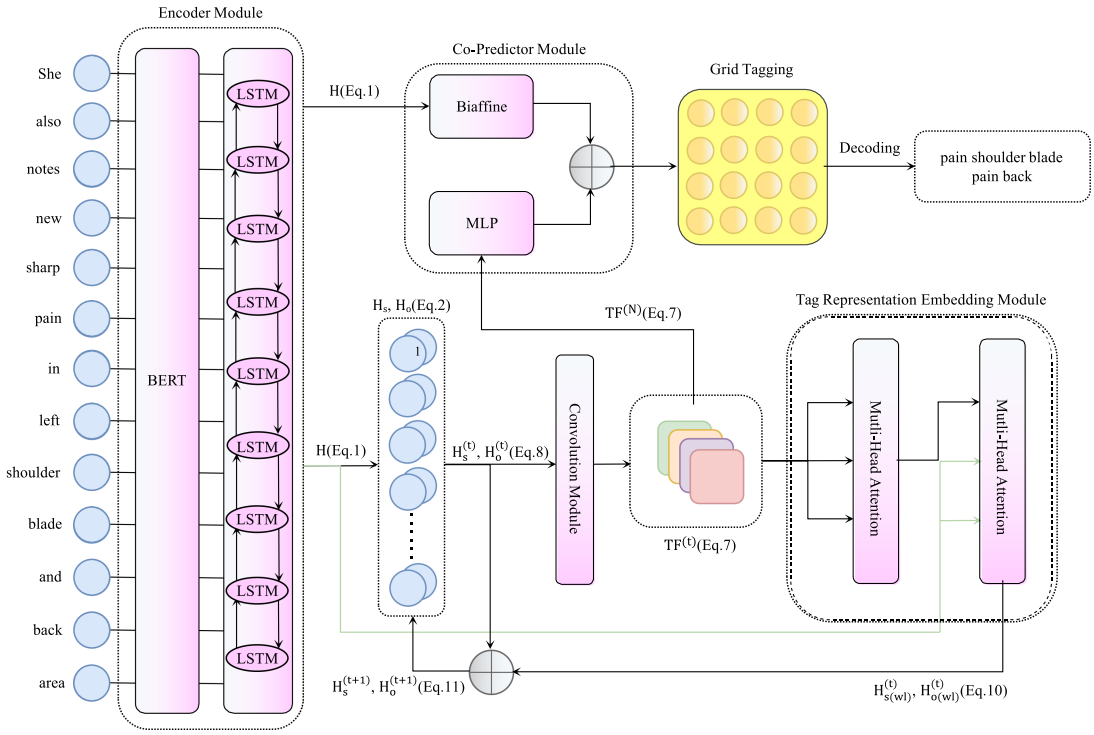

Fig. 3. The overall architecture of our TOE model. H denotes the word representations. $TF^{(t)}$ denotes the tag-aware grid features, where t means that the tag representation embedding process may run several iterations. $\bigoplus$ represents the element-level summation.

Fei et al. (2021) [18] combined Seq2Seq and pointer network to deal with discontinuous NER. A recent study [17] deals with all types of NER through the Seq2Seq model of pointer network based on BART [41], and generates the index and type sequence from the beginning to the end of all possible entities. However, the Seq2Seq model has potential decoding efficiency problems and exposure bias problems.

Span-based methods: Other studies deal with NER by identifying entity spans, that is, enumerating all possible entity spans, removing invalid entity spans or entity types, and finally retaining the final prediction results [42], [43]. Li et al. (2020a) [44] redefine NER as a machine reading comprehension (MRC) task, ask questions for different entity types and extract entities according to the corresponding answers. Li et al. (2021a) [19] convert the discontinuous NER to find the complete subgraph from the span-based entity segment graph, and obtain the competitive result. Unfortunately, due to enumeration, the effect of these methods is affected by the maximum span length and has considerable complexity, especially for longer entities.

Grid-tagging-based methods: Recently, the method based on grid marking [20], [21] has had a good performance. It transforms sentences into 2D tables. The method in Wang et al. (2021) [20] includes three steps: 1) identifying the span of entity segments by marking the head and tail words in the table; 2) Extracting the relationship between entity segment span pairs by marking the head and tail words in another table; 3) The integrated entities are decoded from the entity segment span graph through maximum clique discovery. In contrast, the most advanced method [21] uses a simpler process (two relationships, one table, no block decoding) and is more end-to-end, reducing error propagation. It obtains the features between words through the convolution layer, tags them on the grid, and identifies all possible entities through neighbor words relationships and head-tail relationships. The biggest difference between this method and other previous methods is that it focuses on the relationship between words rather than more accurate entity boundary recognition. In addition, the grid marking method can better avoid the disadvantages of some other methods, such as the disadvantages of sequence-to-sequence method and span based method.

The differences between our model and previous models: Our model follows the SOTA model [21] for discontinuous NER, which is also based on grid tagging. However, the differences include: 1) We design a new module to embed tag representations into our model to enhance the interactions between tags and words. 2) We extend the tag system in [21] with two additional tags to model more fine-grained word-word relationships.

III. Methodology

We define the discontinuous NER task as a grid tagging problem and identify all possible entities through four predefined tags. Our model architecture is shown in Fig. 3. It is mainly composed of four components and a tag system. The four components are the encoder module, the convolution module, the tag representation embedding module and the co-predictor module. Firstly, the encoder module is composed of a pre-training language model BERT [22] and a bidirectional LSTM [23], which is used to generate the word representation of upper and

lower culture from the input sentence. Then, the representation of multiple words on the grid is established and refined through the convolution module. Then, the tag features are captured by self-attention mechanism in TREM. The convolution module and TREM undergo multiple iterations to obtain more detailed features. Then, the co-predictor module [45] is used to jointly infer the relationship between all word pairs. Finally, all possible entities are obtained by decoding.

A. Encoder Module

We use BERT [22] as the text encoder of our model. Give an input sentence $X = \{x_1, x_2, \ldots, x_N\}$, and input them into a pre-trained BERT. The BERT encoded by the multi-layer self-attention structure outputs the context representation of each context tag. To further enhance context modeling, we adopted bidirectional LSTM [23] based on previous work [19], [46]. After BERT encoding, the sentence X can be represented as:

$$\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N\}, \tag{1}$$

where $\boldsymbol{h}_i \in \mathbb{R}^{d_h}$ is the representation of the i-th word and d_h represents the dimension of a word representation.

B. Convolution Module

Since the relationships between words in this paper are directional, each word plays either a subject or object role in one relationship. The subject indicates the tail of the relationship arc and the object indicates the head. As shown in Fig. 2, subjects and objects correspond to the elements in the column and row respectively. We transform word representations into the subject and object spaces as below:

$$\begin{aligned} \boldsymbol{H}_s &= \boldsymbol{W}_1\boldsymbol{H} + \boldsymbol{b}_1 = \{\boldsymbol{h}_1^s, \boldsymbol{h}_2^s, \ldots, \boldsymbol{h}_N^s\}, \\ \boldsymbol{H}_o &= \boldsymbol{W}_2\boldsymbol{H} + \boldsymbol{b}_2 = \{\boldsymbol{h}_1^o, \boldsymbol{h}_2^o, \ldots, \boldsymbol{h}_N^o\}, \end{aligned} \tag{2}$$

where $\boldsymbol{h}_i^s, \boldsymbol{h}_i^o \in \mathbb{R}^{d_h}$ represent the subject and object representations of the i-th word, $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d_h \times d_h}$ and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{d_h}$ are trainable weights and biases respectively.

The convolution module is then used as a representation refiner. Firstly, the Conditional Normalization Layer (CLN) [47] is used to generate the representation of words on the grid, which can be regarded as a three-dimensional matrix $\boldsymbol{V} \in \mathbb{R}^{N \times N \times d_h}$:, in which each element in the V_{ij} grid represents a word pair (x_i, x_j):

$$\boldsymbol{V}_{ij} = CLN(\boldsymbol{h}_i^s, \boldsymbol{h}_j^o) = \gamma_{ij} \odot \left(\frac{\boldsymbol{h}_j^o - \mu}{\sigma}\right) + \lambda_{ij}, \tag{3}$$

where $\boldsymbol{h}_i^s$ is the condition of the normalized gain parameters $\gamma_{ij} = \boldsymbol{W}_\alpha \boldsymbol{h}_i^s + \boldsymbol{b}_\alpha$ and $\lambda_{ij} = \boldsymbol{W}_\beta \boldsymbol{h}_i^s + \boldsymbol{b}_\beta$. $\boldsymbol{W}_\alpha, \boldsymbol{W}_\beta \in \mathbb{R}^{d_h \times d_h}$ and $\boldsymbol{b}_\alpha, \boldsymbol{b}_\beta \in \mathbb{R}^{d_h}$ are trainable weights and biases respectively. μ and σ are the mean and standard deviation across the elements of $\boldsymbol{h}_j^o$.

Then the grid representations are enriched by adding the relative word position information $\boldsymbol{E}^{wp} \in \mathbb{R}^{N \times N \times d_{wp}}$ between each pair of words and the grid position information $\boldsymbol{E}^{gp} \in \mathbb{R}^{N \times N \times d_{gp}}$ that distinguishes the upper and lower triangular areas, and then mix with the word pair information $\boldsymbol{V} \in \mathbb{R}^{N \times N \times d_h}$ to obtain the position area perception representation $\boldsymbol{C} \in \mathbb{R}^{N \times N \times d_c}$ through a multi-layer perceptron (MLP):

$$\boldsymbol{C} = MLP_1([\boldsymbol{V}; \boldsymbol{E}^{wp}; \boldsymbol{E}^{gp}]). \tag{4}$$

Afterwards, the multiple 2D dilated convolutions (DConv) with different dilation rates are used to capture the interactions between the words with different distances, formulated as:

$$\boldsymbol{Q} = GeLU(DConv(\boldsymbol{C})), \tag{5}$$

where $\boldsymbol{Q} \in \mathbb{R}^{N \times N \times d_q}$ is the output and $GeLU$ is a activation function [48].

C. Tag Representation Embedding Module (TREM)

The TREM module is used to embed the tag representations into our model in order to model the interactions between tags as well as tags and words: First, we generate the tag-aware grid feature $\boldsymbol{TF}_l \in \mathbb{R}^{N \times N \times d_t}$ by mapping the grid representation $\boldsymbol{Q}$ into the tag space. Specifically, for the element (i, j) in the grid corresponding to the word pair (x_i, x_j), we generate its tag-aware feature as:

$$\boldsymbol{TF}_l(i, j) = \boldsymbol{W}_l \boldsymbol{Q}_{ij} + \boldsymbol{b}_l, \tag{6}$$

where $\boldsymbol{W}_l \in \mathbb{R}^{d_t \times d_q}$ and $\boldsymbol{b}_l \in \mathbb{R}^{d_t}$ are trainable weights and biases.

Since there are four kinds of tags in this paper, namely NNW, PNW, HTW and THW (cf. Section III-E), we concatenate them together as below:

$$\boldsymbol{TF}^{(t)} = [\boldsymbol{TF}_{NNW}^{(t)}; \boldsymbol{TF}_{PNW}^{(t)}; \boldsymbol{TF}_{HTW}^{(t)}; \boldsymbol{TF}_{THW}^{(t)}], \tag{7}$$

where t means that the TREM module may run several times to refine $\boldsymbol{TF} \in \mathbb{R}^{N \times N \times 4d_t}$. Theoretically, the number M_{num} of tag space mappings can be smaller or larger than the number of tags, because our formulations in (6) and (7) are not constrained by this number. However, we set M_{num} the same as the number of tags heuristically since we consider $\boldsymbol{TF}_l$ as a tag representation. We will empirically show the rationality of such method in the experiments (cf. Table IV).

We input $\boldsymbol{TF}^{(t)}$ into the max-pooling layers ($Maxpool_1$, $Maxpool_2 \in \mathbb{R}^{N \times 4d_t}$) and FFN layers to recover the subject and object word features $\boldsymbol{H}_s^{(t)}$ and $\boldsymbol{H}_o^{(t)}$ at the t-th iteration:

$$\begin{aligned} \boldsymbol{H}_s^{(t)} &= Maxpool_1(\boldsymbol{TF}^{(t)})\boldsymbol{W}_s + \boldsymbol{b}_s, \\ \boldsymbol{H}_o^{(t)} &= Maxpool_2(\boldsymbol{TF}^{(t)})\boldsymbol{W}_o + \boldsymbol{b}_o. \end{aligned} \tag{8}$$

where $\boldsymbol{W}_s, \boldsymbol{W}_o \in \mathbb{R}^{4d_t \times d_h}$ and $\boldsymbol{b}_s, \boldsymbol{b}_o \in \mathbb{R}^{d_h}$ are trainable weights and biases. $Maxpool_1$ and $Maxpool_2$ merge the representations $\boldsymbol{TF}^{(t)}$ along the rows and columns of the table respectively, so as to restore the subject and object word representations, $\boldsymbol{H}_s^{(t)}$ and $\boldsymbol{H}_o^{(t)}$.

Then we use the multi-head self-attention [49] to mine the relationships between these tag-aware word representations:

$$\begin{aligned} \boldsymbol{H}_{s(ll)}^{(t)} &= MultiHeadAttention(\boldsymbol{H}_s^{(t)}, \boldsymbol{H}_s^{(t)}, \boldsymbol{H}_s^{(t)}), \\ \boldsymbol{H}_{o(ll)}^{(t)} &= MultiHeadAttention(\boldsymbol{H}_o^{(t)}, \boldsymbol{H}_o^{(t)}, \boldsymbol{H}_o^{(t)}), \end{aligned} \tag{9}$$

Algorithm 1: Decoding Algorithm.

Input: The relationships $R \in \mathbb{R}^{N \times N \times l_n}$ of all the word pairs, where l_n is the number of word relationship tags. R_{ij}^{l} indicates that the word pair (x_i, x_j) has an l relationship, where $i, j \in [1, N]$.
Output: Entity set E.
1: $E = []$
2: **for** $R_{ij}^{l} \in R$ and $j \geq i$ **do**
3: **if** $Exist(R_{ji}^{THW})$ **or** $Exist(R_{ij}^{HTW})$ **then**
4: $S = [i]$ // Store the head word index
5: **if** $i = j$ **then** // Entity contains only one word
6: $E.add(S)$ // Store the entity span to E
7: **else** // Find next entity word
8: **for** $k \in (i, j]$ **do**
9: $FindNext(S, R_{ik}^{NNW}, R_{ki}^{PNW}, k, j, E)$
10: **return** E
11: // Find next entity word m based on r_1 and r_1
12: **function** $FindNext(S, r_1, r_2, m, e, E)$
13: **if** $Exist(r_1)$ and $Exist(r_2)$ **then**
14: $S.add(m)$ // Add the next word index
15: **if** $m = e$ **then** // Next word is the tail word
16: $E.add(S)$ // Store the entity span to E
17: **else** // Recursively find next entity word
18: **for** $k \in (m, e]$ **do**
19: $FindNext(S, R_{mk}^{NNW}, R_{km}^{PNW}, k, e, E)$

and another multi-head self-attention to mine the relationships between the original word representations and these tag-aware word representations:

$$\boldsymbol{H}_{s(wl)}^{(t)} = MultiHeadAttention(\boldsymbol{H}_{s(ll)}^{(t)}, \boldsymbol{H}_s, \boldsymbol{H}_s),$$
$$\boldsymbol{H}_{o(wl)}^{(t)} = MultiHeadAttention(\boldsymbol{H}_{o(ll)}^{(t)}, \boldsymbol{H}_o, \boldsymbol{H}_o). \quad (10)$$

Since the TREM module may run several times to iteratively refine the tag-aware representations, we add a residual connection [50] to alleviate the gradient vanishment problem:

$$\boldsymbol{H}_s^{(t+1)} = LayerNorm(\boldsymbol{H}_s^{(t)} + \boldsymbol{H}_{s(wl)}^{(t)}),$$
$$\boldsymbol{H}_o^{(t+1)} = LayerNorm(\boldsymbol{H}_o^{(t)} + \boldsymbol{H}_{o(wl)}^{(t)}), \quad (11)$$

where these new features are fed back to the convolution module for next iteration.

D. Co-Predictor Module

After the TREM, we get the tag-aware grid features $\boldsymbol{TF}^{(N)}$ for each word pair. These features are fed into an MLP to predict the relationships between each pair of words. In addition, we enhance the relational classification by combining the MLP predictor with a biaffine predictor. Therefore, we take these two predictors to calculate the two independent relationship distributions (x_i, x_j) of word pairs at the same time, and combine them as the final prediction. For MLP, its input is the output $\boldsymbol{TF}^{(N)}$ of TREM, so the relationship score of each word pair (x_i, x_j) is calculated as:

$$\boldsymbol{y}'_{ij} = MLP_2(\boldsymbol{TF}^{(N)}(i, j)), \quad (12)$$

The input of the biaffine predictor is the output $\boldsymbol{H}$ of the encoder layer, which can be considered as a residual connection [50]. Two MLPs are used to calculate the representation of each word in the word pair (x_i, x_j). Then, the relationship score between word pairs (x_i, x_j) is calculated using a biaffine classifier [51]:

$$\boldsymbol{y}''_{ij} = \boldsymbol{s}_i^{\top} \boldsymbol{U} \boldsymbol{o}_j + \boldsymbol{W}[\boldsymbol{s}_i; \boldsymbol{o}_j] + \boldsymbol{b}, \quad (13)$$

where $\boldsymbol{U}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ are trainable parameters, and $\boldsymbol{s}_i = MLP_3(\boldsymbol{h}_i^s)$ and $\boldsymbol{o}_j = MLP_4(\boldsymbol{h}_j^o)$ represent the subject and object representations respectively. Finally, we combine the scores from the MLP and biaffine predictors to get the final score:

$$\boldsymbol{y}_{ij} = Softmax(\boldsymbol{y}'_{ij} + \boldsymbol{y}''_{ij}). \quad (14)$$

E. Our Tagging System

In the SOTA model [21], two kinds of tags are predicted:

- `Next-Neighboring-Word` (NNW) indicates that the word pair (x_i, x_j) belongs to an entity, and the next word of x_i in the entity is x_j.
- `Tail-Head-Word` (THW) indicates that the word in the row of the grid is the tail of the entity, and the word in the column of the grid is the head of the entity.

Although such tagging design is effective, it has some drawbacks. For example, when the model misses a THW relationship, it will fail to recognize the corresponding entity, which cannot be recovered. Moreover, we believe that although their tagging design is elegant, it results in a sparse tag distribution in the grid and thus loses certain word-word relationships. To enhance the tagging system and model more fine-grained word-word relationships, we propose two new tags:

- `Previous-Neighborhood-Word` (PNW) indicates that the word pair (x_i, x_j) belongs to an entity. The previous word of x_i in the entity is x_j.
- `Head-Tail-Word` (HTW) indicates that the word in the row of the grid is the head of the entity, and the word in the column of the grid is the tail of the entity.

By using these tags, we can model fine-grained word-word relationships and compensate certain error propagation from the model prediction. For example, we jointly predict the NNW and PNW relationships, and when both of them exist, we think that the word pair belongs to the same entity. Similarly, we jointly predict the THW and HTW relationships and when one of them exists, we think that the word pair is the head and tail of an entity. The advantage of using this decoding strategy will be shown in the ablation studies (cf. Table IV).

Moreover, we show the pseudo-code of using this decoding strategy in Algorithm 1. This decoding algorithm is mostly similar to the one used in Li et al. (2022) [21], while the differences exist in finding the head entity words (line 3) and non-head entity words (line 9). Because we add two new tags, PNW and HTW, the condition of head entity words changes from "THW" to "HTW or THW" and the condition

TABLE I
STATISTICS OF THREE DATASETS

	CADEC				ShARe13				ShARe14			
	All	Train	Dev	Test	All	Train	Dev	Test	All	Train	Dev	Test
#Sentences	7,597	5,340	1,097	1,160	18,767	8,508	1,250	9,009	34,614	17,407	1,361	15,850
#Entities	6,318	4,430	898	990	11,148	5,146	669	5,333	19,047	10,354	771	7,922
#Discontinuous	679	491	94	94	1,088	581	71	436	1,650	1,004	80	566
%Discontinuous	10.7	11.1	10.5	9.5	9.8	11.3	10.6	8.2	8.7	9.7	10.4	7.1

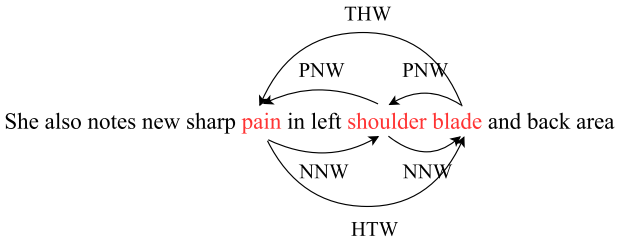

Fig. 4. An example to show the process of recognizing "pain shoulder blade".

of non-head entity words changes from "NNW" to "NNW and PNW".

Based on this decoding algorithm, we also give an example in Fig. 4 to explain the process of recognizing "pain shoulder blade". By using the NNW relationship with the subject "pain" and object "shoulder" and the PNW relationship with the subject "shoulder" and object "pain," we recognize "pain shoulder" as a part of the entity. Similarly, "shoulder blade" is also recognized in the same way. Then, by using the HTW and THW relationships, we recognize "pain" and "blade" are the head and tail of the entity, so that "pain shoulder blade" can be recognized completely.

F. Learning

As shown in Fig. 2, we can see that there may be more than one relationship between each pair of words. Therefore, we adopt a cross-entropy loss function that is extended for multi-label classification [52]. In order to predict the correct tag, we need the score of each target tag to be no less than that of each non-target tag. In addition, we define a threshold so that the scores of target classes are greater than the threshold, and the scores of non-target classes are less than the threshold. The final loss function is:

$$\begin{aligned}\mathcal{L} &= log\left(1+\sum_{n,m} e^{\hat{y}^n_{(i,j)}-y^m_{(i,j)}}\right.\\ &\quad\left.+\sum_{n} e^{\hat{y}^n_{(i,j)}-y_0}+\sum_{m} e^{y_0-y^m_{(i,j)}}\right),\\ &= log\left(e^{y_0}+\sum_{n} e^{\hat{y}^n_{(i,j)}}\right)+log\left(e^{-y_0}+\sum_{m} e^{-y^m_{(i,j)}}\right),\end{aligned} \tag{15}$$

where $n \in \Omega_{neg}$, $m \in \Omega_{pos}$, and Ω_{neg}, Ω_{pos} are the non-target and target tag sets respectively. $\hat{y}^n_{(i,j)}$ and $y^m_{(i,j)}$ are the non-target and target tag scores respectively. y_0 represents the threshold. This loss function is similar to the circle loss [53].

IV. EXPERIMENT SETTING

A. Datasets

In order to evaluate our model, we conducted experiments on three discontinuous NER datasets, namely CADEC [5], ShARe13 [2] and ShARe14 [24], all of which come from the documents in biomedical or clinical fields. They all contain only one entity type, in which the entity type in CADEC is `ADR`, and the entity type in ShARe13 and ShARe14 is `Disease_Disorder`. We use the preprocessing script provided by Dai et al. (2020) [54] for dataset segmentation. In these discontinuous NER datasets, discontinuous entities account for about 10% of the total entities. The statistics of these datasets are shown in Table I.

B. Baselines

Sequence-tagging-based methods assign a tag to each token with different tag schemes, such as BIOHD [16]. **Span-based methods** enumerate all possible spans and combines them into entities [19]. **Hypergraph-based methods** use hypergraphs to represent and infer entity mention [55]. **Seq2Seq-based methods** directly generate the word sequences of the entities at the decoder side [17], [18]. **Grid-tagging-based methods** assign a tag for each pair of words and entities can be decoded out from these tags [20], [21]. We also compare with other methods that cannot be grouped into the above categories, such as the transition-based method [54].

C. Evaluation Metrics

Our evaluation metrics follow previous work [17], [37], [56], using the precision (P), recall (R) and F1. If the token sequence and type of a predicted entity are exactly the same as those of a gold entity, the predicted entity is regarded as true-positive. We run each experiment three times and report their average value.

D. Implementation Details

Our hyper-parameter settings are given in Table II. The hyper-parameters are adjusted according to the fine-tuning on the development sets. In addition, since the datasets come from different fields, we use different pre-trained langauge models to generate word representations. For CADEC, we use BioBERT [57], and for ShARe13 and ShARe14, we use ClinicalBERT [58]. Moreover, we use AdamW [59] as the optimizer. Our model is implemented using PyTorch and trained using NVIDIA RTX 3090 GPU.

TABLE II
HYPER-PARAMETER SETTINGS

Hyper-parameter	value
d_h	768
d_{wp}	20
d_{gp}	20
d_q	64, 80, 96, 128
y_0	0
Dropout	0.1, 0.3, 0.5
Learning rate (BERT)	5e-6
Learning rate (others)	1e-3
Batch size	12, 16
Warm factor	0, 0.1, 0.4
Rounds (TREM iterations)	3

V. RESULTS AND ANALYSES

A. Comparisons With the Baselines

The main results of the baseline models and our model in three discontinuous NER datasets are shown in Table III. We can observe that our model has the best results (F1 values) on the three datasets, which is due to the fact that it not only captures the relationships between words, but also pays attention to the relationships between words and tags and the relationships between tags and tags. In addition, we expand two new tags, which can complement the prediction of neighbor words and head-tail words. As a result, the performance on CADEC, ShARe13 and ShARe14 is improved by 0.83%, 0.05% and 0.66% compared to the previous state-of-the-art ones.

B. Ablation Studies

In addition, we also conduct corresponding ablation experiments, in order to verify the effectiveness of the modules in our proposed model and understand their impacts. The experimental results are shown in Table IV. First of all, if the TREM module is deleted, the F1s of our model on three datasets decrease by 1.24%, 0.59% and 1.03% respectively. This shows that the TREM is effective, that is, the relationships between words and tags are conducive to the prediction of tags.

Then, we also investigate the effect of the iteration number in TREM. As can be seen from the table, no matter reducing the number of iterations (e.g. rounds $= 2$) or increasing the number of iterations (e.g. rounds $= 4$), the performance of our model on all three datasets decrease. Specifically, when we use 2 rounds, the F1 decreases by 0.79%, 1.19% and 0.78% respectively. Similarly, when 4 rounds are used, the F1 also decreases by 1.81%, 1.16% and 0.48% respectively.

After that, we conduct experiments to compare the effect of using the tags in the SOTA model [21] (NNW+THW) and the ones that we propose in this paper (PNW+HTW). As shown in Table IV, the NNW+THW and PNW+HTW tagging strategies achieve almost the same F1s, which is explainable because they entail the same neighbor-word and head-tail relationships but implement these relationships in different directions. The two new tags that we have added are kinds of effective complements to the previous tags. Therefore, the model that combines both of them achieves the best performance on all three datasets.

Next, we investigate the effect of using different decoding methods. As shown in Table IV, the "T(NNW, PNW) L(THW, HTW)" decoding method performs the best. We assume that the reason for this observation is that NNW and PNW relationships occur much more than THW and HTW relationships. Thus, the model is apt to predict more NNW and PNW relationships but fewer THW and HTW relationships. Therefore, it is necessary to tighten the establishing condition of neighbor word relationships by using the logic "AND" between NNW and PNW, but loose the establishing condition of head-tail word relationships by using the logic "OR" between THW and HTW. As seen, the "L(NNW, PNW) T(THW, HTW)" decoding method, which is the opposite of the "T(NNW, PNW) L(THW, HTW)" decoding method, performs the worst in all three datasets, further verifying our assumption.

Finally, we also investigate the effect of the tag mapping number. We test 4 values for the tag mapping number M_{num}, namely 2, 4, 6, 8. We find that when the tag mapping number is set to 4, which is the same as the number of tags, our model achieves the best performance. Such an observation is consistent with our intuition that the tag mapping number should be equal to the tag number. This may be because each tag mapping represents a tag, more or fewer tag mappings may bring troubles to the model for aligning tag representations with tags.

C. Effectiveness on Recognizing Discontinuous Entities

Fig. 5(a)–(c) show the results using sentences containing at least one discontinuous entity. First, our model TOE achieves better results than other grid-tagging models such as Wang et al. (2021) [20] and Li et al. (2022) [21]. This demonstrates the superiority of our model, where both the word and tag relationships are leveraged. In addition, our model also performs better than other kinds of baselines such as the transition-based system [54] and end-to-end generative model [17]. Moreover, as shown in Fig. 5(d)–(f), when only comparing the performance of recognizing discontinuous entities, our model still obtains the best results on all datasets. In conclusion, our model has the advantage of identifying discontinuous entities, which contributes to overall performance improvement.

D. Performance Analysis of Recognizing the Entities With Different Overlapped Types

As mentioned in the previous section, discontinuous entities and overlapped entities basically exist at the same time in the three datasets. In order to analyze the ability of our model to extract various overlapping entities, we divide them into four categories according to previous work [20], [54], i.e., "no overlapped," "left overlapped," "right overlapped" and "multiple overlapped". Table V shows an example for each overlapped category. As shown in Table VI, most of the results of our model are optimal compared with the baseline models, with regards to different overlapped types. This shows that our model is more competitive and more adaptive in extracting various overlapping entities. In addition, in some cases, the F1s of our model and the baseline models are 0. This may be because the number of such entities is significantly smaller in the datasets, compared to the

TABLE III
PERFORMANCE COMPARISONS BETWEEN THE BASELINE MODELS AND OUR MODEL ON THREE DATASETS. THE BOLD NUMBER REPRESENTS THE HIGHEST RESULT IN EACH COLUMN. WE ALSO PERFORM THE SIGNIFICANCE TEST ON THE F1S OF OUR MODEL AND THE SOTA MODEL [21] ON THE DEVELOPMENT SET AND TEST SET

MODEL		CADEC			ShARe13			ShARe14		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Sequence Tagging	Tang et al. (2018) [16]	67.80	64.99	66.36	—	—	—	—	—	—
Hypergraph-based	Wang and Lu (2019) [55]	72.10	48.40	58.00	83.80	60.40	70.30	79.10	70.70	74.70
Seq2Seq	Yan et al. (2021) [17]	70.08	71.21	70.64	82.09	77.42	79.69	77.20	**83.75**	80.34
	Fei et al. (2021) [18]	75.50	71.80	72.40	**87.90**	77.20	80.30	—	—	—
Span-based	Li et al. (2021a) [19]	—	—	69.90	—	—	82.50	—	—	—
Others	Dai et al. (2020) [54]	68.90	69.00	69.00	80.50	75.00	77.70	78.10	81.20	79.60
Grid Tagging	Wang et al. (2021) [20]	70.50	**72.50**	71.50	84.30	78.20	81.20	78.70	82.15	80.39
	Li et al. (2022) [21]	74.09 (70.59)	72.35 (**68.45**)	73.21 (69.50)	85.57 (80.20)	79.68 (**77.83**)	82.52 (78.99)	79.88 (82.29)	83.71 (81.37)	81.75 (81.82)
	TOE (ours)	**77.77** (**74.22**)	70.66 (67.79)	**74.04*** (**70.86***)	85.18 (**81.38**)	**80.12** (77.59)	**82.57** (**79.42***)	**82.26** (**83.77**)	82.57 (**82.22**)	**82.41*** (**82.98***)

*Denotes significance at $p < 0.05$. The numbers in parentheses mean the model results on the development sets.

TABLE IV
ABLATION EXPERIMENTS. WE REPORT THE MODEL PERFORMANCE WHEN DELETING SOME MODULES SUCH AS TREM AND EXTENDING TAGS, OR CHANGING SOME CONFIGURATIONS SUCH AS THE ROUNDS OF TREM, DECODING METHODS AND TAG MAPPING NUMBER M_{num}. $T(l_1, l_2)$ INDICATES THAT WE THINK THE WORD RELATIONSHIP REALLY EXISTS WHEN BOTH l_1 AND l_2 EXIST. $L(l_1, l_2)$ INDICATES THAT WE THINK THE WORD RELATIONSHIP REALLY EXISTS WHEN EITHER ONE OF l_1 AND l_2 EXIST. BEST SETTING: T(NNW, PNW), L(THW, HTW); ROUNDS $= 3$; $M_{num} = 4$. THE NUMBERS IN PARENTHESES MEAN THE MODEL RESULTS ON THE DEVELOPMENT SETS

	CADEC			ShARe13			ShARe14		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Best Setting	77.77 (**74.22**)	70.66 (67.79)	**74.04** (**70.86**)	85.18 (81.38)	80.12 (77.59)	**82.57** (**79.42**)	82.26 (83.77)	82.57 (**82.22**)	**82.41** (**82.98**)
w/o TREM	73.44 (69.69)	72.19 (**69.86**)	72.80 (69.77)	84.85 (79.67)	79.31 (77.44)	81.98 (78.53)	79.50 (82.24)	83.36 (79.90)	81.38 (81.06)
Rounds = 2	75.75 (71.41)	70.93 (69.15)	73.25 (70.26)	85.84 (81.07)	77.55 (75.98)	81.38 (78.31)	79.91 (82.99)	83.44 (80.52)	81.63 (81.73)
Rounds = 4	75.84 (72.88)	69.12 (67.35)	72.23 (69.95)	84.80 (79.22)	78.36 (77.24)	81.41 (78.11)	81.00 (84.15)	82.91 (79.07)	81.93 (81.53)
NNW+THW	73.78 (69.66)	**72.62** (69.73)	73.20 (69.69)	84.33 (79.26)	79.68 (**78.08**)	81.94 (78.67)	77.72 (81.82)	**84.97** (82.19)	81.18 (81.99)
PNW+HTW	74.91 (70.54)	71.47 (68.50)	73.12 (69.49)	84.89 (80.20)	79.46 (77.83)	82.07 (78.99)	78.81 (82.29)	84.27 (81.37)	81.44 (81.82)
T(NNW,PNW),T(THW,HTW)	**78.10** (73.35)	68.37 (66.33)	72.90 (69.65)	**86.07** (**81.47**)	79.13 (76.28)	82.45 (78.77)	**83.02** (**86.06**)	81.72 (77.58)	82.37 (81.60)
L(NNW,PNW),T(THW,HTW)	77.93 (73.33)	67.38 (66.07)	72.23 (69.48)	85.01 (80.56)	79.52 (77.04)	82.17 (78.73)	82.58 (84.66)	81.24 (78.11)	81.91 (81.25)
L(NNW,PNW),L(THW,HTW)	76.18 (71.25)	70.66 (68.13)	73.28 (69.63)	84.55 (80.47)	**80.19** (77.47)	82.31 (78.93)	81.05 (84.01)	83.25 (79.16)	82.14 (81.51)
M_{num} = 2	75.58 (71.80)	69.87 (68.28)	72.61 (70.00)	85.27 (80.39)	77.88 (76.18)	81.38 (78.20)	80.85 (84.49)	81.39 (79.90)	81.11 (82.13)
M_{num} = 6	76.26 (71.88)	70.28 (67.57)	73.14 (69.65)	84.54 (79.41)	79.09 (76.78)	81.72 (78.07)	79.83 (83.33)	83.36 (80.12)	81.55 (81.69)
M_{num} = 8	76.04 (72.13)	69.19 (67.46)	72.44 (69.71)	85.29 (80.92)	78.26 (76.78)	81.62 (78.79)	80.50 (83.59)	83.15 (80.34)	81.80 (81.92)

TABLE V
EXAMPLES OF DIFFERENT OVERLAPPED TYPES

Overlapped Type	Example Sentence	Entity Mention
Left Overlapped	Hair dryness, breakage and loss.	Hair breakage Hair loss
Right Overlapped	Hip, back and leg pain.	Hip pain back pain
No Overlapped	Brain fog and decreased cognitive skills.	Brain fog decreased cognitive skills
Multi. Overlapped	Cough with yellow or bloody sputum.	Cough with yellow sputum Cough with bloody sputum

TABLE VI
EXPERIMENTAL RESULTS OF RECOGNIZING THE ENTITIES WITH DIFFERENT OVERLAPPED TYPES. THE BOLD NUMBER DENOTES THE HIGHEST VALUE FOR EACH TYPE AND DATASET

Model		CADEC	ShARe13	SHAER14
Wang et al.(2021) [20]	**No**	7.69	48.26	40.32
	Left	42.86	64.07	66.15
	Right	**62.22**	13.04	45.80
	Multi.	0.00	29.63	0.00
Li et al.(2022) [21]	**No**	32.79	48.48	50.38
	Left	46.51	69.60	62.40
	Right	51.43	**58.89**	68.77
	Multi.	**17.98**	33.33	0.00
TOE(ours)	**No**	**38.87**	**50.21**	**53.29**
	Left	**48.48**	**69.71**	**66.32**
	Right	56.79	49.59	**72.31**
	Multi.	0.00	**36.36**	0.00

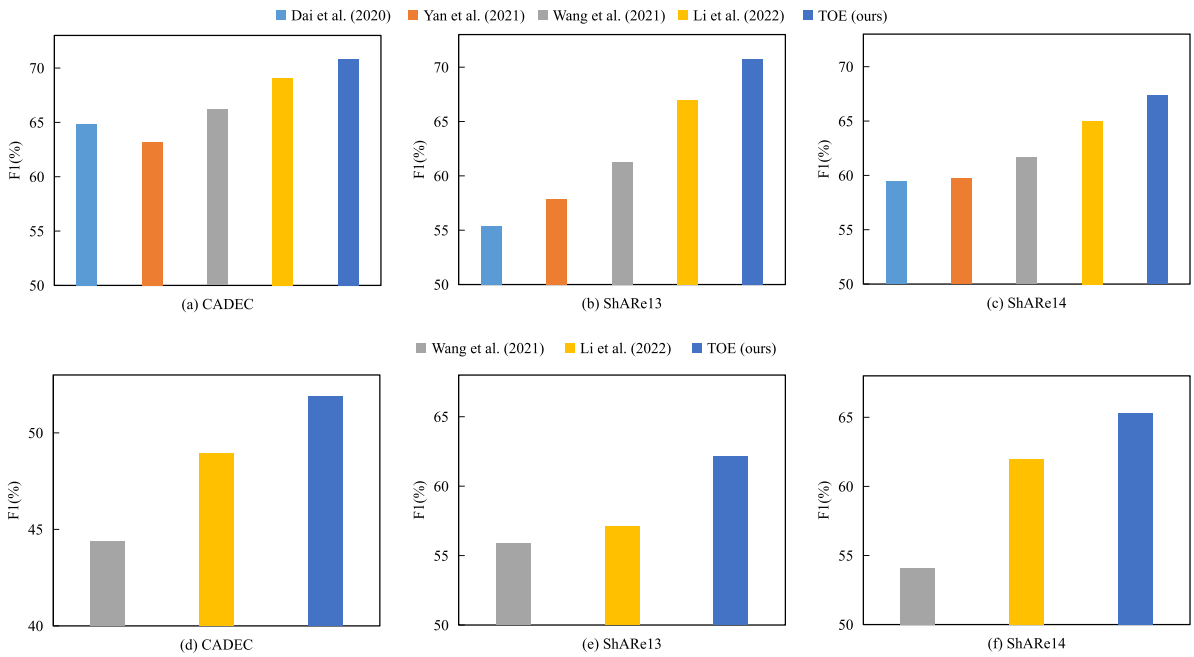

Fig. 5. Results of recognizing discontinuous entities, where (a)–(c) are the F1 values using the sentences containing at least one discontinuous entity, and (d)–(f) are the F1 values considering only discontinuous entities.

TABLE VII
ERROR ANALYSIS ON DIFFERENT ENTITY TYPES. HEAD-TAIL RELATIONSHIP CORRESPONDS TO THE THW AND HTW TAGS, AND THE NEIGHBOR WORDS RELATIONSHIP CORRESPONDS TO THE NNW AND PNW TAGS. ALL RESULTS COME FROM SENTENCES CONTAINING AT LEAST ONE CORRESPONDING ENTITY TYPE

Entity Type	Error Type		CADEC(%)	ShARe13(%)	ShARe14(%)
All	FP	Head-tail relationship correct, neighbor words relationship incorrect	4.05	0.50	1.01
		Head-tail relationship incorrect	40.28	40.33	52.29
	FN	Head-tail relationship correct, neighbor words relationship incorrect	2.43	0.50	0.91
		Head-tail relationship incorrect	53.24	58.67	45.79
	Total		100	100	100
Discontinuous Entity	FP	Head-tail relationship correct, neighbor words relationship incorrect	9.74	5.06	6.87
		Head-tail relationship incorrect	15.48	17.05	35.23
	FN	Head-tail relationship correct, neighbor words relationship incorrect	5.73	3.94	4.65
		Head-tail relationship incorrect	69.05	73.95	53.25
	Total		100	100	100
Flat Entity	FP	Head-tail relationship correct, neighbor words relationship incorrect	0.00	0.00	0.00
		Head-tail relationship incorrect	50.14	39.96	58.92
	FN	Head-tail relationship correct, neighbor words relationship incorrect	0.00	0.00	0.00
		Head-tail relationship incorrect	49.86	60.04	41.08
	Total		100	100	100
Overlapped Entity	FP	Head-tail relationship correct, neighbor words relationship incorrect	12.53	4.70	3.47
		Head-tail relationship incorrect	28.46	17.10	34.74
	FN	Head-tail relationship correct, neighbor words relationship incorrect	1.76	4.27	2.65
		Head-tail relationship incorrect	57.25	73.93	59.14
	Total		100	100	100

ones of other overlapping types. For instance, the number of multiple overlapped entities in the training set of ShARe14 is 20, leading to under-fitting for model training.

E. Error Analysis

In order to understand the disadvantages and advantages of our model, we perform an error analysis and show the results in this section. Errors can be divided into false-positive (FP) and false-negative (FN). Furthermore, FP errors and FN errors can be further divided into two types: "head-tail relationship incorrect" and "head-tail relationship correct but neighbor words relationship incorrect". As shown in Table VII, we analyze the errors of our model on three discontinuous datasets from four directions: "all entities," "flat entities," "overlapped entities" and "discontinuous entities". We can see that regardless of the entity type, the FN and FP errors for "head-tail relationship incorrect" account for high proportions. Especially in recognizing flat entities, 100% errors come from incorrect head-tail relationship recognition. This may be because the head-tail relationship is

TABLE VIII
EFFICIENCY ANALYSIS. SENT/S IS THE NUMBER OF SENTENCES THAT CAN BE PROCESSED PER SECOND

Model	Training (sent/s)	Inference (sent/s)
Dai et al.(2020) [54]	24.7	66.5
Yan et al.(2021) [17]	63.6	19.2
Wang et al.(2021) [20]	39.3	109.7
Li et al.(2022) [21]	116.1	365.7
TOE(ours)	78.5	195.1

a more difficult relationship to recognize compared with the neighbor relationship, since entity boundary is hard to identify, which is a well-known issue in previous work [18], [60]. In addition, we can also observe that in most cases, the FP numbers of "head-tail relationship incorrect" are smaller than the FN numbers, which shows that the recognition coverage of gold-standard entities is still a challenge for our model.

F. Efficiency Analysis

Table VIII shows the training speed and prediction speed of our model. Compared with the model of Li et al. (2022) [21], our model has a slower training speed and prediction speed. This is mainly because we inject the tag embedding into our model and the computation in the TREM can be performed iteratively, resulting in more parameters and calculations. Although the performance improvement of our model leads to a decrease in efficiency compared with the model proposed by Li et al. (2022) [21], our model still has certain advantages in the training and prediction speeds compared with other baseline models [17], [20], [54], which demonstrates that the method based on grid tagging is more efficient than other kinds of methods.

VI. CONCLUSION

This paper extends a SOTA grid-tagging model for discontinuous named entity recognition with tag-oriented enhancement. Our enhanced model has two strengths: (1) It not only pays attention to the relationships between words, but also the relationships between words and tags. (2) It leverages a more fined-grain tagging system to strengthen the prediction of the relationships between words and tags. The experimental results show that the performance of our model on all three benchmark datasets are the best, and the ablation experiments demonstrate the effectiveness of the two enhancements that we propose in the model. Further experimental analyses show that our proposed model can better identify discontinuous entities. Although the enhancements bring a certain efficiency loss, our model is still faster than most baselines in training and prediction. In the future, we will apply our model in more complex information extraction tasks such as nested entity relationship extraction and structured sentiment analysis.

REFERENCES

[1] M. A. Khalid, V. Jijkoun, and M. D. Rijke, "The impact of named entity normalization on information retrieval for question answering," in *Proc. Eur. Conf. Inf. Retrieval*, 2008, pp. 705–710.

[2] T. Zhao, Z. Yan, Y. Cao, and Z. Li, "Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction," in *Proc. 29th Int. Conf. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3948–3954.

[3] E. F. T. K. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proc. 7th Conf. Natural Lang. Learn. HLT-NAACL*, 2003, pp. 142–147.

[4] S. Pradhan et al., "Task 1: ShARe/CLEF eHealth evaluation lab 2013," in *Proc. Int. Conf. CLEF*, 2013, pp. 212–231.

[5] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, "The automatic content extraction (ACE) program-tasks, data, and evaluation," in *Proc. Lang. Resour. Eval. Conf.*, 2004, pp. 837–840.

[6] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus–A semantically annotated corpus for bio-textmining," *Proc. Bioinf.*, vol. 19, no. suppl_1, pp. i180–i182, 2003.

[7] S. Karimi, A. Metke-Jimenez, M. Kemp, and C. Wang, "CADEC: A corpus of adverse drug event annotations," *Proc. J. Biomed. Inform.*, vol. 55, pp. 73–81, 2015.

[8] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: https://doi.org/10.48550/arXiv.1508.01991

[9] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Proc. Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, 2016.

[10] M. Gridach, "Character-level neural network for biomedical named entity recognition," *Proc. J. Biomed. Informat.*, vol. 70, pp. 85–91, 2017.

[11] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1554–1564.

[12] T. Gui, R. Ma, Q. Zhang, L. Zhao, Y.-G. Jiang, and X. Huang, "CNN-based chinese NER with lexicon rethinking," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4982–4988.

[13] X. Mengge, B. Yu, Z. Zhang, T. Liu, Y. Zhang, and B. Wang, "Coarse-to-fine pre-training for named entity recognition," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 6345–6354.

[14] A. O. Muis and W. Lu, "Learning to recognize discontiguous entities," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 75–84.

[15] S. Pradhan et al., "Evaluating the state of the art in disorder recognition and normalization of the clinical narrative," *J. Amer. Med. Inform. Assoc.*, vol. 22, no. 1, pp. 143–154, 2015.

[16] B. Tang, J. Hu, X. Wang, and Q. Chen, "Recognizing continuous and discontinuous adverse drug reaction mentions from social media using LSTM-CRF," *Proc. Wireless Commun. Mobile Comput.*, vol. 2018, 2018, Art. no. 2379208. [Online]. Available: https://doi.org/10.1155/2018/2379208

[17] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various NER subtasks," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5808–5822.

[18] H. Fei, D. Ji, B. Li, Y. Liu, Y. Ren, and F. Li, "Rethinking boundaries: End-to-end recognition of discontinuous mentions with pointer networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 12785–12793.

[19] F. Li, Z. Lin, M. Zhang, and D. Ji, "A span-based model for joint overlapped and discontinuous named entity recognition," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4814–4828.

[20] Y. Wang, B. Yu, H. Zhu, T. Liu, N. Yu, and L. Sun, "Discontinuous named entity recognition as maximal clique discovery," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 764–774.

[21] J. Li et al., "Unified named entity recognition as word-word relation classification," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 10965–10973.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[23] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 260–270.

[24] D. L. Mowery et al., "Task 2: ShARe/CLEF eHealth evaluation lab 2014," in *Proc. Int. Conf. CLEF*, 2014, pp. 31–42.

[25] L. Liu et al., "Empower sequence labeling with task-aware neural language model," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5253–5260.

[26] Y. Lin, L. Liu, H. Ji, D. Yu, and J. Han, "Reliability-aware dynamic feature composition for name tagging," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 165–174.
[27] Y. Cao, Z. Hu, T.-S. Chua, Z. Liu, and H. Ji, "Low-resource name tagging learned with weakly labeled data," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 261–270.
[28] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 282–289.
[29] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 363–370.
[30] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2011, pp. 359–367.
[31] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 740–750.
[32] H. Zhou, Y. Zhang, S. Huang, and J. Chen, "A neural probabilistic structured-prediction model for transition-based dependency parsing," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 1213–1222.
[33] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 2227–2237.
[34] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1638–1649.
[35] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1064–1074.
[36] J. Yang, S. Liang, and Y. Zhang, "Design challenges and misconceptions in neural sequence labeling," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3879–3889.
[37] W. Lu and D. Roth, "Joint mention extraction and classification with mention hypergraphs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 857–867.
[38] A. Katiyar and C. Cardie, "Nested named entity recognition revisited," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 861–871.
[39] B. Wang and W. Lu, "Neural segmental hypergraphs for overlapping mention recognition," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 204–214.
[40] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, "Multilingual language processing from bytes," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 1296–1306.
[41] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.
[42] M. Xu, H. Jiang, and S. Watcharawittayakul, "A local detection approach for named entity recognition and mention detection," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1237–1247.
[43] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "LUKE: Deep contextualized entity representations with entity-aware self-attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 6442–6454.
[44] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, "A unified MRC framework for named entity recognition," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5849–5859.
[45] J. Li, K. Xu, F. Li, H. Fei, Y. Ren, and D. Ji, "MRN: A locally and globally mention-based reasoning network for document-level relation extraction," in *Proc. Findings Assoc. Comput. Linguistics: ACL-IJCNLP*, 2021, pp. 1359–1370.
[46] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5784–5789.
[47] R. Liu, J. Wei, C. Jia, and S. Vosoughi, "Modulating language models with emotions," in *Proc. Findings Assoc. Comput. Linguistics: ACL-IJCNLP*, 2021, pp. 4332–4339.
[48] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*. [Online]. Available: http://arxiv.org/abs/1606.08415
[49] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[51] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," in *Proc. 5th Int. Conf. Learn. Representations*, 2016, pp. 24–46.
[52] J. Su, "Extending "softmax cross entropy" to multi-label classification," 2020. [Online]. Available: https://kexue.fm/archives/7359
[53] Y. Sun et al., "Circle loss: A unified perspective of pair similarity optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6398–6407.
[54] X. Dai, S. Karimi, B. Hachey, and C. Paris, "An effective transition-based model for discontinuous NER," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5860–5870.
[55] B. Wang and W. Lu, "Combining spans into entities: A neural two-stage approach for recognizing discontiguous entities," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6216–6224.
[56] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6470–6476.
[57] J. Lee et al., "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Proc. Bioinf.*, vol. 36, no. 4, pp. 1234–1240, 2020.
[58] E. Alsentzer et al., "Publicly available clinical BERT embeddings," in *Proc. 2nd Clin. Natural Lang. Process. Workshop*, 2019, pp. 72–78.
[59] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–8.
[60] M. Cho, J. Ha, C. Park, and S. Park, "Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition," *Proc. J. Biomed. Inform.*, vol. 103, 2020, Art. no. 103381.