# Clinical Text Entity Recognition Based on Pretrained Model and BiGRU-CRF

Hanjie Mai[1], Xiaobing Zhou[1,*]

[1]School of Information Science and Engineering, Yunnan University, Kumming 650091, P.R China

**Abstract**

This paper introduces Spanish-based entity recognition for clinical documents, a subtask of lberLEF 2022 called LivingNER. The goal is to identify entities in clinical case documents and annotate whether they belong to humans. Among existing NER models, many of them either fail to learn from context or ignore the cross-domain adaptation problem. To address the above issues, in this paper, we propose an end-to-end model based on Beto. The model uses BiGRU-CRF to encode order information and long-range context-dependency efficiently. Furthermore, we use an adversarial learning method to improve the robustness of the model. Our proposed model has an F1 score of 0.703.

**Keywords**

LivingNER, Beto, BiGRU-CRF, Adversarial Learning

## 1. Introduction

In recent years, due to the significant development of Natural Language Processing (NLP), its related technologies have also been applied to the field of biomedicine. For example, using Named Entity Recognition(NER) technology to extract valuable information such as symptoms and diseases from patients' clinical medical records is of great help for medical personnel to study and diagnose diseases [1]. However, species annotations are rare in NLP research, particularly for non-English content. These annotations are critical for scientific disciplines such as medicine, biology, ecology, nutrition, and agriculture.

To solve the lack of species annotation problem, the LivingNER task presents the challenge of annotating species mentions and entity links through NER techniques, providing a large number of exhaustively annotated Spanish clinical case reports. LivingNER is the first track on exhaustive species mention recognition and the basis of non-English content, with clear potential for multilingual adaptation, especially for scientific species mentions, and aims to generate high-quality biological mention recognition components.

In the LivingNER task, we design an end-to-end model to recognize human and non-human entities from medical records. Benefiting from the pre-trained model transformer [2] and its variants have reached the state-of-the-art in various NLP tasks, our proposed model uses a Beto

IberLEF 2022, September 2022, A Coruña, Spain.

m764720843@gmail.com (H. Mai); zhouxb@ynu.edu.cn (X. Zhou)
https://github.com/33Da/ (H. Mai)
0000-0003-1983-0971 (X. Zhou)

[3] as the encoder. It employs the Spanish corpus for pre-training. Furthermore, inspired by Huang et al.[4] and Liao et al.[5], we propose the structure of BiGRU-CRF to solve the problem of long-range dependencies in sentences and extract text sequence features. Finally, we use an adversarial learning [6] approach to generate an adversarial sample for the word embedding layer of the pre-trained model, and use the adversarial sample for training, so that our model has better robustness [7].

This paper is organized as follows: Section 2 describes the problem definition and model details; Section 3 presents the experimental procedure and results of the model; Sections 4 and 5 discuss post-workshop and summarize our work, respectively.

## 2. System Description

In this section, we introduce the problem definition and various parts of the model. Our model consists of three modules: text representation module, feature extraction module, and entity extraction module. The model is shown in figure 1.

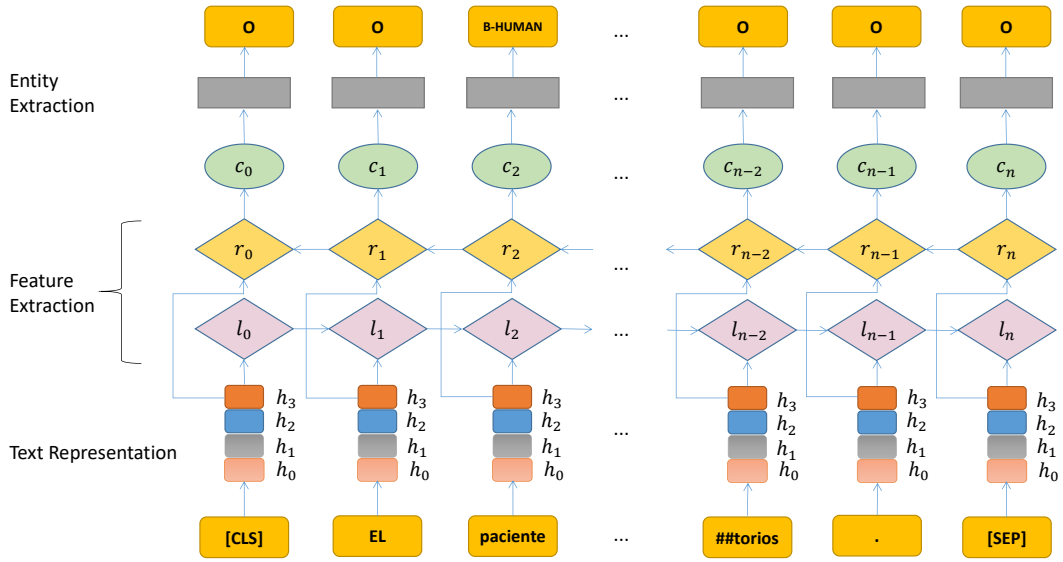


**Figure 1:** Our model consists of three modules: text representation module, feature extraction module, and entity extraction module.

### 2.1. Problem Definition

In the LivingNER corpus, each entity is divided into two categories, human and species. We used the BIO format, which is popular in Named Entity Recognition applications.

- B: The beginning of the entity.

- I: The token is in entity tag range.
- O: The token is outside of the scope.

So that each token of the text could be labeled by the classifiifier according to the 5 categories: O, B-Human, I-Human, B-species, I-species.

## 2.2. Text Representation

For the character information of the text, we convert it to a vector representation and pass it to the model using Beto, which is the Spanish version of Bert [8].

Since the medical record data in the dataset is at the document level, the length of many documents exceeds the maximum text length of 512 supported by Beto. So we first split each document into sentences, turning document-level tasks into sentence-level tasks. These sentences are tokenized according to the dictionary of the pre-trained model Beto.

Then, the WordPiece information is fed into Beto to obtain a text representation vector for each sentence.

Finally, to more comprehensively represent the information of each token, we extract the vectors of the last four hidden layers in Beto and concatenate them according to their last dimension as the final text representation vector.

$$H = Beto(input) \tag{1}$$

$$H_{beto} = [H_n, H_{n-1}, H_{n-2}, H_{n-3}] \tag{2}$$

where $input$ is the mapping index of the dictionary provided by Beto after sentence tokenization. $n$ denote the number of hidden layers. $H$ is all hidden layer vectors generated by the Beto model with dimension $[n, batch\_size, seq\_len, d_h]$. $H_i$ represents the hidden layer vector of the $i$-th layer. $H_{beto}$ means to combine the output vectors of the last four hidden layers with dimension $[batch\_size, seq\_len, d_h * 4]$. $seq\_len$ and $d_h$ represent sentence length and embedding dimension, respectively.

## 2.3. Feature Extraction

In order to solve the problem of long-range dependencies in sentences and extract text sequence features, we use BiGRU network [9] to extract contextual information from the text. Specifically, the tokens in the text are sequentially input into the network cells at each time step, and the cell information is updated through a gating mechanism. A vector $h_{forward}$ containing text forward sequence features and a vector $h_{backward}$ containing text backward sequence features are output and merged to get $[H_{forward}; H_{backward}]$. The dimension is $[batch\_size, seq\_len, d_h * 2]$.

$$H_{forward} = GRU_{forward}(H_{beto}) \tag{3}$$

$$H_{backward} = GRU_{backward}(H_{beto}) \tag{4}$$

$$H_{gru} = [H_{forward}; H_{backward}] \tag{5}$$

## 2.4. Entity Extraction

In this task, we require the output sequence to satisfy some constraints, such as tag B cannot follow tag I in our label scheme. But it is difficult for BiGRU to learn these constraints, so we use CRF [10] to ensure that the output sequence is valid. When training the data, the CRF layer can learn these constraints automatically.

Specifically, we input the output of BiGRU into the linear layer, through which we can obtain the matrix $C$, the dimension is [*batch_size*,*seq_len*,5]. $c_{i,j}$ corresponds to the score of the $j$-th tag of the $i$-th token in a sentence. For a sequence of predictions $y$, we define its score to be

$$s(x, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} c_{i, y_i} \tag{6}$$

where $A_{i,j}$ denotes the score of a transition from the tag $i$ to the tag $j$. A softmax layer over all possible tag sequences yields a probability for the sequence $y$:

$$p(y \mid x) = \frac{1}{Z(x)} \exp(s(x, y)) \tag{7}$$

where $Z(x) = \sum_Y \exp(s(x, Y))$, and Y denotes all possible tag sequences.

## 2.5. Adversarial Learning

To improve the robustness of the model, we use an adversarial learning approach for training. Actually, it is a special kind of data augmentation method. Specifically, we attack Beto's token embedding layer in each batch to obtain optimal adversarial examples under a certain constraint space. They are fed into the model to do gradient updates.

The paper adopts the Projected Gradient Descent (PGD) method [6] for training, which is considered to be the best in first-order adversarial. Compared with the previous Fast Gradient Method (FGM) [11], which obtains the optimal adversarial examples in only one iteration, PGD performs multiple iterations in a batch, each iteration generates a part of adversarial examples. Finally, the adversarial examples generated by multiple iterations are superimposed to obtain the optimal value. The adversarial example generation formula is as follows:

$$r_{adv}^{t+1} = \Pi_{\|r_{adv}\|_F \leq \epsilon} \left( r_{adv}^{t} + \alpha g\left(r_{adv}^{t}\right) / \left\| g\left(r_{adv}^{t}\right) \right\|_2 \right) \tag{8}$$

where $r_{adv}^{t}$ is an adversarial example generated at step $t$. $g(\cdot)$ represents the gradient calculation function. Both $\alpha$ and $\epsilon$ are hyperparameters representing step size and adversarial constraint range, respectively.

## 2.6. Training Loss

The model is trained with cross-entropy loss. Let $D$ be the labeled training data set, $D_j$ is the data representing the $j$-th batch, $y$ and $p$ represent the real labels and the model prediction labels of each token.

$$loss = \sum_{i \in D_j} L\left(y^i, p^i\right) \tag{9}$$

where $L$ is the cross-entropy function.

# 3. Experiment

## 3.1. Dataset

The clinical cases in the LivingNER dataset contain more than 10 different clinical fields and are encoded in plain text UTF8. Each clinical case is stored as a single file. The number of medical cases included in the training set, test set, and validation set is shown in Table 1, respectively.

**Table 1**
Statistics on the number of training, validation, and test documents.

| Train | Validation | Test |
|---|---|---|
| 1000 | 500 | 485 |

Since the longest sentence input length supported by Beto is 512, but many documents are longer than 512, we split the document into sentences using ' . ' as a delimiter. The sentences are then passed into the model. The table below shows the number of sentences in the training and test sets, respectively.

**Table 2**
The number of training, validation, and test set sentences. Because the competition organizer did not provide specific test set documentation, we cannot count the number of sentences in the test set.

| Train | Validation | Test |
|---|---|---|
| 7191 | 3168 | N/A |

## 3.2. Results

In order to prove the effectiveness of the models mentioned above, we propose three additional models for comparison. We use AdmaW [12] as the optimizer with a batch size of 1. A total of 50 epochs are trained. The results are shown in Table 3.

**Table 3**
Rows 1 to 4 in the table are the results of the proposed models on the validation set. We use Beto+BiGRU+CRF+PGD as the final model, and the last two rows show the scores it got in the competition. We made a total of two submissions called 'workshop' and 'post-workshop'.

| | Precision | Recall | Micro-F1 |
|---|---|---|---|
| Beto | 0.8157 | 0.8247 | 0.8202 |
| Beto+BiGRU | 0.8398 | 0.9438 | 0.8888 |
| Beto+BiGRU+CRF | 0.8676 | 0.9471 | 0.9056 |
| Beto+BiGRU+CRF+PGD | 0.8908 | 0.9444 | 0.9168 |
| workshop | 0.1803 | 0.1593 | 0.1692 |
| post-workshop | 0.8214 | 0.6145 | 0.7030 |

**Beto**. Using Beto as the sentence encoder, the encoded vector is directly fed to the linear layer for entity extraction.

**Beto+BiGRU**. First, the sentences are encoded using Beto. Then the BiGRU network is used to extract feature information. Finally, the linear layer performs entity extraction.

**Beto+BiGRU+CRF**. On the basis of the Beto+BiGRU model, the output of the linear layer is sent to the CRF to constrain the output sequence.

Table 3 shows that the Beto+BiGRU+CRF+PGD model achieves a Micro-F1 of 0.9168 on the validation set after introducing adversarial training. Compared with the Beto+BiGRU+CRF model, the Micro-F1 improves by 0.0112.

## 4. Post Workshop

In this contest, we made a total of two submissions. This is because the location of a large number of entities was incorrectly marked in the first commit. We speculate that the reason for this is that when the prediction file is generated, the location of the entity returned by the model corresponds to the sentence. However, the entity location required for the final submission result is the corresponding document.

For the above problem, we restore the sentences to the original document. The position of the entity is then calculated from the starting position of the document.

## 5. Conclusions

This study has introduced the approach of a team named Mark in the subtask LivingNER of lberLEF 2022. A Beto model is introduced to encode clinical text and use BiGRU-CRF for feature extraction and predictive labeling. Finally, adversarial learning is used for training. We think that only the context feature is too simple for feature selection, and the syntactic structure is ignored. Therefore, in future research work, syntactic analysis tools and graph neural networks can be considered to extract syntactic features, then syntactic and contextual features can be combined to predict sentence annotations.

## 6. Acknowledgments

This work was supported by the Natural Science Foundations of China under Grants 61463050.

## References

[1] K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. F. Jones, R. Forshee, M. Walderhaug, T. Botsis, Natural language processing systems for capturing and standardizing unstructured clinical information: a systematic review, Journal of biomedical informatics 73 (2017) 14–29.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[3] J. Canete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, Pml4dc at iclr 2020 (2020) 1–10.
[4] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).
[5] Q. Liao, J. Wang, J. Yang, X. Zhang, Ynu-hpcc at ijcnlp-2017 task 1: Chinese grammatical error diagnosis using a bi-directional lstm-crf model, in: Proceedings of the IJCNLP 2017, Shared Tasks, 2017, pp. 73–77.
[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks (2021).
[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: 2nd International Conference on Learning Representations, ICLR 2014, 2014.
[8] J. D. M.-W. C. Kenton, L. K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
[9] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, in: EMNLP, 2014.
[10] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001).
[11] T. Miyato, A. M. Dai, I. J. Goodfellow, Adversarial training methods for semi-supervised text classification, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=r1X3g2_xl.
[12] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2018.