Journal Homepage: -www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI:10.21474/IJAR01/18405
DOI URL: http://dx.doi.org/10.21474/IJAR01/18405

RESEARCH ARTICLE

A BIDIRECTIONAL ENCODER-DECODER MODEL WITH ATTENTIONMECHANISM FOR NESTED NAMED ENTITY RECOGNITION

Samassi Adama[1,2], Brou Konan Marcellin[1,2], Kouamé Appoh[1,2] and Touré Kidjegbo Augustin[1]

1. Ecole Doctorale Polytechnique, Institut National Polytechnique (INP-HB), Yamoussoukro, Côte d'Ivoire.
2. Laboratoire de Recherche en Informatique et Télécommunication (LARIT).

Manuscript Info

Manuscript History
Received: 10 January 2024
Final Accepted: 14 February 2024
Published: March 2024

Key words:-
Attention Mechanism, Fine-Tuning, Named Entity Recognition, Sequence Labeling

Abstract

Named entity recognition is a fundamental task for several natural language processing applications. It consists in identifying mentions of named entities in a text, then classifying them according to predefined entity types. Most labeling methods for this task use a label to recognize flat named entities because they belong to a single entity type. Therefore, they cannot recognize named entities that belong to multiple entity types.In this work, we concatenated all the labels of a word of a named entity into a joint in order to recognize flat or nested named entities. Then, we proposed a bidirectional encoder-decoder model with attention mechanism that uses this joint label to fine-tune a pre-trained language model for named entity recognition.We experimented our method on GENIA (a nested named entity dataset) and on two flat named entity datasets: CoNLL-2003 and i2b2 2010. Using the BioBERT model, our method achieved an F1 score of 78.85% on the GENIA dataset, 93.22% and 87.51% on CoNLL-2003 and i2b2 2010 respectively. These results show that our method can effectively recognize flat named entities as well as nested named entities.

Copy Right, IJAR, 2024,. All rights reserved.

Introduction:-

The task of Named Entity Recognition (NER) consists in identifying mentions of named entities in a text, and classifying them according to predefined entity types (or categories). It is the preprocessing step of several natural language processing (NLP) applications such as information retrieval [1] and machine translation [17]. According to the authors [2] and [3], a named entity (NE) is an expression that designates objects or concepts. Entity types can be person, organization, location, etc., in the general field and dna, proteins, genes, etc., in the biomedical domain.

Previous works for NER [4, 24, 28]focus on flat named entities. They train a sequence labeling model on a flat NER dataset $D = \{(X_i, Y_i)\}_{i=1}^{m}$containing samples in the form of pairs (X_i, Y_i). The sentence $X = \{x_1, x_2, \dots, x_n\}$ is the input of the model and $y = \{y_1, y_2, \dots, y_n\}$ is the corresponding gold label sequence. Each sentence is annotated according to a labeling scheme such as BIO (Begin, Inside, Other) [7], which assigns each word of an entity a label consisting of a position indicator (B or I) and an entity type such as person (PER) or location (LOC) (Figure 1).

Corresponding Author:- Samassi Adama
Address:- PhD Student in Computer Science.Ecole Doctorale Polytechnique, Institut National Polytechnique (INP-HB).

```
Nader   B-PER
Jokhadar    I-PER
had O
given   O
Syria   B-LOC
the O
lead    O
with    O
a   O
well-struck O
header  O
in  O
the O
seventh O
minute  O
.   O
```

Figure 1:-Example sentence (first column) and its corresponding labels represented in BIO [7] format.

However, there arenested named entities, wich are entites nested within each other (Figures 2 and 3). For example, **interleukin-2** and **interleukin-2 receptor alpha gene** are nested entities [32]. "interleukin-2" is a protein and it is also a DNAbecause it is part of "Mouse interleukin-2 receptor alpha gene", which is a DNA.

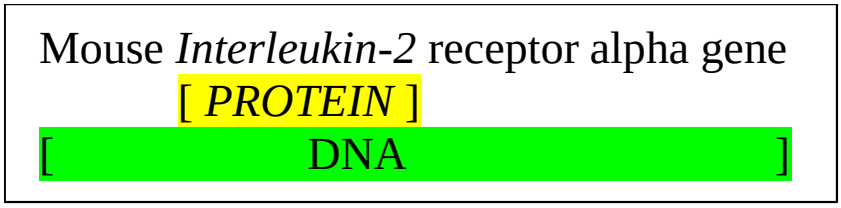

Figure 2:-Interleukin-2 and **Mouseinterleukin-2 receptor alpha gene**, are nested entities. "interleukin-2" is a protein and it is also part of "Mouse interleukin-2 receptor alpha gene", which is a DNA [32].

There are several methods for recognizing nested named entities. Some of them use hypergraphs [11,47], parse trees [42], dependency graphs [43] or constituent analysis [44] to model the nested structures, and then employ neural network models to recognize nested named entities. Other methods [12, 46] proposed to merge all the labels of a word of a nested entity into a joint label [12], which can be employed by sequence labeling models to recognize nested named enities. Figure 3 shows examples of joint labels created by the joint labeling technique [46]. In (Agrawal et al. 2022) [12], the authorsfine-tuned pre-trained BERT [13] models for the nested named recognition task using the joint label and the BiLSTM-CRF [4] model.

Phrase	Level 1	Level 2	Joint Label
Interleukin-2	B- protein	B-DNA	B- protein + B-DNA
receptor	O	I-DNA	O+ I-DNA
alpha	O	I-DNA	O + I-DNA
gene	O	I-DNA	O + I-DNA
expression	O	O	O + O

Figure 3:-Examples of joint labels obtained by concatenating the BIO labels at all levels (L1 and L2) of a word. "+" is used as a delimiter. B-protein+B-DNA and O+I-DNA are joint labels.

In this paper, we focus our work on the joint labeling method proposed by (Agrawal et al. 2022) [12]. However, we employed a new sequence labeling model where the decoders are based on Recurrent Neural Networks (RNNs)to fine-tune BioBERT [14] model to support flat and nested named entity recognition. Our model is built according to the architecture of the Seq2Biseq model proposed by (Dinarelli et al.2019) [15], wich employs two GRU [36] decodersto model the dependencies between the output labels. However, theses decoders focus only on the hidden state of the encoder at a time step, which is not sufficient for label prediction. Therefore, we introduced into each decoder of Seq2Biseq [15] a context vector, computed using the attention mechanism proposed by [6]. Thecontext vector allows each decoder to focus not only on the hidden state of the encoder at a time step, but also on other relevant hidden states of the encoder for the prediction of a label at a time step. We called this new model AttnSeq2biseq (Attention-based Seq2biseq).

The contributions of this paper are as follows:
- We proposed abidirectional encoder-decoder model with attention mechanism for sequence labeling task. Our model is based on the Seq2Biseq model [15], which employs a birectional GRU encoder and two GRU decoders. We introduced into the two decoders of Seq2Biseq [15] attention mechanisms which let them to focus only on the relevant words words in the input sentence.
- We fine-tuned the BioBERT v1.0model[14] on English NER datasets for named entity recognition using our model. For this purpose, we used the joint labeling scheme on GENIA,and the BIO labeling scheme on CoNLL-2003 and i2b2 2010 datasets. Our method achieves F1-scores of 78.85%, 93.22% and 87.51% on GENIA, CoNLL-2003 and i2b2 2010 datasets respectively.

The rest of our study is organized as follows. Section **Related work:-** reviews related works. Our methodology for NER is presented in Section **Methodology:-**. The experiments carried out are described in section **Experiments:-**. The results of the experiments are discussed in section **Results and Discussion:-**. Finally, the conclusion is made in section **Conclusion:-**.

Related Work:-

Flat Named Entity Recognition:

We focus on the methods that employneural network models ([20, 21], [24-28])for flat named entity Recognition on CoNLL-2003 and i2b2 2010 datasets.

CoNLL-2003 dataset. (Huang et al., 2015) [4] proposed the BiLSTM-CRF model for flat NER, and obtained an F1 score of 90.10%. (Ma and Hovy, 2016) [25] proposed the BiLSTM-CNN-CRF model, where the encoder uses a convolutional neural network (CNN) [22] to extract the character-level representation of each word from the input sequence. The final vector representation of each word, obtained by concatenating its distributional representation and its character-level representation, is processed by a BiLSTM network to construct its context-sensitive representation. This model obtains an F1 score of 91.21%. (Panchendrarajan and Amaresan, 2018) [24] proposed the

BiLSTM-BiCRF model which uses two CRF-based decoders and obtained an F1 score of 90.84%. (Che et al., 2020) [26] proposed the TCN-CRF model, where the encoder is a temporal convolutional network (TCN) integrating convolution kernels to extract word features. The F1 score obtained with the model is 91.42%.

I2b22010 Dataset. (Chalapathy et al., 2016) [27] proposed a method for biomedical NER (BioNER) that uses a BiLSTM-CRF model initialized with GloVe [18] or Word2Vec [19] word embeddings. The results obtained are respectively 81.30% and 83.81% in terms of F1 score. (Zhu et al., 2018) [28] proposed another BioNER method in which a BiLSTM-CRF model is initialized with ELMo embeddings [29]. The result obtained is 86.84% in terms of F1 score. (Si et al., 2019) [30] used BERT embeddings [13], trained on the MIMIC-III dataset, in a BiLSTM-CRF model and obtained an F1 score of 90.25 on the i2b2 2010 dataset.

Nested Named Entity Recognition:
Nested NER methods can be divided nested NER methods into three main approaches: the sequence labeling approaches, the structure-based approaches and the span-based approaches.

Sequence Labeling approaches assign a label to every word ([4], [8-9], [12], [24-26]). Span-based methods enumerate all possible spans and then combine them into entities [48-50]. Structured-based Methods use hypergraphs to represent nested entity structure [11, 47]. We focus our work on sequence labeling approaches, particularly the joint labeling method proposed by (Agrawal et al. 2022) [12].

Sequence Labeling based approaches
Sequence Labeling approaches assign a label to every word ([4], [8-9], [12], [24-26]), then they employ a sequence labeling model to recognize named entities. For example, the authors (Ju et al., 2018) [32] stacked BiLSTM-CRF models to recognize nested named entities. The result obtained in terms of F1 score is 74.70% on the GENIA dataset. However, this method suffers from error propagation between the BiLSTM-CRF models.

(Strakova et al., 2019) [8] proposed two methods to model the structure of nested entities. In the first method, they employed a BILOU encoding scheme [58] to concatenate the labels of each nested entity into a single multi-label. Then, they used a BiLSTM-CRF model [31] to predict this label. The second method considers nested NER as a sequence-to-sequence problem, where a sequence of labels is generated from a sequence of words by a seq2seq model [39]. The seq2seq model used by the authors is a BiLSTM-LSTM model, where the encoder and the decoder are based on LSTM. In this model, the decoder uses a hard attention mechanism [34] on each word when predicting its label. However, an LSTM decoder generates the labels sequentially and predicts the next label based on the previously predicted labels. Therefore, it suffers from the exposure bias [57] problem. (Shibuya et al., 2020) [9] proposed a method that uses a BiLSTM encoder to compute the contextual representation of each word in the input sentence. Next, they employ a CRF decoder for each entity type to recognize the nested entities, from the outermost entities to the inner ones. The method achieves an F1 score of 77.36% on the GENIA corpus.

(Wang et al., 2020) [10] proposed Pyramid, a recursive model which embeds entity mentions in a text into flat NER models (layers), stacking them from the bottom to the top. The method achieves an F1 score of 79.31% on the GENIA corpus with the pre-trained contextual embeddings BERT and Flair.

Joint Labeling Method.
There has been few research dedicated to identifying the nested entities using amult-label. (Agrawal et al. 2022) [12] proposed a joint labeling method, which employs the joint labeling technique [46] and a BiLSTM-CRF [4] model to fine-tune BERT models for nested named entity recognition. However, the joint labeling technique increases the number of labels in the training data [46], which may lead to a degradation in the performance of the BiLSTM-CRF model.

Other Approaches
Structure-Based Approaches employ hypergraphs [11], [47], parse trees [42], dependency graphs [43] or constituent analysis [44] to model the structure of nested entities. For example, (Katiyar and Cardie,2018) [11] employed hypergraphs to recognize the mentions nested entities. In this method, the authors select the mentions of nested entities using a hyper-parameter, which must be adjusted to obtain a better F1 score.

Span-based Approaches consider nested entities as spans in a sentence [37], [45], [48], [49], [51], [52]. For this, they use two steps: the detection of spans from a sentence and their classification into entity types.

Model Architecture:-

The architecture of our AttnSeq2Biseq model is inspired from the work of (Dinarelli et al.2019) [15] and (Bahdanau et al.2014) [16]. Our model consists of three main layers: The Embedding Layer, the Encoder Layer and the Decoder Layer (Figure 4).

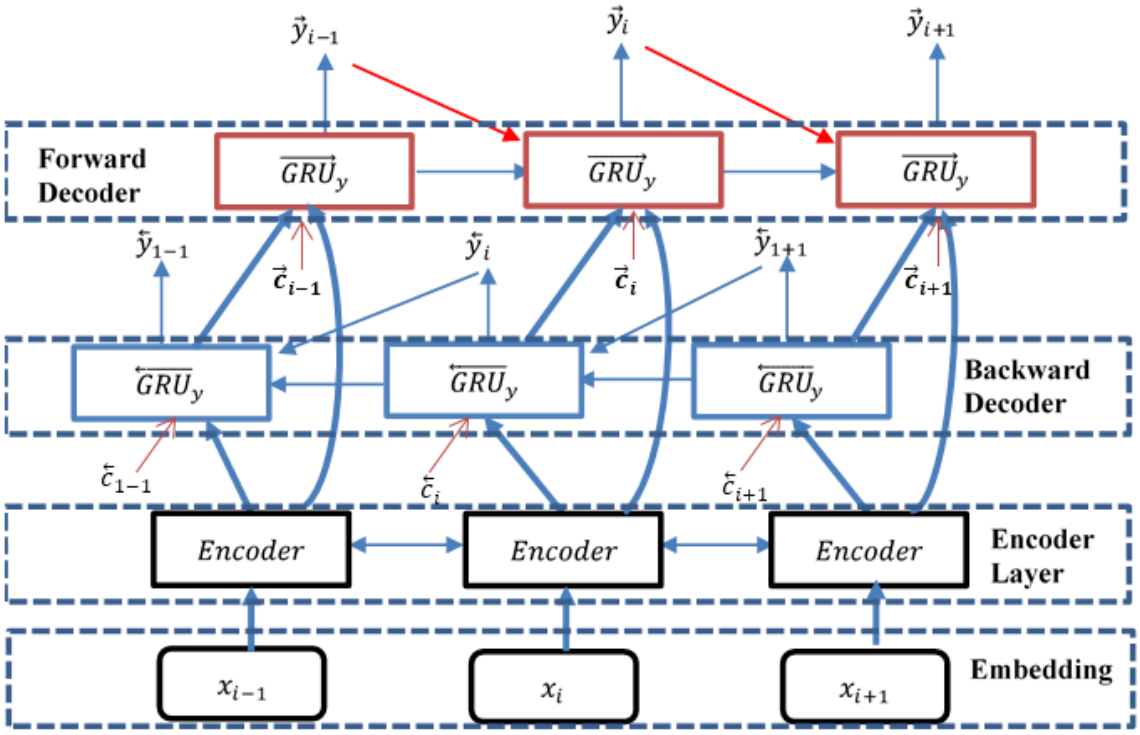

Figure 4:-The architecture of our AttnSeq2Biseq. $X = (x_{i-1}, x_i, x_{i+1})$ is a sentence, and $y = (\vec{y}_{i-1}, \vec{y}_i, \vec{y}_{i+1})$ the corresponding label sequence generated by the model. $\overleftarrow{\mathbf{c}}_\mathbf{i}$ and $\vec{\mathbf{c}}_\mathbf{i}$ are context vectors computed by attention mechanism [16].

Embedding Layer:

We used the embedding layer to obtain the vector representations (feature vectors) of the words in the input sentence.Let $x = (x_1, \ldots, x_n)$ be the sentence with n words. $x_i \in \mathbb{R}^{|v|}$ is the i-th word, v is the size of the vocabulary. We obtained the feature vector $\mathbf{x_i}$ of a word x_i by concatenating its embedding $\mathbf{w_i}$ and its character-based representation $\mathbf{h_i^c} \in \mathbb{R}^{d_c}$ as in [15], where [;] denotes concatenation operation:

$$\mathbf{x_i} = [\mathbf{w_i}; \mathbf{h_i^c}] \quad (1)$$

We used the pretrained models Word2vec [19], Flair [41] and BioBERT [14] to obtain the word embedding $\mathbf{w_i}$. We used the Word2vec embeddings of dimension d = 200, induced from 23M documents of PubMed and PubMed Central (PMC) (Moen et al.2013) [53].

We used the BioBERT v1.0 [14] model, which contains 12 layers that can be used to generate the embeddings of the words of the input sentence. But in this paper, we used the last four layers to get the word embedding $\mathbf{w_i}$. The reason is that the last layers produce more optimal features than the bottom layer. The BioBERT model employs a subword tokenizer to break each word of the sentence into one or more sub-words, in order to handle unknown words and morphological variation. Therefore, several strategies can be used to produce an embedding of a word in BioBERT.

In this work, we used the embedding of the first sub-word (first subword pooling strategy) [13]to represent the entire word. However, because the NER datasets are labelled at word level, after the tokenization process, we assign the label of a word to its first sub-word, and assign padding labels to its other sub-words.

Encoder Layer:

The Encoder is a bidirectional GRU consisting of a forward GRU unit ($\overrightarrow{\text{GRU}}$)and a backward GRU unit ($\overleftarrow{\text{GRU}}$). The forward$\overrightarrow{\text{GRU}}$ processes the sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$in the forward direction to generate the sequence of left contextual representations $\overrightarrow{\mathbf{h}} = (\overrightarrow{\mathbf{h}}_1, \overrightarrow{\mathbf{h}}_2, \dots, \overrightarrow{\mathbf{h}}_n)$, while the backward$\overleftarrow{\text{GRU}}$ processes the same sequence in the opposite direction and generates the sequence of right contextual representations $\overleftarrow{\mathbf{h}} = (\overleftarrow{\mathbf{h}}_1, \overleftarrow{\mathbf{h}}_2, \dots, \overleftarrow{\mathbf{h}}_n)$:

$$\overrightarrow{\mathbf{h}}_i = \overrightarrow{\text{GRU}}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{x}_i), i = 1, \dots, n; \quad \overleftarrow{\mathbf{h}}_i = \overleftarrow{\text{GRU}}(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{x}_i), i = n, \dots, 1 \tag{2}$$

Note that the vectors $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ are the respective hidden states of the GRU units$\overrightarrow{\text{GRU}}$ and $\overleftarrow{\text{GRU}}$ at time step i. The contextual word representation $\mathbf{h}_i \in \mathbb{R}^{2d}$, which captures the global context (left and right contexts) of each word word $x_i$is obtained by the concatenation of the hidden states $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i], i = 1, \dots, n \tag{3}$$

where [;] is the concatenation operator.

The Decoder Models

The sequence of contextual word representations $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n)$ obtained from the encoder is fed into the decoder layer which generate the most probable output sequence $y = y_1, y_2, \dots, y_n$. Here, we assume that there is a hidden dependency between **H** and the label sequence y that can be captured by the conditional probability:

$$p(y|\mathbf{H}) = p(y_1, y_2, \dots, y_n | \mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n) \tag{4}$$

Since Equation 4 can be decomposed from left to right or from right to left, we propose to decode the sequence **H** using two decoders, the Backward Decoder and the Forward Decoder, implemented by GRU hidden units as in [15] and attention mechanism [16].

Attention Mechanisms.

We integrated into the Backward Decoder (resp. Forward Decoder) a context vector $\overleftarrow{\mathbf{c}}_i$ (resp. $\overrightarrow{\mathbf{c}}_i$), which lets it to choose which part of the encoder's hidden states $\{\mathbf{h}_i\}$to pay attention to. A context vector is computed using an attention mechanism (Bahdanau et al,2014) [16], and is the weighted sum of the encoder's hidden states $\{\mathbf{h}_i\}$.

The Backward Decoder Layer

The Backward Decoder($\overleftarrow{\text{GRU}_y}$)isa GRU hidden unit integrating a context vector $\overleftarrow{\mathbf{c}}_i$ produced by attention mechanism[16]. It decodes the sequence of hiden states **H**of the encoder from right to left and generates the label sequence $\overleftarrow{y} = \overleftarrow{y}_i, \dots, \overleftarrow{y}_n$ that maximizes the conditional probability $p(\overleftarrow{y}|\mathbf{H})$:

$$p(\overleftarrow{y}|\mathbf{H}) = \prod_{i=n}^{1} p(\overleftarrow{y}_i | \overleftarrow{y}_{i+1}, \dots, \overleftarrow{y}_n, \mathbf{H}) \tag{5}$$

$p(\overleftarrow{y}_i | \overleftarrow{y}_{i+1}, \dots, \overleftarrow{y}_n, \mathbf{H})$ is the probability of each possible label occuring at the decoding time step i, conditioned upon the previously generated labels $\overleftarrow{y}_{i+1}, \dots, \overleftarrow{y}_n$.

$$p(\overleftarrow{y}_i | \overleftarrow{y}_{i+1}, \dots, \overleftarrow{y}_n, \mathbf{H}) = \overleftarrow{g}(\overleftarrow{\mathbf{s}}_i, \mathbf{h}_i, \overleftarrow{\mathbf{c}}_i) \tag{6}$$

$\overleftarrow{g}$ is the output layer, composed of a linear layer followed by a log-softmax function:

$$p(\overleftarrow{y}_i | \overleftarrow{y}_{i+1}, \dots, \overleftarrow{y}_n, \mathbf{H}) = \text{log-softmax}\left(\overleftarrow{W}_o [\mathbf{h}_i; \overleftarrow{\mathbf{s}}_i] + \overleftarrow{\mathbf{V}}_c \overleftarrow{\mathbf{c}}_i + \overleftarrow{\mathbf{b}}_0\right) \tag{7}$$

$\overleftarrow{\mathbf{s}}_i$ is the hidden state of the decoder $\overleftarrow{GRU}_y$ at time step i, calculated by a GRU unit:

$$\overleftarrow{\mathbf{s}}_i = \overleftarrow{GRU}_y(\overleftarrow{\mathbf{s}}_{i+1}, [\overleftarrow{\mathbf{y}}_{i+1}; \mathbf{h}_i; \overleftarrow{\mathbf{c}}_i]) \quad (8)$$

The context vector $\overleftarrow{\mathbf{c}}_i$ is calculated for a time step i as the weighted sum of the hidden states $\{\mathbf{h}_j\}$ of the encoder:

$$\overleftarrow{\mathbf{c}}_i = \sum_{j=1}^{n} \overleftarrow{\alpha}_{ij} \mathbf{h}_j, \quad (9)$$

where $\sum_j \overleftarrow{\alpha}_{ij} = 1$ and $\overleftarrow{\alpha}_{ij} > 0, \forall i, j$.

The weight $\overleftarrow{\alpha}_{ij}$ of each hidden state $\mathbf{h}_j$ of the encoder is computed for time step i as [16]:

$$\overleftarrow{\alpha}_{ij} = \frac{\exp(\overleftarrow{e}_{ij})}{\sum_{k=1}^{n} \exp(\overleftarrow{e}_{ik})} \quad (10)$$

$\overleftarrow{e}_{ij}$ is an alignment score, computed by [16]:

$$\overleftarrow{e}_{ij} = \overleftarrow{v}_a^T \tanh(\overleftarrow{\mathbf{W}}_s \overleftarrow{\mathbf{s}}_{i+1} + \overleftarrow{\mathbf{U}}_h \mathbf{h}_j) \quad (11)$$

$\overleftarrow{v}_a^T \in \mathbb{R}^d$, $\overleftarrow{\mathbf{W}}_s \in \mathbb{R}^{d \times d}$, and $\overleftarrow{\mathbf{U}}_h \in \mathbb{R}^{d \times 2d}$ are weight matrices.

Forward Decoder Layer

The The Forward Decoder ($\overrightarrow{GRU}_y$) isa GRU hidden unit integrating a context vector $\vec{\mathbf{c}}_i$. It processes **H** from left to right to generate the label sequence $\vec{y} = \vec{y}_1, \ldots, \vec{y}_n$ that maximizes conditional probability:

$$P(\vec{y}_i | \mathbf{H}) = \prod_{i=1}^{n} p(\vec{y}_i | \vec{y}_1, \ldots, \vec{y}_{i-1}, \mathbf{H}) \quad (12)$$

$p(\vec{y}_i | \vec{y}_1, \ldots, \vec{y}_{i-1}, \mathbf{H})$ is the probability that each possible label occurs at decoding time step i, conditioned by the previously generated labels $\vec{y}_1, \ldots, \vec{y}_{i-1}$:

$$p(\vec{y}_i | \vec{y}_1, \ldots, \vec{y}_{i-1}, \mathbf{H}) = \vec{g}(\vec{\mathbf{s}}_i, \mathbf{h}_i, \overleftarrow{\mathbf{s}}_i, \vec{\mathbf{c}}_i, \overleftarrow{\mathbf{c}}_i) \in \mathbb{R}^{|L|} \quad (13)$$

$\vec{g}$ is the output layer, composed of a linear layer followed by a log-softmax function:

$$\vec{p}(\vec{y}_i | \vec{y}_1, \ldots, \vec{y}_{i-1}\}, \mathbf{H}) = \text{log-softmax}(\overrightarrow{W}_o [\vec{\mathbf{s}}_i; \mathbf{h}_i; \overleftarrow{\mathbf{s}}_i] + \vec{\mathbf{V}}_c [\vec{\mathbf{c}}_i; \overleftarrow{\mathbf{c}}_i] + \vec{\mathbf{b}}_o) \quad (14)$$

$\overleftarrow{\mathbf{s}}_i$ is the hidden state of the decoder $\overrightarrow{GRU}_y$ at time step i, calculated by a GRU unit:

$$\vec{\mathbf{s}}_i = \overrightarrow{GRU}_y(\vec{\mathbf{s}}_{i-1}, [\vec{\mathbf{y}}_{i-1}; \mathbf{h}_i; \vec{\mathbf{c}}_i]) \quad (15)$$

The context vector $\vec{\mathbf{c}}_i$ is calculated for a time step i as the weighted sum of the hidden states $\{\mathbf{h}_j\}$:

$$\vec{\mathbf{c}}_i = \sum_{i=1}^{n} \vec{\alpha}_{ij} \mathbf{h}_j \quad (16)$$

The weight $\vec{\alpha}_{ij}$ of each hidden state $\mathbf{h}_j$ of the encoder is computed at time step i as:

$$\vec{\alpha}_{ij} = \frac{\exp(\vec{e}_{ij})}{\sum_{k=1}^{n} \exp(\vec{e}_{ik})} \quad (17)$$

$\overleftarrow{e}_{ij}$ is an alignment score, computed by [16]:

$$\vec{e}_{ij} = \vec{v}_b^T \tanh\left(\overrightarrow{W}_s \vec{s}_i + \overrightarrow{W}_h \mathbf{h}_j\right) \quad (18)$$

$\overleftarrow{v}_a^T \in \mathbb{R}^d$, $\overleftarrow{\mathbf{W}}_s \in \mathbb{R}^{d \times d}$, and $\overleftarrow{\mathbf{U}}_h \in \mathbb{R}^{d \times 2d}$ are weight matrices.

Note that in Equation (13) both left and right label context representations $\vec{\boldsymbol{s}}_i$ and $\overleftarrow{\boldsymbol{s}}_i$ are used when predicting the label $\vec{y}_i$.Therefore, the Forward Decoder acts a bidirectional decoder.

Experiments:-

Datasets:

We conducted our experiments on three datasets: CoNLL-2003[40], i2B2 2010 [35] and GENIA [38]. CoNLL-2003[40] and i2B2 2010 are flat NER datasets annotatedwithBIO labeling scheme. GENIA is a nested dataset that we annotated using the joint labeling technique [46]. We split each dataset into three sets: the training set, the development and test sets. The training set is used to train the model, the development set to tune the hyperparameters of the model. The test sset is used to evaluate the best model.

CoNLL-2003 Dataset.

The CoNLL-2003 dataset [40], obtained from Reuter's 1996 news material, contains four types of named entities: names of persons (PER), name of place (LOC), names of organization (ORG) and other entities (MISC). The dataset was divided into a training set, a development set, and a test set.

I2b2 2010 Dataset.

In this work, we used the data provided by the i2b2 2010 challenge [35] to train a clinical concept extraction system. Three clinical concepts are annotated in this corpus: problems, tests and treatments.

Genia Dataset.

The GENIA dataset is a collection of biomedical literature compiled and annotated with different levels of linguistic and semantic information. The original version contains 36 feature classes. However, we used the simplified version provided by [51], which contains four levels of nesting, and the entities are grouped in it into only five major classes: protein, DNA, RNA, cell line, cell type. We used 81% of the dataset for the training set, 9% for the development and 10% test set.

Table 1:-Statistcsfor theCoNLL-2003 dataset.

Dataset	Types		Train	Dev	Test	Overall
CoNLL-2003	4	Sentence Percentage	14987 68,01%	3466 15,73%	3584 16,26%	22037 100%
i2b2 2010	3	Sentence Percentage	15023 81%	1669 9%	1854 10%	18546 100%
GENIA	5	Sentence Percentage				

In the GENIA dataset, the maximum number of nesting levels is four (4) [12, 50]. So, four columns are used for each word when labeling. Table 2 shows example of sentence annotated using the the joint labeling technique [46].

Evaluation:

In this paper, we used the F-score (F_1) to evaluate the performance of our method for named entity recognition. It is the harmonic mean of Precision and Recall, thus making it possible to establish a balance between these two measures. It is calculated using equation (21). TP (True Positive) is the number of named entities (NEs) recognized by the model.FN (False Negative) is the number of unidentified, whileFP (False Positive)is the number of NEs that the model has misidentified.

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP+FN} \quad (20)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (21)$$

Training and Fine-tuning:
We added our AttnSeq2Biseq model on top of the BioBERT models to obtain the BioBERT-AttnSeq2Biseq model. Then we trained this modelon a training dataset. During training, we fine-tunedthree hyperparameters (Table 2) on the development dataset(the number of epochs, thebatch size and the learning rate) until the F1-score does not improve significantly on this dataset. Specifically, we performed several experiments to tune the hyperparameters on the development dataset. Finally, we kept the best model, wich is evaluated on the test dataset. The parameters used to fine-tune our BERT+Attnseq2Biseq model are reported in Table 2, We used Adam optimizer and set the number of epochs to 5, the learning rate to 5e-5, the batch size to 32 and the dropout probability to 0.5. The minimum momentum is set 0.8 and the maximum momentum to 0.9.

Implementation
We used Pytorch 1.10, Python 3.7 libraries and the code of the Seq2Biseq[1]model (Dinarelli et al.2019) [15] to implement our AttnSeq2Biseq. The model is trained on a single GTX 3060Ti GPU.

Table 2:-Model parameters.

Parameters	**Value**
Train batch size	32
Dev batch size	32
Test batch size	32
Learning rate	5e-05
Epochs	10
Dropout	0.5
Dimension of character-level embeddings	30
Dimension of label embeddings	200
GRU Hidden Layer Dimension	200

Results and Discussion:-
This section discusses the performance of our method as well as that of existing methods for named entity recognition in the three datasets CoNLL-2003, i2b2 2010 and GENIA.

Comparison of Results on the I2b2 2010 dataset
We compare our results on I2b2 2010 dataset with three (3) other methods. Table 3 shows that our method outperforms all the other methods, with an F-score of 91.51%, on the i2b2 2010 dataset. The method proposed by (Si et al., 2019) [30] comes second, with an F_1-score of 93.22%.

Table 3:-Comparison of results with existing methods for the i2b2 2010 dataset.

I2b2 2010		
Reference	**Input Features**	**F_1**(%)
[27] (BiLSTM-CRF, Chalapathy et al.2016)	Word2vec[19] embeddings	81,30
[27] (BiLSTM-CRF, Chalapathy et al.2016)	Glove embeddings	83,81
[28] (BiLSTM-CRF, Zhu et al.2018)	ELMo (clinic) [29] embeddings	86,84
[30] (BERT+BiLSTM-softmax, Si et al.2019)	BERT [13]	90,25
(Our approach) AttnSeq2Biseq(Seq2Biseq [15] + additive attention)	BioBERT embeddings+ character embeddings	**91,51**

[1]http://www.marcodinarelli.it/software.php

Comparison of Results on the CoNLL-2003 dataset

We compare our results on CoNLL-2003 dataset with five (5) other methods. As shown in Table 4, our method outperforms all those methods, achieving an F_1-score of 93.22%.

Table 4:- Comparison of results with existing methods for the CoNLL2003 dataset.

CoNLL-2003		
Reference	**Input Features**	**F_1(%)**
[4] (BiLSTM-CRF, Huang et al.2015)	SENNA [21] embeddings	90.10
[24] (BiLSTM-BiCRF, Panchendrarajan et al.2018)	Pre-trained embeddings, POS embeddings	90.84
[25] (BiLSTM-CNN-CRF, Ma Hovy 2016)	Glove + character embeddings	91.21
[26] (TCN-CRF, Che et al.2020)	Glove + character embeddings	91.42
[8] (Seq2seq + hard attention, Straková et al.2019)	Word2vec [19] embeddings	90.77
[8] (Seq2seq + hard attention, Straková et al.2019)	BERT + Flair [41] embeddings	93.0
(Our approach) AttnSeq2Biseq (Seq2Biseq [15] + additive attention)	BioBERT+Flair embeddings+character embeddings	**93,22**

Comparison of Results on the CoNLL-2003 dataset

We compare our results on GENIA dataset with nine (9) existing methods. Table 5 shows that the method proposed by (Wang et al., 2020) [10] is the most efficient with an F1 score of 79.31%. Our method using the AttnSeq2Biseq model and the label modeling technique [46] comes in second place with an F1 score of 78.85%. In addition, our method performed better than the two methods (Agrawal et al.2022) [12] and (Liao et al.2022) [48].

Table 5:-Comparison of results with existing methods for the GENIA corpus.

Reference	Input Features	F_1(%)
[32] (Ju et al. 2018, Sequence labeling)	Pre-trained word embeddings [2]	74,70
[9] (Shibuya et al.2020, Sequence labeling)	Glove [18] embedding	77,36
[10] (Wang et al.2020, Sequence labeling)	BERT, Flair [41]	**79,31**
[11] (Katiyar et al.2018, Structure-based)	Word2vec [19] embeddings	73,80
[47] (Luo et al.2020, Structure-based)	Word embedding [32]	76,0
[49] (Cui et al.2023, Span-based)	BERT	78,30
[8] (Straková et al.2019, Sequence labeling)	Word2vec + character embeddings	76.23
[12] (Agrawal et al.2022, Sequence labeling)	BioBERT[14]	74,38
[48] (Liao et al.2022, Span-based)	Pre-trained word embeddings	72,70
[8] (Straková et al.2019, Sequence labeling)	Word2vec + character embeddings	76,4
[8 (Straková et al.2019, Sequence labeling)	BERT + Flair + character embeddings	78,31
(Our approach) Att-Seq2Biseq (Sequence labeling)	BioBERT + Flair + character embeddings	78,85

Discussion:-

We compare our results with other methods. Tables3 and 4 show that our method outperforms all those methods on the flat named entity recognition datatsets CoNLL-2003and i2b2 2010, and the joint labeling method proposed by (Agrawal et al.2022) [12] by +0.54 F_1-score. Specifically, our method outperforms the method [30] on the i2b2 2010 dataset by +1.26 F_1-score, and outperforms themethod [8] on the CoNLL-2003dataset by +0.22 F_1-score. The reasons are as follows.Our method uses a sequence labeling model where the decoders are based on decoder RNN, which can model long-range dependencies between the labels. However, on the i2b2 2010 dataset, the method [30] employs a BERT+BiLSTM-softmax. But a softmax decoder does not take into account dependencies between output labels.On the CoNLL-2003 and GENIA datasets,the method proposed by (Strakova et al., 2019) [8] is based on a BiLSTM-LSTM model (a seq2seq model), which contains a bidirectional LSTM encoder and a forward LSTM decoder. However, a forward LSTM decoder can only take into account the context of the previously predicted labels $(\boldsymbol{y_1}, \ldots, \boldsymbol{y_{i-1}})$ (Figure 5) when decoding the label y_i at current time step i.

[2] pre-trained embeddings trained on MEDLINE summaries according to [49]

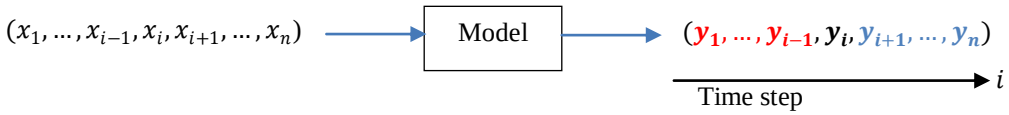

Figure 5:-A model receives an input sequence $(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n)$ andgenerates the corresponding output sequence of labels $(y_1, \dots, y_{i-1}, y_i, y_{i+1}, \dots, y_n)$

On GENIA dataset, the application of the joint labeling on a training datasetresults to a large number of possible labels. Therefore, a large number of labels will lead to a significant increase in the training time of a BiLSTM-CRF model, since the time complexity of a CRF algorithm is $O(NL^2)$, where N is the length of a sentence. The joint labeling increases the number of labels in the training data [46], which may lead to a degradation in the performance of the BiLSTM-CRF model. This is because the forward algorithm of a linear CRF [6] has a time complexity of $O(NL^2)$ [59], where N is the length of the sentence and L the number of possible labels in the dataset. However, according to (Shen et al. 2017) [56], a decoder based on recurrent neural networks (RNNs) such as LSTM or GRU is faster to train than a CRF decoder when the number of labels is large.This can explain the fact that our method is more efficient than that proposed by(Agrawal et al.2022) [12].

Our model uses decoders based on recurrent neural networks, therefore it suffers from the exposure-bias problem[57]. For future works, we want to integrate in our model the solution proposed by [57] or [58] solve this problem.

Conclusion:-

In this paper, we proposed an encoder-decoder model based on GRU and attention mechanismsfor sequence labeling tasks.Then, we employed this model to fine-tune the BioBERT v1.0 model fornamed entity recognitionusing the joint labeling scheme. Experimental results on the GENIA, i2b2 2010 and CoNLL2003 datasets showed that our method is effective for flat and nested named entity recognition.

References:-

[1] Boukhari, K., & Omri, M. N. (2017, July). Information retrieval approach based on indexing text documents: Application to biomedical domain. In 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 2213-2220). IEEE.
[2] Goyal, A., Gupta, V., & Kumar, M. (2018). Recent named entity recognition and classification techniques: a systematic review. Computer Science Review, 29, 21-43.
[3] Nouvel, D. (2012). Reconnaissance des entités nommées par exploration de règles d'annotation-Interpréter les marqueurs d'annotation comme instructions de structuration locale (Doctoral dissertation).
[4] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.
[5] Cho, Kyunghyun, van Merrienboer, Bart, Gulcehre, Caglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, 2014.
[6] Lafferty, J., Mccallum, A., Pereira, F. C. N. et Pereira, F. (2001). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. 282–289.
[7] Ratinov, L., & Roth, D. (2009, June). Design challenges and misconceptions in named entity recognition. In Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009) (pp. 147-155).
[8] Straková, J., Straka, M., & Hajič, J. (2019). Neural architectures for nested NER through linearization. arXiv preprint arXiv:1908.06926.
[9] Shibuya, T., & Hovy, E. (2020). Nested named entity recognition via second-best sequence learning and decoding. Transactions of the Association for Computational Linguistics, 8, 605-620.
[10] Wang, J., Shou, L., Chen, K., & Chen, G. (2020, July). Pyramid: A layered model for nested named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 5918-5928).

[11] Katiyar, A., & Cardie, C. (2018, June). Nested named entity recognition revisited. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1).
[12] Agrawal, A., Tripathi, S., Vardhan, M., Sihag, V., Choudhary, G., & Dragoni, N. (2022). BERT-Based Transfer-Learning Approach for Nested Named-Entity Recognition Using Joint Labeling. Applied Sciences, 12(3), 976.
[13] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[14] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234-1240.
[15] Dinarelli, M., & Grobol, L. (2019). Seq2biseq: Bidirectional output-wise recurrent neural networks for sequence modelling. arXiv preprint arXiv:1904.04733.
[16] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[17] Jain, A., Paranjape, B., & Lipton, Z. C. (2019). Entity projection via machine translation for cross-lingual NER. arXiv preprint arXiv:1909.05356.
[18] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
[19] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[20] Hammerton, J. (2003). Named entity recognition with long short-term memory. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 (pp. 172-175).
[21] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE), 2493-2537.
[22] LeCun, Y. (1985). Une procedure d'apprentissage ponr reseau a seuil asymetrique. Proceedings of Cognitiva 85, 599-604.
[23] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[24] Panchendrarajan, R., & Amaresan, A. (2018, December). Bidirectional LSTM-CRF for Named Entity Recognition. In PACLIC.
[25] Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354.
[26] Che, C., Zhou, C., Zhao, H. Y., Jin, B., & Gao, Z. (2020). Fast and effective biomedical named entity recognition using temporal convolutional network with conditional random field. Math. Biosci. Eng, 17, 3553-3566.
[27] Chalapathy, R., Borzeshi, E. Z., & Piccardi, M. (2016). Bidirectional LSTM-CRF for clinical concept extraction. arXiv preprint arXiv:1611.08373.
[28] Zhu, H., Paschalidis, I. C., & Tahmasebi, A. (2018). Clinical concept extraction with contextual word embedding. arXiv preprint arXiv:1810.10566.
[29] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.
[30] Si, Y., Wang, J., Xu, H., & Roberts, K. (2019). Enhancing clinical concept extraction with contextual embeddings. Journal of the American Medical Informatics Association, 26(11), 1297-1304.
[31] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016, June). Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 260-270).
[32] Ju, M., Miwa, M., & Ananiadou, S. (2018, June). 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, Volume 1 (Long Papers) (pp. 1446-1459).
[33] Wan, J., Ru, D., Zhang, W., & Yu, Y. (2022, May). Nested Named Entity Recognition with Span-level Graphs. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 892-903).
[34] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning (pp. 2048-2057). PMLR.

[35] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 /VA challenge on concepts, assertions, and relations in clinical text. Journal of the American Medical Informatics Association, 18(5):552–556, 2011.
[36] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
[37] Tan, C., Qiu, W., Chen, M., Wang, R., & Huang, F. (2020, April). Boundary enhanced neural span classification for nested named entity recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 05, pp. 9016-9023).
[38] Kim, J.D.; Ohta, T.; Tateisi, Y.; Tsujii, J. GENIA corpus—A semantically annotated corpus for bio-textmining. Bioinformatics 2003, 19, i180–i182. [CrossRef] [PubMed].Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In NIPS, 2014.
[39] Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In NIPS, 2014.
[40] Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050.
[41] Akbik, A., Blythe, D., & Vollgraf, R. (2018, August). Contextual string embeddings for sequence labeling. In Proceedings of the 27th international conference on computational linguistics (pp. 1638-1649).
[42] Finkel, J. R., & Manning, C. D. (2009, August). Nested named entity recognition. In Proceedings of the 2009 conference on empirical methods in natural language processing (pp. 141-150).
[43] Yu, J., Bohnet, B., & Poesio, M. (2020). Named entity recognition as dependency parsing. arXiv preprint arXiv:2005.07150.
[44] Yang, S., & Tu, K. (2021). Bottom-up constituency parsing and nested named entity recognition with pointer networks. arXiv preprint arXiv:2110.05419.
[45] Li, F., Wang, Z., Hui, S. C., Liao, L., Zhu, X., & Huang, H. (2021). A segment enhanced span-based model for nested named entity recognition. Neurocomputing, 465, 26-37.
[46] Alex, B., Haddow, B., & Grover, C. (2007, June). Recognising nested named entities in biomedical text. In Biological, translational, and clinical language processing (pp. 65-72).
[47] Luo, Y., & Zhao, H. (2020). Bipartite flat-graph network for nested named entity recognition. arXiv preprint arXiv:2005.00436.
[48] Liao, T., Huang, R., Zhang, S., Duan, S., Chen, Y., Ma, W., & Chen, X. (2022). Nested Named Entity Recognition Based on Dual Stream Feature Complementation. Entropy, 24(10), 1454.
[49] Cui, S., & Joe, I. (2023). A multi-head adjacent attention-based pyramid layered model for nested named entity recognition. Neural Computing and Applications, 35(3), 2561-2574.
[50] Rudra Murthy, V., & Bhattacharyya, P. (2018). A deep learning solution to named entity recognition. In Computational Linguistics and Intelligent Text Processing: 17th International Conference, CICLing 2016, Konya, Turkey, April 3–9, 2016, Revised Selected Papers, Part I 17 (pp. 427-438). Springer International Publishing.
[51] Sohrab, M. G., & Miwa, M. (2018). Deep exhaustive model for nested named entity recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 2843-2849).
[52] Zheng, C., Cai, Y., Xu, J., Leung, H. F., & Xu, G. (2019). 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 (EMNLP-IJCNLP). Association for Computational Linguistics.
[53] Moen, S. P. F. G. H., & Ananiadou, T. S. S. (2013). Distributional semantics resources for biomedical text processing. Proceedings of LBM, 39-44.
[54] Ratinov, L., & Roth, D. (2009, June). Design challenges and misconceptions in named entity recognition. In Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009) (pp. 147-155).
[55] Lester, B., Pressel, D., Hemmeter, A., Choudhury, S. R., & Bangalore, S. (2020). Constrained Decoding for Computationally Efficient Named Entity Recognition Taggers. arXiv preprint arXiv:2010.04362.
[56] Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., & Anandkumar, A. (2017). Deep active learning for named entity recognition. arXiv preprint arXiv:1707.05928.
[57] Yang, P., Sun, X., Li, W., Ma, S., Wu, W., & Wang, H. (2018). SGM: sequence generation model for multi-label classification. arXiv preprint arXiv:1806.04822.
[58] Najafi, S. (2018). Sequence labeling and transduction with output-adjusted actor-critic training of RNNs.