Applying NER in Medical Records

Tran The Trung, Phan Thanh Binh, Tran Minh Trung, and Chau Phan Phuong Mai

Data Science, University of Science and Technology of Hanoi

***Abstract*—Health records are essential documents that contain valuable information about not only patient information but also the history of treatment. However, the availability of massive medical information needs to be processed and analyzed, which is time-consuming and requires lots of specialized knowledge. Therefore, in the study, we attempt to automate the extract information procedure from those documents following two approaches: bi-LSTM-CRF and BERT variations. MACCROBAT dataset is utilized to evaluate the performance of different models. Both approaches produce acceptable results. The former approach, despite the simpler architecture design, shows better result.**

***Keywords*—Electronic Health Records, Named Entity Recognition, BERT, LSTM.**

I. Introduction

In the biomedical field, health records are an essential document for informed decision-making. They contain information about a patient's medical history, such as his or her past diagnoses, medications, allergies, test results, etc. These documents exist for several reasons. Firstly, they provide a comprehensive overview of an individual's health background, which is crucial for healthcare providers to make precise decisions about the patient's future treatments. Secondly, they support the patient's continuity of care between different healthcare facilities by ensuring that everybody involved in the patient's care is on the same page, so that treatment is optimized effectively. Moreover, it is also an important resource for researchers to find new treatments. Thus, analyzing and gaining in-depth understanding from health records has long been a vital task. To streamline this process, the healthcare industry has transitioned to electronic health records (EHRs). This shift, evident in the widespread adoption of EHRs by hospitals, facilitates easier accession and management of patient data.

However, that is not enough. Manually reading this document is labor-intensive since not only specific domain knowledge is required but also high processing speed as these records are growing in volume, variety, and complexity. One of many ways computers can help alleviate the burden on human workloads is by swiftly and automatically extracting keywords. In this work, to achieve such ability, we tried some deep learning models to carry out the Named Entity Recognition (NER) task on text documents specialized in the biomedical field. The NER model is a type of artificial intelligence (AI) program used in Natural Language Processing (NLP) tasks. Its job is to identify and categorize words into specific, predefined entities within a text. In the content of medical records, these entities include entities such as diseases and disorders, medications, symptoms, and more. Applying an NER model to medical records takes advantage of the fact that it extracts key information faster, which saves time and increases performance for downstream tasks.

The objective of this study is to implement a model that can automate extracting medical information from EHRs using two named entity recognition approaches: sequence labeling-based and span-based. The expected outcomes are demonstrated in Figure 1.

Fig. 1: Example for the model's output.

The remaining sections of the report are organized as follows:

- Section II gives us the general view of some NER methods nowadays.
- In section III, we presented the chosen dataset, along with our visualizations, analysis and the preprocessing steps conducted.
- In section IV, we illustrated the chosen models for the NER task.
- In section V, we showed the perfomance evaluation of those models.

II. Literature Review

For Named Entity Recognition problem, people usually have two separate approaches which is sequence labeling-based methods and span-based methods.

a) Span-based: This method will try to predict the start and end positions of each entity directly. So given a sentence, the output of the method would be a set of entity spans along with their corresponding labels. And thus, transformer-based models would be the appropriate candidates for this approach since they have the ability to handle a whole sentence

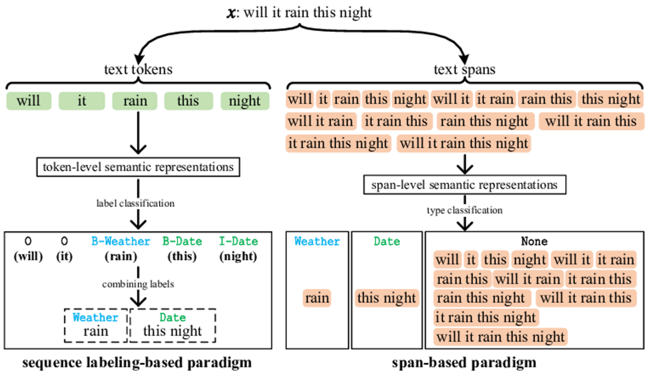


Fig. 2: The sequence labeling-based and the span-based[4].

in parallel and help us resolve the problem of nested or overlapping entities

b) Sequence labeling-based: This method works on the problem by assigning labels to each individual token. In other words, each token will be assigned a label indicating its entity type and whether it is the beginning of the entity or not. RNN models and their variants like LSTM or GRU are some of the most typical ones that represent this approach for their strategy of executing sequentially token by token by maintaining a memory or state of previous inputs.

For this project, we will be using BERT and its variants for span-based approach and a hybrid model of CNN-LSTM-CRF for the latter one.

III. Dataset

We utilized MACCROBAT[1], one of the most common datasets for automated labeling (i.e. named entity recognition and relation extraction) purpose in biomedical documents. According to the authors, the dataset is the result of their innovative typing system for clinical texts which is able to accurately represent the variety of terms and events used in clinical records without demanding connections to organized concepts or terminology. 200 clinical case reports (CCRs) are included with manual annotations in the standoff format[1]. However, the standard data format required for NER task is BIO tags [2] which means documents must be separated into tokens and each token must be assigned to a label of whether it is the beginning, inside or outside of the annotated entities. Fortunately, this standardization task has already been done by an user in the Hugging Face community, result in this dataset [3] which we used directly. The data will be split 90% for training and 10% for testing the model. Figure 3 show an example of a document from the dataset.

A. Visualization

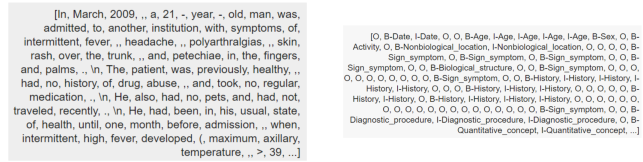
We used the histogram Figure 4 for showing all entities. There are 41 different special terms. Two of the most frequent entity are Sign_symptom and Diagnostic_procedure. This can be easily explained since there should be the symptoms of the patients and also how they can treat it through diagnostic procedures.

In case of Figure 5 illustrates the word clouds for the tokens composed the entities in the dataset. The word clouds provide insight into some of the most frequent words that appear in our dataset. It can easily be seen that the largest words are medical terms such as CT, mass, or pain which are unsurprising for medical records

B. Preprocessing

To feed the list of tokens in Figure 3 into the model, we need a way to represent them as vectors, this step is known as encoding or embedding. To carry out this, we used the Byte

[1]http://brat.nlplab.org/standoff.html
[2]https://en.wikipedia.org/wiki/Inside-outside-beginning_(tagging)
[3]https://huggingface.co/datasets/ktgiahieu/maccrobat2018_2020


[In, March, 2009, ,, a, 21, -, year, -, old, man, was, admitted, to, another, institution, with, symptoms, of, intermittent, fever, ,, headache, ,, polyarthralgias, ,, skin, rash, over, the, trunk, ,, and, petechiae, in, the, fingers, and, palms, ., \n, The, patient, was, previously, healthy, ,, had, no, history, of, drug, abuse, ,, and, took, no, regular, medication, ., \n, He, also, had, no, pets, and, had, not, traveled, recently, ., \n, He, had, been, in, his, usual, state, of, health, until, one, month, before, admission, ,, when, intermittent, high, fever, developed, (, maximum, axillary, temperature, ,, >, 39, ...]

[O, B-Date, I-Date, O, O, B-Age, I-Age, I-Age, I-Age, I-Age, B-Sex, O, B-Activity, O, B-Nonbiological_location, I-Nonbiological_location, O, O, O, O, B-Sign_symptom, O, B-Sign_symptom, O, B-Sign_symptom, O, O, B-Sign_symptom, O, O, B-Biological_structure, O, O, B-Sign_symptom, O, O, O, O, O, O, O, O, O, O, O, B-Sign_symptom, O, O, B-History, I-History, I-History, I-History, I-History, O, O, O, B-History, I-History, I-History, O, O, O, O, O, B-History, I-History, O, B-History, I-History, I-History, I-History, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, B-Sign_symptom, O, O, B-Diagnostic_procedure, I-Diagnostic_procedure, I-Diagnostic_procedure, O, B-Quantitative_concept, I-Quantitative_concept, ...]

Fig. 3: Example from the dataset.

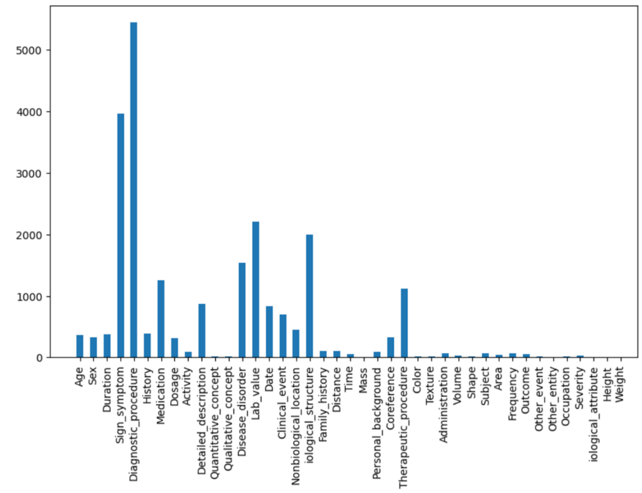


Fig. 4: Histogram of entities in the dataset.

Fig. 5: Word Clouds of words in the dataset.

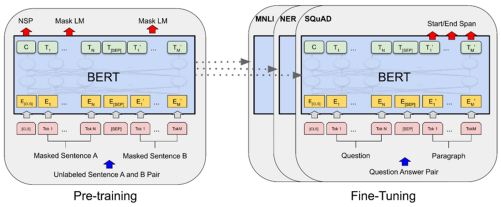


Fig. 6: BERT's architecture.

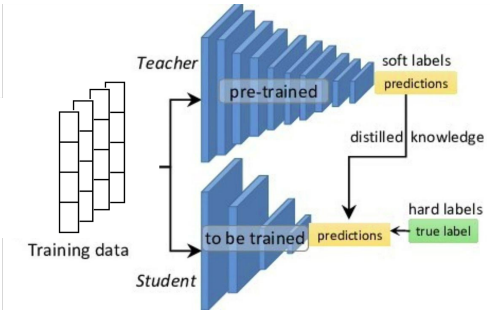


Fig. 7: DistilBERT's architecture.

Pair Encoding (BPE) technique [4]. BPE maps tokens from the corpus to their indexes in the vocabulary, but the noticeable feature is that it can handle out of vocabulary words with minimized reliance on the *unknown* token by breaking them down into known sub-word units until character level. This is beneficial when we want to reduce the vocabulary size or fine tune a model with pretrained vocabulary.

IV. Methods

A. Span-based methods

1) BERT[2]: In most cases, training the model from scratch in NLP is not feasible because of the complexity of the model and because of it, making use of a pre-training language model is one of the easiest ways to increase the overall performance. There are two main ways of implementing pre-trained model. One is the feature-based approach which applies a corresponding architecture with a separate task to make use of the feature from a pre-trained model such as ELMo. The other way is to train downstream tasks by fine-tuning all the parameters. This is called a fine-tuning approach with an outstanding example of GPT. However, both of these methods use a unidirectional language model meaning the model only takes the previous token as the input which is a huge constraint. And to resolve the problem, researchers have come up with the idea of using the Masked Language Model (MLM). It works by masking some tokens and letting the model predict the middle word based on the surrounding context. This new method has been a huge breakthrough and it's named Bidirectional Encoder Representations from Transformers (BERT).

BERT framework consists of two steps. One is to pre-train on unlabeled data and then we fine-tune on labeled data of the downstream task. A special point about BERT is that it makes use of a single architecture for all of the tasks which minimizes the gap between pre-trained architecture and final downstream architecture.

- Pre-training: BERT use two unsupervised tasks:

Mask LM: This procedure randomly masks some tokens by 15%-symbol by a token called [MASK] and then forces the model to predict it. However, this might create a difference in fine-tuning datasets where there are no such kind of [MASK] tokens. To alleviate this, in those 15%-symbol chosen tokens, only 80%-symbol of them were actually masked. 10%-symbol will be replaced by a random token while 10%-symbol will remain intact. The model will not know which token it needs to predict until the last layer so it has to remember the contexts of all input tokens. Next Sentence Prediction (NSP): Not only is the connection between tokens, but BERT also needs to learn the relationship between tokens for some specific downstream tasks. In other words, the researchers have to train the model so as to it can learn to generate from a corpus. More specifically, we will generate a pair of sentences A and B where B might be the sentence right after A or a completely random sentence in a 50/50 probability and the model will predict whether A and B are truly a pair of sentences.

- Fine-tuning:

The fine-tuning task is simply to put the input and output and then fine-tune all the parameters or in our case, we will fine-tune it on NER tasks which are on token levels.

2) DistilBERT[7]: One problem with BERT is its huge complexity which requires high computational resources. Then how can we make it smaller and faster while still maintaining a comparable performance? For this, we have a variant called DistilBERT which makes use of knowledge distillation to transfer knowledge from a larger model to a smaller one. The intuition behind the distillation method is to have two models, one is a teacher model which is a large one that has been trained on a huge dataset. This will be used as a "guidance" for a smaller model called the student model which will be our main model of deployment. To do this, normally the student model will integrate a small part of the output from the teacher model instead of only the ground truth. In other words, we will try to force the student model to mimic the distribution to make it more similar to the teacher. This, in turn, alleviates the training process of the student model, making it more compact and requiring less computational resources. Additionally, the teacher model also creates what we call "soft targets" for the training process. Soft target having high entropy might give out more information which means the student can be more generalization in practical situations. Applying this idea to BERT, people have successfully created DistilBERT which generally has the same architecture as BERT but reduces the number of layers by a factor of 2. This creates a version of BERT that is more efficient while still being quite effective for various NLP tasks.

[4]https://huggingface.co/learn/nlp-course/chapter6/5

3) RoBERTa[5]: Another variant of BERT is RoBERTa which stands for A Robustly Optimized BERT Pretraining Approach. It is designed based on the same architecture of BERT but integrates some advanced techniques to improve the overall performance.

- Dynamic masking: The original BERT implements a static masking where we put the [MASK] token in the same place every epoch and they avoid using the same mask by duplicating the training data 10 times so that each sequence is masked in 10 different ways. Still, each training sequence is seen with the same mask four times during the training procedure. RoBERTa uses a form of dynamic masking where the masking pattern changes between training epochs. This helps the model to generalize better by preventing it from memorizing the repeated patterns generated by the [MASK] token.
- Next Sentence Prediction: Another change is that RoBERTa removes the NSP task during pretraining which turns out to slightly improve the downstream task performance.
- Large Mini-Batches: RoBERTa also uses a larger batch size which allows it to increase the parallelism and the efficient in training.
- Hyperparameters Optimization: The RoBERTa researchers have found out the sensitivity of training to the Adam epsilon term so they also focus on tuning it. They have changed some parameters like peak learning rate or number of warmup steps. And lastly, RoBERTa is trained only with full-length sentences.

4) DeBERTa[3]: DeBERTa stands for Decoding-enhanced BERT with disentangled attention. This is the only variants that make some changes to the original architecture of BERT to imporve the attention mechanism and decoding process.

- Disentangled Attention: In BERT or in most of the traditional Transformer-based model, each word is represented using a content embedding and optionally adding a position embedding. DeBERTa, on the other hand, introduces a disentangled attention mechanism that separates the positional and content information of each token in the input layer. This adds more information of the relationship between each token in a sentence
- Enhanced Mask Decoder: The disentangled attention mechanism already takes into account the content and relative position of the token but not its absolute position. This is usually ignored in most of the model but still quite crucial for the final prediction. DeBERTa solves this by incorporating the embedding of this information right before the softmax layer.

B. Sequence labeling-based method

For this approach, we use a hybrid method of Bi-LSTM-CNN-CRF[6]. This model contains three main components, which is showed in Figure 8.

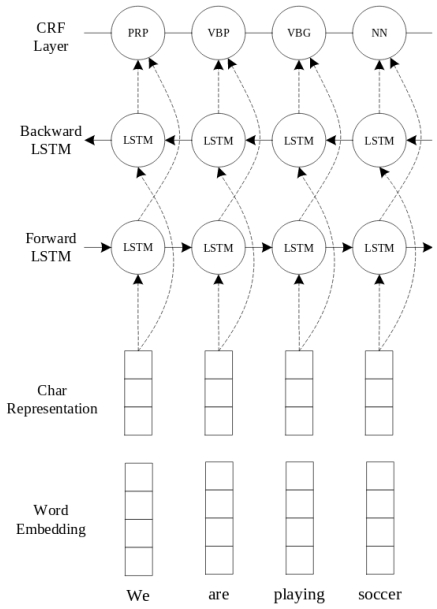


Fig. 8: The architechture of Bi-LSTM-CNN-CRF.

a) CNN: Convolutional Neural Network or CNN is often used in Computer Vision tasks since it can capture the spatial relationship between pixels of an image. And in this case, we will also make use of CNN to generate the character-level representation for each word. The intuition is that CNN will be able to know the spatial coherence across characters. Then, we use maxpool on top of this layer to extract meaningful features which will in the end give us a dense vector representation of each word. For the final embedding vector, we will concatenate the result with the pre-trained GloVe embeddings to create a unified representation for each word.

b) Bi-LSTM: Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network that was first designed to resolve the problem of long-term dependencies in data. Traditional RNN usually struggle with vanishing gradient when the model can not train on long sequences and LSTM handles it with the help of memory cells and some additional gates: forget gate, input gate, and output gate. The term ”Bi” in front stands for Bidirectional which indicates that the information will travel both forward and backward, making it richer and potentially more informative.

c) CRF: Continuous Random Field (CRF) is a probabilistic discriminative model which is usually used in prediction tasks for sequences. Unlike other predictive models for sequences, CRF does not require any independence assumptions or specific set of features. Thus, it can capture dependencies between input and output in a much more complex and flexible way. Conventional deep learning models designed for sequential input does a great job of constructing features as well as capturing reliance between input items by utilizing outputs of previous layers as input for current layer. However, they forget to take into account the order

of the labels since each output is mapped to the desired label separately. Therefore, we can use CRF as an extension to existing LSTM model to capture that dependency. More specifically, CRF use 1 to predict the probabilistic score for the whole token sequence X with label sequence Y rather than each token-label pair individually.

$$p(Y|X) = \frac{\exp(\sum_{k=1}^{K} F_k(X,Y))}{\sum_{Y \in Y'} \exp(\sum_{k=1}^{K} F_k(X,Y'))} \quad (1)$$

V. Evaluation

For the evaluation, we will use F_1 score to compute the model's accuracy. It is the trade-off between precision and recall which is highly useful in our case where the class distribution for NER is usually imbalance.

$$F_1 score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (2)$$

TABLE I: $F_1 Score$ of different models.

CNN-LSTM-CRF	BERT	DistilBERT	RoBERTa	DeBERTa
0.865	0.818	0.782	0.82	0.84

Surprisingly, the hybrid LSTM model surpasses all the BERT ones. This really shows the LSTM with the help of some other advanced techniques like CNN or CRF can still be compared to transformer-based models and it is not outdated yet. For the rest, DeBERTa and RoBERTa really show their improvements compared to the traditional model with some new techniques and research. And DistilBERT, unsurprisingly, performs the worst since it is the most compact model out of all of these but still maintains quite a good performance.

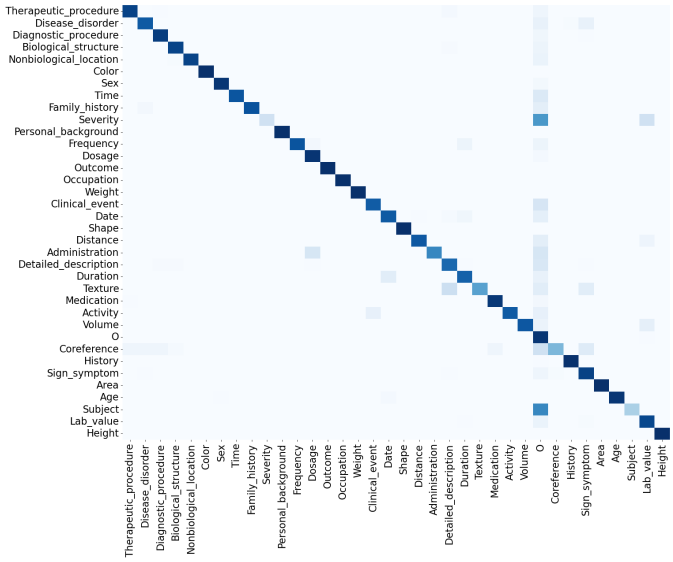


Fig. 9: Confusion matrix of Bi-LSTM-CNN-CRF model.

The Figure 9 shows the details of our best-performing model. One noticeable point worth mentioning is that most of the mistakes made by the model are classifying the entity as O, or in other words, not being able to detect the entity. Meanwhile, the errors in misclassifying are quite low. For the rest of the matrix, most of the entities are predicted quite well and there's no huge error that should be investigated more in.

VI. Conclusion

In this project, we have been able to implement several state-of-the-art NER models and compare their performance on a medical dataset. In the end, we see that LSTM with the aid of some other models has been able to compete with some high-complexity transformer based models and produce quite a good result.

Still, there are some problems we are still struggling with during the project. First and foremost, training LLMs model is quite time-consuming and requires huge computational resources. And secondly, the limit of the diversity of the available dataset is also a constraint for us. And thus, in the future, we hope to find different methods apart from LLMs such as Graph Neural Networks, a promising approach for NER nowadays. Secondly, we will try to research and integrate more datasets to increase the number of detected entities while also focusing more on in-depth data in the field of medicine.

References

[1] J Harry Caufield et al. "A Comprehensive Typing System for Information Extraction from Clinical Narratives". In: *medRxiv* (2019).

[2] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[3] Pengcheng He et al. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. arXiv: 2006.03654 [cs.CL].

[4] Bin Ji et al. *Win-Win Cooperation: Bundling Sequence and Span Models for Named Entity Recognition*. 2022. arXiv: 2207.03300 [cs.CL].

[5] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[6] Xuezhe Ma and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: https://aclanthology.org/P16-1101.

[7] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].