

# A Tumor Named Entity Recognition Model Based on Pre-trained Language Model and Attention Mechanism

Xin Tao<sup>a</sup>, Renyuan Liu<sup>a</sup> and Xiaobing Zhou<sup>a</sup>

<sup>a</sup>*School of Information Science and Engineering, Yunnan University, Kunming 650091, P.R. China*

## Abstract

Named entity recognition is to recognize the mention of a certain thing or concept in a text in natural language processing, and it is the basis of many natural language processing tasks such as relation extraction, event extraction, knowledge graph, machine translation, and question answering systems. This paper describes the solution of CANTEMIST's named entity recognition subtask. The core idea of the method is to process it as a sequence labeling task and uses a neural sequence model to solve it. We use a pre-trained language model for semantic feature embedding, a recurrent neural network for semantic inference, and a label-based attention mechanism to predict the output. In the final test, our F1 score is 0.719.

## Keywords

name entity recognition, clinic records, multi-head attention mechanism, deep learning

## 1. Introduction

At present, cancer has become one of the major diseases endangering human health. About 9 million people die of cancer every year in the world. It is very urgent to find an effective treatment, but medical resources are difficult to meet the needs in the face of a large number of patients. In a large number of individual clinical records of cancer patients, extracting information and knowledge from them can provide help for curing cancer patients. However, in the face of these free-text clinical records produced by professional doctors, efficient extraction of information and knowledge is a very challenging task. Natural language processing(NLP) technology is a very effective method to process individual clinical records. It can extract information from unstructured or semi-structured data to form structured data for further research.

The Cantemist [1] task provides a well-annotated data set in which experts annotated tumor-related information in the unstructured clinical records of cancer patients. High-quality annotation data is very helpful for developing NLP systems. The task includes three types of subtasks:

1. **CANTEMIST-NER**: finding mentions of tumor morphology in oncology cases.

*Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*


EMAIL: taoxinwy@126.com (X. Tao); bluewind159@qq.com (R. Liu); zhoux@ynu.edu.cn (X. Zhou)

URL: <https://taoxin778.github.io/> (X. Tao)

ORCID: 0000-0002-6883-8403 (X. Tao)



© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

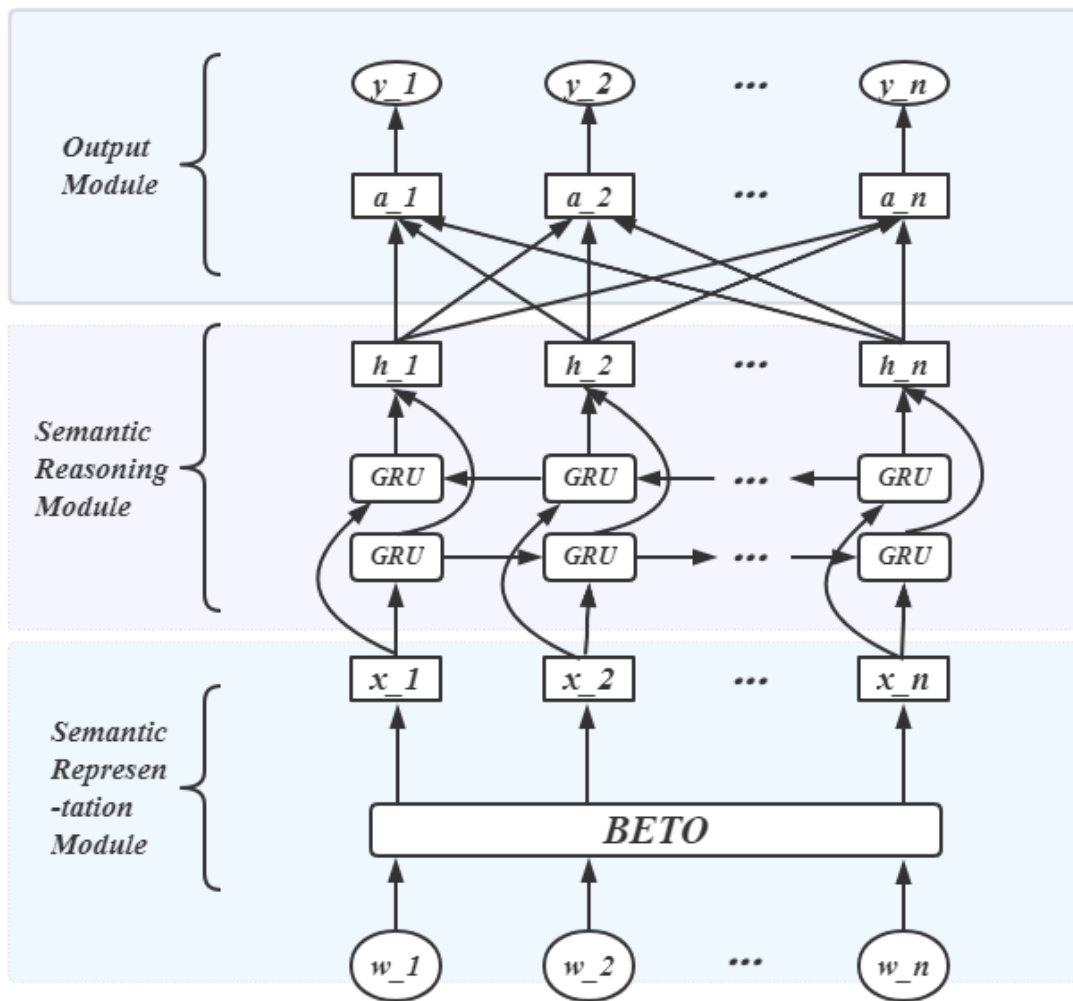
2. **CANTEMIST-NORM**: recognition and mapping to concept identifiers from ICD-O-3.
3. **CANTEMIST-CODING**: oncology clinical coding (multi-label classification) assigning ICD-O-3 codes to clinical case documents.

Our team participates in the CANTEMIST-NER subtask. Named entity recognition (NER) is the basis of NLP tasks such as relation extraction, event extraction, knowledge graph, machine translation, and question answering systems. This subtask is aimed to identify related entity in tumor morphology. A tumor refers to the growth or formation of new abnormal tissues. The tissue and cell types that make up the morphology usually determine the expected growth rate, the severity of the disease, and the recommended type of treatment. We deal with NER using a neural sequence model that combines a pre-training language model with a multi-head attention mechanism, and our model achieves good result in the final evaluation.

## 2. Related Work

NER of clinical text is a very popular research direction, and the categories of recognized named entities are also very large, such as disease, patient, symptom, drug, body state, etc. At present, there are four research methods: rule-based method, dictionary-based method, traditional machine learning method, and deep learning method. The rule-based method [2] mainly relies on the heuristic rules customized by experts. Its advantages are good effect and fast speed. However, due to the limitations of various forms of rules and expert experience, it is impossible to list all the rules. Therefore, the scope of application of the rule-based method is limited and cannot be generalized. In the dictionary-based method [3], a medical dictionary database is used to match entities in text, which is similar to the rule-based method. The disadvantage is that the dictionary needs to be updated in time to ensure the recognition of new entities. The traditional machine learning approach takes the NER task as a sequence labeling task and finds the most reasonable label sequence by modeling the sentence sequence. Common models of traditional machine learning are HMM [4] and CRF [5]. The performance of this method depends very much on feature engineering.

Thanks to the growth of computing resources, deep learning becomes a competitive method to solve NLP problems. The deep learning method also regards the NER task as a sequence labeling task, which can avoid complex feature engineering compared with traditional machine learning method. For biomedical NER tasks, various deep learning models have achieved a pleasing performance. There are two main aspects worth paying attention to: The first aspect is how to improve the reasoning ability of the model. For example, the Bi-LSTM-CRF [6, 7] model can effectively learn the mutual constraints between tags by using CRF as the output layer of the model, thereby improving model performance. The second aspect is to use the large-scale corpus to train language models to obtain better semantic representation, to improve their semantic expression ability. BioBert [8] is a domain-specific language representation model trained on a large-scale biomedical corpus. It achieves state-of-the-art performance on biomedical data sets. Our core idea also refers to these two aspects to build models.



**Figure 1:** Architecture of the whole model. It consists of semantic representation module, semantic inference module, and output module.

### 3. Methodology

In this section, we describe the methods used in the task. Our model is a neural sequence labeling model, which consists of a semantic representation module, semantic reasoning module, and output module. As shown in Figure 1: the semantic embedding module of the model is a pre-train language model similar to BERT [9]; the bidirectional GRU [10] is used for semantic reasoning in the semantic reasoning module; the final output module is based on the attention mechanism [11].

### 3.1. Task and Dataset Description

The goal of the CANTEMIST-NER task is to automatically find tumor morphology mentions in the clinical records of tumor patients. The specific form is to find the corresponding character offsets of tumor morphology mentions in the given *UTF-8* plain text medical document. The entire corpus contains a total of 5982 medical documents, of which 1301 are annotated and 4681 are unannotated. Experts annotate some medical documents by referring to the labeling guidelines that contain rules for annotating morphological tumors in Spanish tumor clinical cases and map these annotations to *CIEO-3*. All entities associated with tumor morphology are labeled *MORBOLOGIA\_NEOPLASI* type, which include the tumor name, anatomical location, histological morphology, and other information. The distribution of the corpus is shown in Table 1.

**Table 1**

The distribution of datasets. The number of clinical records and sentences in each dataset are counted.

	Train	Development1	Development2	Test
Number of Documents	501	250	250	300
Number of Sentences	18628	9114	8349	10775
Number of Entities	6396	3341	2660	3633
Average Document Length	906.90	901.06	732.22	825
Average Sentence Length	20.54	20.80	18.22	19.17
Average Entity Length	2.30	2.27	2.30	2.25

### 3.2. Semantic Representation Module

Semantic representation plays a very important role in the NLP system. Good representational engineering can effectively improve the performance of the system. We choose BETO [12], a pre-trained language model based on Transformer architecture, as the semantic representation module. BETO is a BERT model trained on large Spanish corpus. BETO is trained with the Dynamic Masking technique [13] and Whole-word Masking technique and is an improvement over BERT. And BETO has 12 encoder layers of the transformer with 16 attention-heads, using 1024 as hidden size, and 110M parameters in total. Given a word sequence  $\{w_1, w_2, \dots, w_n\}$ , it is embedded as a high-dimensional semantic vector after passing through the BETO model  $\{x_1^w, x_2^w, \dots, x_n^w\}$ .

### 3.3. Semantic Reasoning Module

In the semantic reasoning module, we use a Recurrent Neural Networks (RNNs) to extract more advanced semantic features. The advantage of RNNs is that it can capture longer distance dependence when processing sequence information, compared with Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs). Gated Recurrent Neural Networks (GRU) and Long Short-Term Memory (LSTM) are two variants of RNN, and both GRU and LSTM are proposed to solve the problem of gradient disappearance or explosion in backpropagation. However, compared with LSTM, GRU can reduce the amount of computation while achieving the same effect.

The multi-layer GRU stack is used to further extract semantic information from the semantic representation module. For a sentence, the embedded semantic representation  $v = \{x_1^w; x_2^w; \dots; x_n^w\}$  of the sentence is obtained after the semantic representation module. Then the embedded semantic representation is fed into the stacked bidirectional GRU layer to obtain a sequence of forward hidden states  $\{\vec{h}_1; \vec{h}_2; \dots; \vec{h}_n\}$  and backward hidden states  $\{\overleftarrow{h}_1; \overleftarrow{h}_2; \dots; \overleftarrow{h}_n\}$ . Finally, the two hidden states are concatenated as the final sentence representation.

$$h_i = [\vec{h}_i, \overleftarrow{h}_i]$$

$$H = \{h_1; h_2; \dots; h_n\}$$

### 3.4. Output Module

During the label decoding phase, we use a hierarchically-refined label attention network [14] that explicitly utilizes label embedding and captures potential long-term label dependencies by giving each word an incremental refinement of the label distribution. Label attention network learns the interaction between semantic representation space and labels representation space through multi-head attention mechanism [11].

Firstly, the given set of candidate labels  $L = \{l_1, l_2, \dots, l_{|L|}\}$  are embedded in the label representation space. The embedding vector of each label  $l_i$  is:

$$x_i^l = f(l_i)$$

The representation space of the label is to randomly initialize the embedded vector of each label, where  $f$  is to map each label to the representation space by looking up the table.

Key-value pairs are used to calculate the distribution of attention  $\alpha$  in the multi-head attention. The sentences output by the semantic reasoning module represent  $h_i$  as the query vector, and the key  $k_j$  vector and value  $v_j$  vector are label embedded vector  $x_j^l$ , namely  $k_j = v_j = x_j^l$ . The matrix form of the standard attention mechanism is as follows:

$$H^l = \text{attention}(Q, K, V) = \alpha V$$

$$\alpha = \text{soft max}\left(\frac{Q^T K}{\sqrt{d}}\right)$$

where  $Q \in \mathbb{R}^{d \times n}$ ,  $K \in \mathbb{R}^{d \times |L|}$ . And  $n$  and  $h$  are the sentence length and the hidden state size of GRU respectively, and the embedding dimension of the label representation feature must equal to  $h$ . The  $|L|$  is the total number of labels. Compared with the standard attention mechanism, the interaction of the multi-head attention mechanism can effectively infer multiple potential label distributions in parallel. The forward propagation of multi-head attention is as follows:

$$H^l = \text{concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_k) + H^w$$

$$\text{head}_i = \text{attention}(W_i^Q Q, W_i^K K, W_i^V V)$$

where  $W_i^Q \in \mathbb{R}^{\frac{d}{k} \times d}$ ,  $W_i^K \in \mathbb{R}^{\frac{d}{k} \times d}$ , and  $W_i^V \in \mathbb{R}^{\frac{d}{k} \times d}$  are a parametric matrix of linear transformation that can be learned in training.  $k$  is the number of parallel heads.

If it is not the last layer of the label attention network, the hidden state of the GRU and the output of attention are connected as the final output:

$$H = [H^w, H^l]$$

The last layer of label attention network directly outputs the weight matrix in the attention mechanism as label prediction:

$$\alpha = \begin{bmatrix} \hat{y}_1^1 \dots \hat{y}_1^{L_1} \\ \hat{y}_2^1 \dots \hat{y}_2^{L_2} \\ \dots \\ \hat{y}_n^1 \dots \hat{y}_n^{L_n} \end{bmatrix}$$

$$\hat{y}_i = \arg \max(\hat{y}_i^1, \hat{y}_i^2, \dots, \hat{y}_i^{L_i})$$

where  $\hat{y}_i^j$  represents the probability that the  $i$ -th word is the  $j$ -th label, and  $\hat{y}_i$  represents the predictive label of the  $i$ -th word in the sentence.

## 4. Experiments

This section introduces the experimental of the model, mainly including data preprocessing, experimental setting, and error analysis.

### 4.1. Data Preprocessing

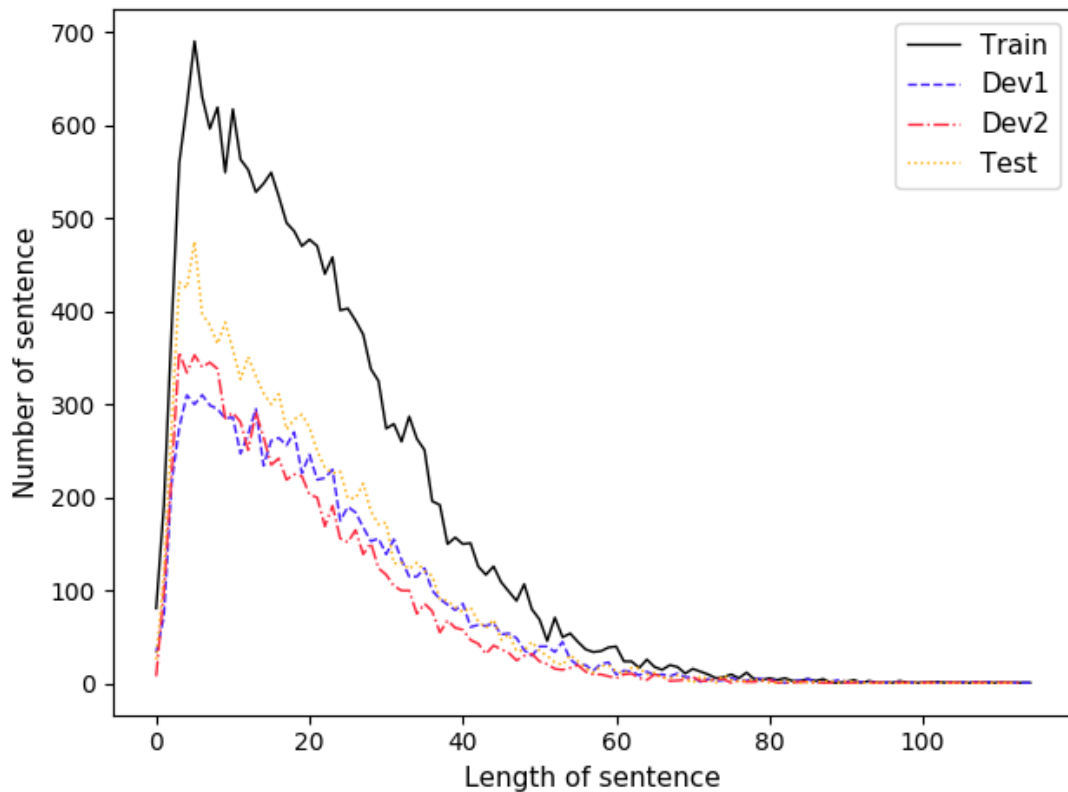
In the data preprocessing stage, we divide the text according to the sentence level and word level. Then, according to the annotation file, all the sentences are labeled in the format of BIO. After getting the labeling sequence of each sentence, we clean the sentence. Specific operations include deleting all punctuation marks and special characters; replacing numbers with "Cifra"; deleting single-word sentences (mainly subtitles). As shown in Figure 2, the sentence distribution after preprocessing is counted to provide a reference for the model's sentence length hyperparameter setting. As can be seen from Figure 2, most sentences are less than 80 in length. Considering the Bert model, a masked language model is used for training. In the masked language model, 15% of the words are divided into several unknown words, and the unknown words are predicted by other words in the sentence. In the semantic embedding module of our model, the hyperparameter of sentence length is set to 100 to reduce the noise caused by excessive zero paddings.

This task is the same evaluation as most NER tasks, using the F1 score as an evaluation metric. Because this task only defines one type of entity, there is no data imbalance problem. The micro-average F1 score can be used to effectively evaluate the system performance. The calculation formula is as follows:

$$precision(P) = \frac{TP}{TP + FP}$$

$$recall(R) = \frac{TP}{TP + FN}$$

$$F - measure(F_1) = 2 \frac{P \times R}{P + R}$$



**Figure 2:** Sentence length distribution of training set, development set, and test set. In all datasets, 97.3% of sentences are less than 80 in length.

## 4.2. Results

According to Section 3.4, if the label attention network has only one layer, it is equivalent to using the Softmax layer. So we compare the performance of the label attention network with different layers. When the number of layers of the label attention network is set to 1, 2, 3, 4, and the results on the development set are shown in Table 2. We find that the performance of the development set can be improved, as the number of layers of the tag attention network increases. However, when the number of layers is increased to 4, the result is not improved. Besides, the embedded dimension of the label or the hidden state size of the GRU also affects the performance of the model.

To compare the effect of the label attention network, we conduct experiments with BERT’s token classification model as a comparison. BertForTokenClassification<sup>1</sup> is a fine-tuning method of the BERT model in downstream tasks, and it has obtained an F1 score of 0.924 on the CoNLL-2003 named entity recognition data set. It uses BERT as the semantic embedding and then uses the fully connected layer as the output model. We replace the semantic embedding module as the semantic embedding of BertForTokenClassification, to keep the semantic embedding

<sup>1</sup>[https://huggingface.co/transformers/model\\_doc/bert.html#bertfortokenclassification](https://huggingface.co/transformers/model_doc/bert.html#bertfortokenclassification)

**Table 2**

The results of the development set. The "Number of layers" refers to the number of layers embedded in the label attention network. The "Size" is the dimension of the label embedding vector of and the hidden state size of GRU.

Number of Layers	Size	F1	P	R
1	200	0.708	0.737	0.682
1	400	0.711	0.728	0.694
2	200	0.713	0.733	0.694
2	400	0.713	0.735	0.691
3	200	<b>0.72</b>	0.727	<b>0.714</b>
3	400	0.713	<b>0.739</b>	0.69
4	200	0.714	0.732	0.697

the same. The comparison between the results our model and of BertForTokenClassification is shown in Table 3.

**Table 3**

Comparative experiment results with Baseline and BertForTokenClassification.

	Precision	Recall	F1
Baseline	0.181	<b>0.737</b>	0.291
BertForTokenClassification	<b>0.721</b>	0.685	0.703
Our_Moudel	0.716	0.721	<b>0.719</b>

We also fine-tune the other hyperparameters of the model, and Table 4 shows all the hyperparameters of the model submitted to the final evaluation results. And the F1 score of the result on the test set is 0.719. Our experimental implementation is based on the PyTorch [15] framework and the transformers package [16] is used to get the pre-trained language model. In the data preprocessing stage, the Spacy [17] is used for tokenization and tagging.

**Table 4**

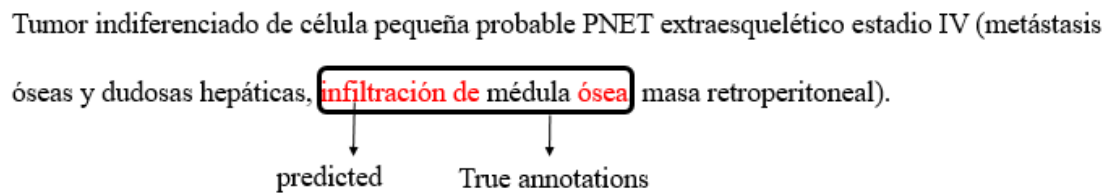
Hyperparameters settings of the final model.

Hyperparameters	Value
Max Length of Sentence	100
The Dimension of Label Embedding	200
The Hidden State Size of GRU	200
The Number of Head	5
The Number of Attention Layer	3
Dropout Rate	0.5
Learn Rate	0.0001
Batch Size	16
Epochs	15



### 4.3. Error Analysis

We analyze the error of the final submission result, and there are two main errors: the wrong recognition of the entity boundary and the missed recognition of the entity. For the first type of error, as shown in Figure 3, the true annotation is "*infiltración de médula ósea*", but the model recognizes two sub-parts "*infiltración de*" and "*ósea*". We believe that the model may not be able to learn the semantic dependency information of long entities because the average entity length in the training set is 2.30. Therefore, when recognizing such long entities, the model tends to recognize them as multiple short entities, resulting in the precision of 0.716. Regarding the second error, we believe that the model's ability to model sentence semantics is insufficient and cannot fully recognize entities. This also results in a low recall, with a final recall of 0.721.



**Figure 3:** A case of wrong identification of entity boundary.

## 5. Conclusion

We develop an automated system based on the Cantimist dataset to effectively identify cancer entities in clinical texts. The system consists of three parts: the semantic embedding module, the semantic reasoning module, and the output module. In the semantic embedding module, BEOT, a BERT model trained in Spanish, embeds the text information into a high-dimensional semantic space and obtains the semantic representation of the text. In the semantic reasoning module, a bidirectional GRU that can capture long-distance dependencies is used to model sentences in the forward and backward directions. In the output module, a label-based multi-head attention mechanism is used to infer the constraint relationship between labels. We select the best hyperparameters through tuning experiments and obtain good results compared with the baseline in the final evaluation.

However, our model still has a lot of room for improvement. There are some defects in the model's ability to model sentence semantics, such as some entities missing and entity boundary error. Our future work is mainly to solve the problem of model modeling of sentence semantics. It may be that this data set is a professional text about tumor morphology and BEOT is trained on ordinary corpus. The model we proposed is not able to model sentence semantics well. In the next work, we will try to use graph neural networks to model text semantics to have better results in the recognition of long entities.

## Acknowledgments

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

## References

- [1] A. Miranda-Escalada, E. Farré, M. Krallinger, Named entity recognition, concept normalization and clinical coding: Overview of the cantemist track for cancer text mining in spanish, corpus, guidelines, methods and results, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), CEUR Workshop Proceedings, 2020.
- [2] G. K. Savova, J. Fan, Z. Ye, S. P. Murphy, J. Zheng, C. G. Chute, I. J. Kullo, Discovering peripheral arterial disease cases from radiology notes using natural language processing, in: AMIA Annual Symposium Proceedings, volume 2010, American Medical Informatics Association, 2010, p. 722.
- [3] M. Song, H. Yu, W.-S. Han, Developing a hybrid dictionary-based bio-entity recognition technique, BMC medical informatics and decision making 15 (2015) S9.
- [4] S. Zhao, Named entity recognition in biomedical texts using an hmm model, in: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP), 2004, pp. 87–90.
- [5] L. Ratinov, D. Roth, Design challenges and misconceptions in named entity recognition, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), 2009, pp. 147–155.
- [6] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).
- [7] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv preprint arXiv:1603.01354 (2016).
- [8] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.
- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [10] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).
- [11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
- [12] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Spanish pre-trained bert model and evaluation data, in: to appear in PML4DC at ICLR 2020, 2020.
- [13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [14] L. Cui, Y. Zhang, Hierarchically-refined label attention network for sequence labeling, arXiv preprint arXiv:1908.08676 (2019).

- [15] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).
- [16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Huggingface’s transformers: State-of-the-art natural language processing, ArXiv abs/1910.03771 (2019).
- [17] M. Honnibal, I. Montani, spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing, To appear 7 (2017).