Feature Symptoms Mapping for Scoring Clinical Patient Notes with BERT-CNN-BiLSTM: Final Report

G099 (s2221549, s2170508, s2185096)

Abstract

This project demonstrated and evaluated the mapping from patient notes to standard expressions of feature symptoms using natural language processing (NLP) models with 1000 patient notes data taken by medical students based on the self-description of 10 standard patients. Here, several models are reproduced for comparison, and we proposed an advanced network architecture that obtained satisfactory f1-score and accuracy of 0.8641 and 0.8366, which outperforms other models we reproduced and proves the potential of using NLP to develop a more precise and efficient patient note scoring system.

1. Introduction

Identifying symptoms and writing patient notes are important skills for a licensed physician. Therefore, training and assessment are needed for medical students to learn how to record the complaint of the patient, physical examination results and possible diagnoses. As the assessment process is human labour and time-consuming, especially in the circumstance that qualified physicians are already under work pressure nowadays, auto-scoring systems are being developed by the industry. In recent years, Natural language processing (NLP) has been leveraged to develop a computer-aided grading system for patient notes in the USMLE Step 2 Clinical Skills Exam (Salt et al., 2019). However, the grading performance is limited due to key feature symptoms that are significant for diagnosis can be expressed in various ways (e.g., the key feature 'loss of interest in activities can be interpreted as 'not doing exercise anymore'). Therefore, mapping patient expressions in the patient note with standard key features are fundamental for accurately scoring medical students' notes. In this project, we are working on 'translating' and 'summarising' patient notes into standard feature symptoms by implementing a natural language processing model with 1000 patient notes taken by medical students based on the self-description of 10 standard patients to be the dataset. To better understand the overall performance of different models on this specific task, several models, namely the BERT baseline, BERT-LSTM, BERT-LSTM(CRF), BERT_BiLSTM and CNN-BiLSTM are reproduced to test the result. Then, an advanced BERT-CNN-BiLSTM model is designed to train a better model. Based on the experiments, our self-proposed BERT-CNN-BiLSTM network outperformed other reproductions and achieved an f1-score of 0.8641 and an accuracy of 0.8366. This project reveals the potential for NLP to map patient note expression into the standard expression of feature symptoms. By doing so, the examination scoring criteria are enabled to be fully computational, contributing to precise auto-scoring that improves grading efficiency, avoids human scoring bias, and reduces the human labour cost.

2. Related work

2.1. Healthcare Text Processing

Dealing with medical or healthcare texts has always been a challenging task. Although clinical texts are written in existing natural languages, the complex symptom descriptions, peculiar terms, and capricious grammatical forms make them more like a special language. Clinical texts come from diverse sources, most of which do not follow a uniform writing format even if they contain some common writing habits. All obstacles make the NLP task of clinical texts difficult, while the value of processing these texts is often evident.

Processing clinical texts can be an efficient way to improve the quality of clinical nursing (Dzau & Ginsburg, 2016). (Marafino et al., 2018) provides a model for processing-intensive care patient information and predicting their results using NLP. They extracted data from the first day of hospitalizing ICU for 101,196 patients to predict mortality and found that NLP treatment significantly improved outcomes. Accuracy can reach 0.923 and 0.897, respectively, by training the data of one hospital and predicting the data of another hospital. (Tissot et al., 2020) utilizes NLP in the review of Electronic Health Records (EHR) to simulate trial recruitment in clinical care.

2.2. History of NER

Based upon the judgment of the core ambition of the task we need to achieve, which is mapping patient notes content into fragmentary feature text, we can estimate our overall task as NER problem, which refers to named entity recognition. It is a problem of naming different types of entities in a text (Nadeau & Sekine, 2007). In the past decade, advanced algorithms are basically composed of model structures such as RNN, LSTM, CNN, BERT and CRF, and some excellent-result architectures bring some inspiration for our work.

NER tasks are often regarded as a word-based NER or character-based NER according to the smallest unit of operation on features. Before the BERT pre-training model came out, most researchers chose to do character-based NER since there were not enough large data to obtain a satisfying result of word embedding. Heaps of tasks in minority languages tend to be processed by word-based NER. In (Şeker & Eryiğit, 2012), a Turkish NER structure based only on CRF model is proposed, and excellent results are obtained for naming common entities in Turkish. The authors of CharNER (Kuru et al., 2016) constructed the network of bi-LSTM stacks that operate on character features to avoid language disparities. This character level NER achieved the most advanced results of its time in seven languages. In (Chiu & Nichols, 2016), both word-level and character-level features were extracted by combining bi-LSTM and CNN, and the results on NER task were exceptional.

2.3. NER for medical texts

Even though the NER task for medical texts is not currently the hottest project, it is still a significant and potential application. Based on the structure described above, the advantages of other text processing can be replicated in these medical texts. In the biomedical field, many convoluted and obscure terms and features are urgently needed for classification, especially in the high-level embedding design of them (such as rare feature terms composed of low-level words). In order to solve this problem, authors of (Cho et al., 2020) combined bi-LSTM and CNN to enhance the effect of the attention mechanism on features and then combined CRF to obtain state-of-art results on the JNPBA dataset. For Electronic Health Records (EHRs) processing, the baseline of (Hofer et al., 2018) was built on a bi-LSTM infrastructure and adjusted by other parameters to achieve the best results. Ingeniously, the authors devised a method to merge char-level, word-level and case-level features as inputs, enabling the network to learn more information. Our network structure also has reference to part of this design.

3. Dataset and task

In this section, general ideas about the source of data and the expected tasks will firstly be given. Then, information about data volume, dataset splitting, and labelling methods, along with some other data-related implementation details, will be introduced.

3.1. Data Availability and Task

In this project, the data are patient notes made by medical students obtained from the USMLE® Step 2 Clinical Skills examination (Papadakis, 2004), which is not composed of real case data. Instead, patient notes in this exam are collected from the diagnoses of standardised patients' medical conditions, thereby not involving any ethical and privacy issues, and the data are ensured to be correct and reliable. The annotations were made by professional physicians and indicated strings from each note that correspond to specific types of standardised expressions of feature symptoms. Our primary task is to extract the standardised expressions of feature symptoms when given an input patient note, which can be regarded as a name entity recognition (NER) task. Meanwhile, we will compare the performance of different NLP combinations under this task in order to reveal some insights and seek a satisfying result.

3.2. Detailed Description of Data

All training data is contained in three files: *features.csv*, *patient_notes.csv* and *train.csv*. The *patient_notes.csv* contains about 40,000 patient note records, and *featues.csv* stores features for each condition summarised by professional physicians. The *train.csv* contains 1,000 patient notes and their corresponding symptoms, 100 for each of 10 standard patient cases. The *train.csv* is an available training set, but we may pre-process the huge remaining data to convert them into the trainable format, while part of the *train.csv* will be used for the validation.

Before implementation, the training set and validation set are split according to standard patient cases. Of the 100 patient notes of each case, 80% (80 notes) were used for training, while the other 20% (20 notes) were used for validation, forming a training set of 800 notes and a validation set of 200 notes. Note that due to the limiting data volume, we did not split an extra test set for testing. Hence, the validation set was validated at the end of each epoch to provide information about the training quality (for model selection and to avoid overfitting) and did not use for model development.

With respect to the number of features, there are 143 entities corresponding to standardised expressions of feature symptoms in total. The label volume of our project is larger than most of the other NER tasks. Therefore, in our real implementation, two tagging schemes are examined, namely Inside-outside-beginning (IOB) tagging and non-IOB tagging. For the IOB tagging, each entity will be tagged as B-entity, which indicates the beginning token and I-entity, which means inside token, and the total label volume under this tagging scheme is 287, containing the beginning and inside tag for each of the 143 features and 1 for outside token 'O'. As for non-IOB tagging, only the outside token 'O' is preserved while not specifying the beginning token and inside token, leaving each feature to correspond to one tag. Hence, the label volume of non-IOB tagging is reduced to 144.

4. Methodology

In this section, we first introduce and explain the components we use throughout the experiment and then describe the way to combine these components to form reasonable network architectures. Also, essential parameters of the network will be specified and explained in detail.

4.1. Modules as Component

Bio ClinicalBERT. [1] The bidirectional encoder representations from transformer (BERT) has a deep bidirectional network structure and can be pre-trained to perform contextual embedding by jointly conditioning on both left and right context in all layers (Devlin et al., 2018). Despite this, BERT also faces the limitation of the corpus. In consideration that our project is aiming at tackling medical NER problem, which might have quite different vocabulary compared to normal NER tasks because of speciality corpora is included, we chose to use the *Bio ClinicalBERT* that is specially pre-tained with clinical texts (Alsentzer et al., 2019) for a better contextual embedding performance. Note that we did not further fine-tune the BERT model with dataset and instead used BERT for embedding only.

LSTM. The LSTM layer is implemented with the provided library from PyTorch. In the experiment, the performance of both uni-directional and bi-directional LSTM will be examined. Meanwhile, the number of layers of LSTM is also considered and tested in the experiment to provide insight for model fine-tuning.

CRF. [2] The conditional random field (CRF) is a finite state model with unnormalised transition probabilities that enables the model to take contexts into consideration during the classification instead of only associating a label with a single feature (Lafferty et al., 2001). In this project, a linear-chain CRF is used for loss calculation and prediction emission. As we know that the hidden markov model (HMM) features learning a markov process with a hidden state by observing another observable process (Rabiner & Juang, 1986), with joint probabilities p that:

$$p(\boldsymbol{y},\boldsymbol{x}) = \frac{1}{Z}\prod_{t=1}^{T}\exp\left\{\Sigma_{k=1}^{K}\theta_k f_k(y_t, y_{t-1}, x_t)\right\} \tag{1}$$

where θ denotes parameter vector, Z is a normalisation constant and each feature function is represented as $f_k(y_t, y_{t-1}, x_t)$. As CRF is calculating conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ based on the p($\boldsymbol{y}$,$\boldsymbol{x}$) of HMM, the equation of CRF can be written as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\prod_{t=1}^{T}\exp\left\{\Sigma_{k=1}^{K}\theta_k f_k(y_t, t_{t-1}, \boldsymbol{x}_t)\right\} \tag{2}$$

$$Z(\boldsymbol{x}) = \Sigma_{\boldsymbol{y}}\prod_{t=1}^{T}\exp\left\{\Sigma_{k=1}^{K}\theta_k f_k(y_t, t_{t-1}, \boldsymbol{x}_t)\right\} \tag{3}$$

where $\{f_k(y, y', \boldsymbol{x}_t)\}_{k=1}^{K}$ is a set of real-valued feature functions. For an HMM, state transition from i to j shares the same score, while the CRF is able to take the input context into consideration depending on the current observation vector by adding a feature that consists of the concatenated identical matrix $\{y_t = j\}$, $\{y_{t-1} = 1\}$ and $\{x_t = o\}$. Therefore, for a NER task that processes sequences, CRF can be used for better results as it features considering the context in sequence.

Remark1. The equations of HMM and CRF are borrowed from an introduction written by (Sutton & McCallum, 2010) and provided to give a basic idea of the reason why we used it in our network.

4.2. Baseline Model

To firstly obtain a general idea about NLP model's performance on this task, a baseline model was built with a simple BERT embedding following a fully connected layer and a dropout layer. The fully connected layer functions to lower the embedding dimension of BERT from 768 to match our label size of 144 for non-IOB tagging and 287 for IOB tagging, while the dropout layer is set with a dropout probability of 0.3, intending to avoid overfitting.

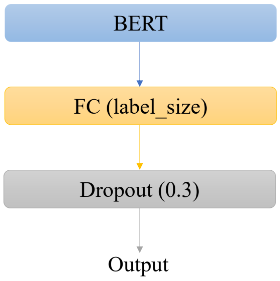

Figure 1. Model structure of the simple baseline

4.3. Self Proposed Model

We proposed a network structure with CNN module based on the idea of a few-shot learning CNN-BiLSTM for medical text NER (Hofer et al., 2018). Similarly, the structure that adds a CNN before LSTM is proposed by (Chiu & Nichols, 2016). The kernel size from Hofer's experiment is set to be 3, while Chiu's is set to be 4. However, the kernel size of 4 might cover most part of the tokens, which would result in poor feature extraction. Thus, in our case, the kernel size of 4 has not been chosen. Also, different from Hofer, who introduced only one convolution layer, we added two separate 1d convolution layers (with kernel size 2 and 3) for the character level feature extraction. Meanwhile, Hofer used a dropout layer following the character embedding and a fully connected layer following the concatenated features, which we think might cause the loss of important feature information. Therefore, the specific

[1] The Bio_ClinicalBERT pre-trained model is directly obtained from HuggingFace transformer library https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT.

[2] In this project, the CRF is borrowed from kmkurn's pytorch-crf implementation (https://github.com/kmkurn/pytorch-crf), which is mostly developed based on the AllenNLP CRF module (https://github.com/allenai/allennlp).

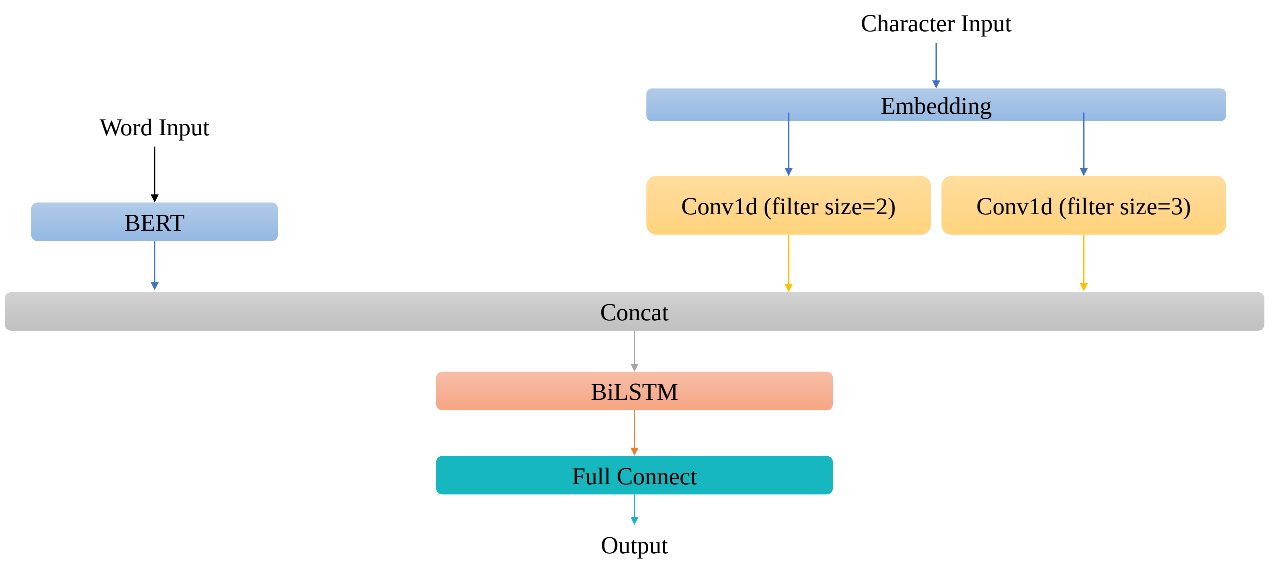

Figure 2. Model structure of our BERT-CNN-BiLSTM

dropout layer and fully connected layer are removed from our model. By doing these modifications, our network uses BERT for word level embedding and a self-implemented simple embedding for character level. Then, the features from character embedding are extracted with two CNNs and the out channel size is 77, which enables the channel of CNNs can be concatenated to be 144. Subsequently, a further concatenation is done with the BERT's embedding, which has a dimension of 768, forming the total input dimension of 912 for the BiLSTM. The final fully connected layer will compress the dimension to 144, which fits our label size for the following prediction.

4.4. Data Preprocessing

The given raw data were filtered with regular expressions to remove special characters. The regularised data were then organised into data frame with three keys: *patient_note_id*, *entity_label*, and *token*. Note that the initial tokens were simply split with space. For the setting BERT, as the input length should be consistent in the model, each patient note was truncated or pad to the max length of 250 tokens. This number was chosen after examining the length of all patient notes, and the possibility of an increased token number due to word piece tokenisation has also been taken into consideration. We intended to keep most of the patient notes complete to prevent the loss of critical information (some labels are annotated at the end of the note). Simultaneously, word to index was done directly with the tokenisation of pre-trained *Bio ClinicalBERT* model, while tag to index was given by ourselves with index ranging from 0 to 143 that each indicates one entity label. Because truncation and padding were applied, the mask was used to determine the valid sequence, allowing us not to labelling start tag, end tag and padding, as they will be ignored by PyTorch automatically. To tackle the problem caused by word piece tokenisation that a work can be split into multiple tokens, we follow the rule that subtokens (start with ## after tokenisation) were labelled as their former chief token (*e.g.*for 'sun', '##ny', the label of '##ny' would be identical to 'sun').

4.5. Experiment Settings

Two models, namely BERT-LSTM and CNN-BiLSTM by (Hofer et al., 2018), are reproduced to compare with the baseline and our self-proposed model. The BERT is only used for contextual embedding in both models, and no further fine-tuning on BERT is going to be implemented. Meanwhile, an ablation study of CRF is performed on BERT-LSTM in order to examine the improvement contributed by the optimisation of CRF. Lastly, as the Bidirectional LSTM (BiLSTM) is used in our model, several other counterparts with BiLSTM are also reproduced to compare with our model.

Except for the general model structure, another parameter that may affect model performance is also tested on the model structure that receives the best result. Because the label volume in our task is larger than other traditional NER tasks, we hypothesise that higher model performance is achievable by increasing the layer number of LSTM. As LSTM may generally work well for up to 4 stacked layers (Wu et al., 2016), the adjustment of layer number in our experiment ranges from 1 to 3.

In terms of some other experiment settings, for loss computation, the cross-entropy loss is adopted for models without CRF module, while CRF will directly provide loss and prediction for models with CRF module. Also, the large batch size is witnessed resulting in poor performance in our preliminary experiments, which might be due to the limiting amount of the training data and the large label volume, and meanwhile causing an extremely time-consuming scenario when added CRF module, the batch size is set to be 1 in our main experiments, both for a better training outcome and keep consistent with the CRF module to reduce training time cost. Furthermore, the initial learning rate of all experiments is 0.001, while continuous parameter update and optimisation were done using AdamW optimiser, which features decoupled weight decay (Loshchilov & Hutter, 2019) and the cosine annealing learning rate strategy with warm restart based on previous parameters before the restart (Loshchilov & Hutter, 2016) was chosen for updating the learning rate. This selection of optimiser and scheduler aims at achieving a relatively faster convergence but avoids local minima at the same time to ensure the generalisation capacity of the model.

In evaluation, two metrics were used to describe model performance, namely NER accuracy and NER f1-score. The traditional overall accuracy is simply computing the percentage of correctly predicted tokens. However, we know that there is a large amount of outside token ’O’, which will result in high overall accuracy. Therefore, we compute the NER accuracy instead, which only takes meaningful tokens from the label into account, and draw the corresponding percentage from the prediction. Plus, a NER f1-score, which is a common metric in NER tasks (Hofer et al., 2018; Cho et al., 2020), that excluded the influence of outside token ’O’ is also used as a metric for model comparison.

5. Experiment Results

In this section, the convergence performance is revealed and discussed to analyse the capacity of each model. Subsequently, the detailed model performance will be given, and further model comparison will be made in the later parts. Also, to provide a better understanding of the prediction output, several prediction examples will be visualised in the last part of this section.

5.1. Convergence Performance

Models trained with IOB tagging are tested firstly (Figure 3). The baseline model shows an apparent slow convergence and difficulty in obtaining a good NER accuracy, resulting in an NER Acc. of around 30%. Conversely, after adding LSTM module to the model, the BERT-LSTM no-CRF model has better convergence performance that almost reached the maximum accuracy at approximately the 60th epoch. However, CRF did not provide obvious improvement compared to BERT-LSTM no-CRF that the trend and the max value of their NER accuracy are similar, which is out of our expectation.

5.2. Model Comparison

5.2.1. Models with IOB tagging

The detailed performances of the IOB tagging models are shown in Table 2. It can be seen that an LSTM module can bring considerable performance to the model. Despite, as mentioned in section 5.1, the improvement of adding CRF module is trivial.

Model	Acc.	f1
Baseline	0.3476	0.3396
BERT-LSTM	0.6686	0.6738
BERT-LSTM(CRF)	**0.6692**	**0.6736**

Table 1. Performance of Baseline, BERT-LSTM and BERT-LSTM(CRF) trained with IOB tagging.

5.2.2. Models with non-IOB tagging

Under the scenario of non-IOB tagging, the BERT baseline model obtained an even worse trend of convergence. However, an improvement is witnessed in the BERT-LSTM no-CRF model, with an even faster convergence and reaching 0.7198 on NER f1-score (Table 2).

Model	Acc.	f1
Baseline	0.2132	0.2407
BERT-LSTM	**0.6659**	**0.7198**

Table 2. Performance of baseline and BERT-LSTM trained with non-IOB tagging.

5.2.3. Parameter Comparison

By varying the number of the layer in LSTM, three experiments were done (Table 3). There are no apparent differences between the performance of these three models. The BERT-LSTM with the number of the layer set to 1 performs slightly better than others. Therefore, the number of layers of 1 is chosen to develop our self-proposed model.

num_layer	Acc.	f1
1	**0.6659**	**0.7198**
2	0.6592	0.6981
3	0.6911	0.6956

Table 3. Comparison of model performance by varying the number of layer in LSTM. All three models are trained with BERT-LSTM no-CRF.

5.2.4. Overall Comparison

For the final comparison, the overall results are shown in Table 4. By comparing the performance of BERT-LSTM and BERT-BiLSTM, the result reveals that enabling bi-direction can provide considerable performance, increasing the NER f1-score from 0.7198 to 0.7496. Hofer’s CNN-BiLSTM architecture provides a further improvement, which proved the efficacy of applying a 1d convolution on character level embedding. Our self-proposed BERT-CNN-BiLSTM outperforms all other models compared, obtaining an NER f1-score of 0.8641 and an NER accuracy of 0.8366, which indicates our model structure of two 1d convolution layers and removing dropout and fully connected layer after feature concatenation has demonstrated novelty in comparison to the CNN-BiLSTM that Hofer implemented.

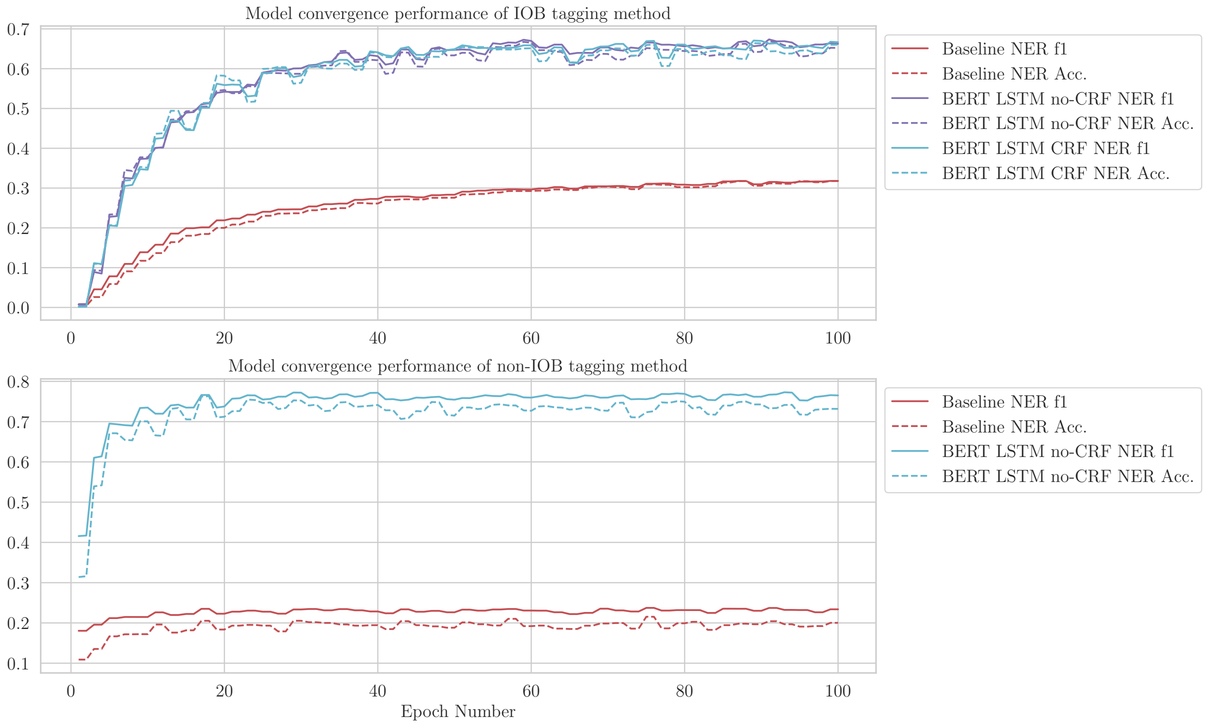

Figure 3. Model convergence performance of IOB and non-IOB tagging method. For IOB tagging, three models are examined: the BERT no-CRF baseline, BERT-LSTM no-CRF and BERT-LSTM with CRF. For non-IOB tagging, the the BERT no-CRF baseline and BERT-LSTM no-CRF are compared.

Model	Acc.	f1
Baseline	0.2132	0.2407
BERT-LSTM	0.6659	0.7198
BERT-BiLSTM	0.7145	0.7496
CNN-BiLSTM	0.7749	0.7896
BERT-CNN-BiLSTM	**0.8366**	**0.8641**

Table 4. Performance of all models trained with non-IOB tagging. Number of layers in LSTM and BiLSTM is set to 1.

Remark2. The CNN-BiLSTM was reproduced strictly according to the structure described by Hofer (Hofer et al., 2018). In our implementation, we found opposite result that adding a fully connected layer after the concatenation of features before feeding into the final BiLSTM affects the convergence performance of the model.

5.3. Prediction Example

In this part, several prediction examples are given, for a better understanding of our task, and some analysis are done to explain the results. Note that because of the length limitation of the column, only part of the predicted output is chosen to visualise.

A completely correct prediction is shown below in example 1. The ’O’ indicates outside token, which means not related to any standard expression of feature symptoms, while the ’409’ and ’407’ are two labels, representing the standard expression of ’45-year’ and ’Female’.

```
Example 1:
Prediction:
['O', 'O', '409', '409', '409', '407', 'O']
Label:
['O', 'O', '409', '409', '409', '407', 'O']
```

However, in some cases, the prediction is not completely correct. For instance, the case below (example 2) shows the model partially recognises the label ’403’, which indicates ’Heavy-caffeine-use’. This may be due to stop words are also included in the label, which possibly causes confusion in the model of whether should a stop word be recognised as the no meaning(’O’) or as a feature symptom (*e.g.*’403’).

```
Example 2:
Prediction:
['403', '403', 'O', 'O', 'O', 'O']
Label:
['403', '403', '403', '403', '403', '403']
```

Another very similar prediction as example 2 is given in example 3, where ’406’ annotates the standard expression of ’Insomnia’. By checking the original dataset, there are several ways to express this feature. For example, if no stop word is contained, it can be expressed as ’trouble sleeping’, ’decreased sleep’, and ’difficulty falling asleep’. On the other hand, the stop word is contained in cases like ’trouble going **to** sleep’ and ’difficulty **to** sleep’. In the specific case of example 3, one of the ’403’ features is predicted as ’O’, and it is highly possible that labelling the stop word as a meaningful feature is going to account for the incorrect prediction.

```
Example 3:
Prediction:
['406', 'O', '406', '406', '406', 'O']
Label:
['406', '406', '406', '406', '406', 'O']
```

6. Discussion & Conclusion

Limitation. However, our model structure still has limitations. Firstly, an ablation study has not been done on a single convolution layer of kernel size 2 and 3. Thereby the effect of these two layers is not clear enough. Adding one set of ablation studies would make the experiment more rigorous. Secondly, the CRF is thought not quite successfully applied to the model. Due to the training with CRF being time and resources consuming, and we have only limited computational power, no more experiments are done on models with CRF. By studying other related works, the CRF module should provide a considerable improvement on the model. Moreover, with respect to the dataset and annotation method, the label provided is not consistent and accurate enough. For example, ’17y/o’ may be annotated as ’describing age’ in some patient notes but not annotated in some others, which may cause the validation process to ignore some correctly recognised expressions. Meanwhile, the annotation is given as a continuous phrase. For instance, the whole phrase ’pressure on her chest’ is labelled as one feature, which causes us to give the same entity for all of these four words. However, it is apparent that the preposition ’on’ should not be given an entity of one type of the standard expression feature symptom. To tackle this problem, stop words like articles, prepositions, and conjunctions can be removed from the dataset before training.

Future Work. The model still has the potential to be further improved. A more satisfactory result is achievable by further implementing CRF after the BiLSTM module and using CRF for loss calculation and prediction emission. Apart from this, instead of only using CNN for character level feature extraction, the sequence of character features also matters. Therefore, an LSTM module can also be added to extract extra sequence features at the corresponding character level (Yadav & Bethard, 2019; Luo et al., 2019). By concatenating features from BERT, CNN and LSTM, more features can be fed into the word level BiLSTM, thereby improving the performance. Furthermore, instead of considering this as an NER task, it can also be regarded as a translation task that translates a patient note into the standard expression of feature symptoms, which implies the real sentence can be concatenated with the corresponding label phrase in the embedding phase, as a new idea to tackle this problem. Moreover, in terms of the dataset, more analysis and visualisation similar to section 5.3 can be done on it to optimise the annotating method and bring insight for adjustment from the perspective of the label and the pre-processing of data. As mentioned in the limitation part, removing the stop words during data pre-processing might be able to solve the incorrect prediction we found in our prediction examples.

Contribution & Conclusion. Looking forward, our experiments demonstrated the performance of models with different component modules on a special clinical NER task with 144 labels in non-IOB tagging and 287 labels in IOB tagging, and proposed an advanced BERT-CNN-BiLSTM network that achieved a competent result on mapping phrases to the specific standard expression of feature symptoms in patient notes. Our model outperforms all other model reproductions mentioned with an NER f1-score reaching 0.8641 and an NER accuracy counterpart of 0.8366. Referring to the overlays presented in this project, the great potential is revealed for NLP methods to be further developed for a more precise and efficient auto-scoring system for medical students’ patient note writing examination. By contributing more efforts based on this, the training of medical students can be more standardised, and human bias in scoring can be avoided. In addition, the significant increment in scoring efficiency and the lower labour cost is achievable.

References

Alsentzer, Emily, Murphy, John R., Boag, Willie, Weng, Wei-Hung, Jin, Di, Naumann, Tristan, and McDermott, Matthew B. A. Publicly available clinical bert embeddings, 2019.

Chiu, Jason PC and Nichols, Eric. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370, 2016.

Cho, Minsoo, Ha, Jihwan, Park, Chihyun, and Park, Sanghyun. Combinatorial feature embedding based on cnn and lstm for biomedical named entity recognition. *Journal of biomedical informatics*, 103:103381, 2020.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

Dzau, Victor J and Ginsburg, Geoffrey S. Realizing the full potential of precision medicine in health and health care. *Jama*, 316(16):1659–1660, 2016.

Hofer, Maximilian, Kormilitzin, Andrey, Goldberg, Paul, and Nevado-Holgado, Alejo. Few-shot learning for named entity recognition in medical text. *arXiv preprint arXiv:1811.05468*, 2018.

Kuru, Onur, Can, Ozan Arkan, and Yuret, Deniz. Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 911–921, 2016.

Lafferty, John D., McCallum, Andrew, and Pereira, Fernando C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Loshchilov, Ilya and Hutter, Frank. Sgdr: Stochastic gradient descent with warm restarts, 2016.

Loshchilov, Ilya and Hutter, Frank. Decoupled weight decay regularization, 2019.

Luo, Ying, Xiao, Fengshun, and Zhao, Hai. Hierarchical contextualized representation for named entity recognition, 2019.

Marafino, Ben J, Park, Miran, Davies, Jason M, Thombley, Robert, Luft, Harold S, Sing, David C, Kazi, Dhruv S, DeJong, Colette, Boscardin, W John, Dean, Mitzi L, et al. Validation of prediction models for critical care outcomes using natural language processing of electronic health record data. *JAMA network open*, 1(8):e185097–e185097, 2018.

Nadeau, David and Sekine, Satoshi. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

Papadakis, Maxine A. The step 2 clinical-skills examination. *New England Journal of Medicine*, 350(17): 1703–1705, 2004. doi: 10.1056/NEJMp038246. PMID: 15102993.

Rabiner, L. and Juang, B. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. doi: 10.1109/MASSP.1986.1165342.

Salt, Jessica, Harik, Polina, and Barone, Michael A. Leveraging natural language processing: Toward computer-assisted scoring of patient notes in the usmle step 2 clinical skills exam. *New England Journal of Medicine*, 94(3): 314–316, 2019. doi: 10.1097/ACM.0000000000002558.

Şeker, Gökhan Akın and Eryiğit, Gülşen. Initial explorations on using crfs for turkish named entity recognition. In *Proceedings of COLING 2012*, pp. 2459–2474, 2012.

Sutton, Charles and McCallum, Andrew. An introduction to conditional random fields, 2010.

Tissot, Hegler C, Shah, Anoop D, Brealey, David, Harris, Steve, Agbakoba, Ruth, Folarin, Amos, Romao, Luis, Roguski, Lukasz, Dobson, Richard, and Asselbergs, Folkert W. Natural language processing for mimicking clinical trial recruitment in critical care: a semi-automated simulation based on the leopards trial. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2950–2959, 2020.

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V., Norouzi, Mohammad, Macherey, Wolfgang, et al. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

Yadav, Vikas and Bethard, Steven. A survey on recent advances in named entity recognition from deep learning models, 2019.