Named Entity Recognition on Medical Data

Swapnil Gupta
M.Tech, IISC Bangalore
swapnilgupta.229@gmail.com

1 Introduction

The task of Named Entity Recognition(NER) can be described as the identification of named entities in computer readable text and annotating it with a suitable pre-defined categorization tag to it. It is an important subtask for information extraction for natural language documents. It also finds a cruicial application in reference resolution, other types of disambiguation, and meaning representation in other natural language processing applications. The problem finds a special place in language community because it comes at the intersection of several natural language tasks like POS tagging, semantic parsing, thematic meaning representations and can form an important part of several applications like question-answer systems, textual entailment etc.

There are two sub-parts to this problem

- Detection of named entities.
- Annotating them with correct NER tag

NER is widely used in the Medical Community for effective medical text analysis where the named entities generally belong to the categories like Drugs, Diseases, treatments etc. Its applications include getting information on symptoms of newly evolving diseases, identifying the side-affects of the existing drugs and to get feedback on different kinds of treatment.

In the vast research literature on the topic following are some of the key challenges identified related to this field

- Chunking and Text representations.
- Term ambiguity and term variability.
- Modelling of non-local dependencies.
- Incorporating External Knowledge Resources.

The aim of this assignment is two-folds. First to do an analysis of the impact of different kinds of features on a Conditional Random Field (CRF) based model in NER. Second to compare it with a recurrent neural network (RNN) based deep sequence tagging model. The report is organized as follows. In section(2) implementation details of both the models is discussed. Then in Section(3) a description of all the features used for CRF model is given. In section(4) experimental results are presented and finally section(5) contains observations followed by conclusion in section(6).

2 Model Design

The input of our model is a text corpus and we need to output a tag corresponding to each token of the corpus. Below are two models which have been shown to be effective in the above settings in varied applications.

2.1 Conditional Random Fields

Conditional Random Fields are class of statistical modelling methods used for structural predictions. They are a type of discriminative undirected probabilistic graphical models. Compared to other graphical models which belongs to sequence modelling family like HMM's, CRF's are way more flexible. Unlike HMM's here corresponding to each token in the corpus we can give several different kinds of features as well as can effectively incorporate both past and future context words for each token. Due to the above reason CRF's are considered to be state-of-art in structural predictions.

There are various types of CRFs which are used for sequence modelling. First, the linear chain CRF in which the prediction for a token only takes the dependency from the label for the previous token in the sequence and the can include a rich set of features for the current word.

Second, Dynamic conditional random elds are sequence models which allow multiple labels at each time step, rather than single labels as in linear-chain CRFs.

For this task a Linear Chain CRF model is being used and the tool utilized to build such a model is **sklearn-crfsuite**. This is a python tool which helps to built a Linear Chain Crf . The algorithm used for parameter estimation of this model is **LBFGS** with **elastic regularization**.
Elastic regularization is combination of L1 and L2 regularization. There are two hyper parameters **c1 and c2** which are the coefficients of L1 and L2 regularization terms respectively in the loss function.

2.2 Bi-GRU based Transducer model

In recent years RNN based models have been very successful in sequence modelling problems. Since in NER task we have to assign a label to each token of a sentence, hence an RNN Transducer model becomes a natural choice for the task. A schematic structure of a tranducer RNN model is shown in the figure(1) below.

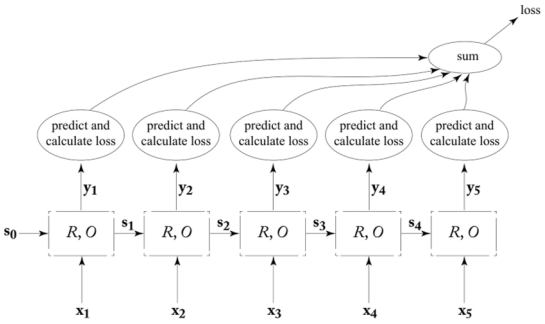

Fig 1:Image taken from the book Neural Network Methods for Language Processing by Yoav Goldberg.

Capturing both past and future context to assign tags to each token has been very successful in the task of NER. Hence for our deep model we are using Bi-directional GRU cells. GRU has been shown to be very effective in capturing long range dependencies. Word embeddings for each token are also treated as trainable weights using an embedding layer. Word2vec model is used to initialize the embeddings. Adam optimizer with cross entropy loss is used for the Back Propagation through time algorithm.

For this model there are two important hyperparameters which needs to be tuned

- No. of hidden states for the GRU cell
- length of the embeddings for each token.

3 Features Analysis for CRF model

The identification of proper templates of features and selecting the most important features among them plays very significant role in doing Named Entity Recognition using models like CRF. As the first task of this assignment a variety of different features are designed and their analysis is done. Below is the description of the features which are helpful for the task.

- **Word context:** The importance of context is ubiquitous in natural language tasks. Hence for this task for each token a 3-word context window keeping the current token at the center is used. Under this for the label prediction of the current token all the token specific features, discussed in subsequent points, of the current token as well as that of its immediate left and right tokens are used. Bigger context windows were also tried but didn't show any improvement in results.

- **Prefix and Suffix:** The words prefix and suffix can provide significant information about the word's label as various diseases share common suffix or prefix. For instance, the word *'tis'* is part of several diseases like *'hepatitis'* and *'meningitis'*. Hence the last and first three characters of each token are used for extracting this info.

- **Capitalization and Digit information:** The diseases generally start with capital letters or include some/ all letters capital(eg: hepatitis B, OEIS-complex) .But in this dataset it was observed that this type of structure was not followed much rigorously so this feature might not be so useful. Various diseases come with name including digits such as 'H1N1' so this feature is also selected. Both of them are Boolean features.

- **POS-tags:** Parts of speech information plays a crucial role in first task of chunking or actually segmenting named entities from the text corpus. An off the shelf POS tagging tool from nltk library is used for the purpose.

- **Cluster Id using Word2vec:** Word2Vec has been shown to be very effective in capturing similarities between different words in a corpus. Hence, we tried to explore if tokens having same label will have similar Word2Vec representations. Hence using Word2Vec representations we clustered all the tokens in 3 clusters using K-Means and after that every token was assigned a cluster Id(0-2). We will see in the results section that this feature is actually helpful in increasing the F1-score of the disease and treatment categories.

- **No-of-contexts using WordNet:** WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. So for every token, the no of synsets corresponding to the token are calculated which provides the number of contexts in which the corresponding token can be used. It was observed in the data set that the disease-tokens and treatment-tokens were having only one or two synsets while other-tokens were having several synsets.
  Eg: "angioplasty",label -T, No of synsetes: 1
  "reduction" ,label -O, No of synsets: 3

- **Word-length:** The word length also provides some information about the word as the average length of diseases are generally more than common words. But this feature doesn't seems to help in differentiating treatments from diseases. Not much impact was seen on model's performance after adding this feature.

4 Experiments and Results

4.1 Dataset

The data provided is in the form of tokenized sentences and each word is labelled as one of the three things

- **D**: Disease
- **T**: Treatment
- **O**: Other

There are a total of 3655 sentences. Which are divided into 70:10:20 portions for training, development and testing respectively.

4.2 Evaluation Metric

In the given dataset the data is very skewed towards 'O' (other) class. Under such settings Accuracy measure can be be misleading and hence precision, recall based metric is used for the analysis.

Precision:

$$\frac{True\ positive}{True\ positive + False\ positive}$$

Recall:

$$\frac{True\ positive}{True\ positive + False\ negative}$$

F1-Score:

$$\frac{2 * Precsion * Recall}{Precsion + Recall}$$

The F1-score for every label is calculated individually. Since, the no. of tokens for the class is 'O' in our training data is very large hence we have taken we have reported an unweighted average of the F1-score as otherwise the F1-score of 'O' will have a dominating effect.

4.3 Conditional Random Fields

CRF is a very versatile model because it can accommodate any no. of and any type of features corresponding to each token. Performance of CRF is hence highly correlated with the quality of features that are fed to the model. The impact of some of the selected features is presented below by incrementally adding them to our base feature set referred to as Set I.

Set I: Context window, prefix-suffix, Digit information, word length **F1 Score: 0.72**

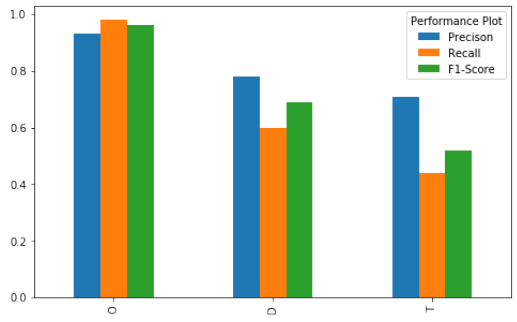

Figure 3

Set II: Set I, POS Tag, Capitalization **F1 Score: 0.77**

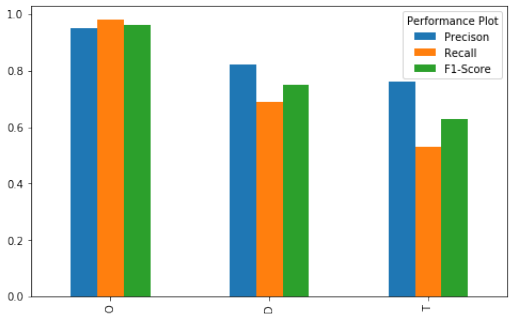

Figure 4

Set III: Set II, Cluster Id, No. of contexts **F1 Score: 0.79**

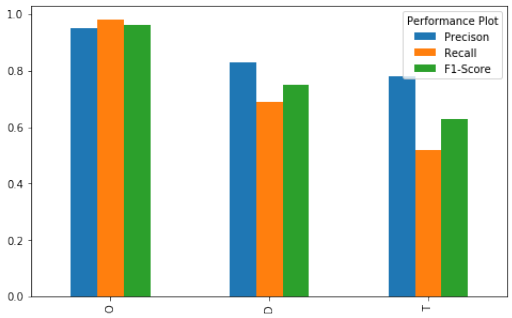

Figure 5

The below table summarizes the results of the model using all the features on the test set.

	Precision	**Recall**	**F1-Score**
O	0.95	0.98	0.96
D	0.83	0.69	0.75
T	0.78	0.52	0.66
Average	0.85	0.73	**0.79**

T1: Table to show Classification Report on Test data for CRF

4.4 Bi-GRU based Tranducer Model

Unlike CRF's, feature engineering is not a part of deep models and only an embedding vector which is tuned for the task is fed to the model corresponding to each token. Due to computational power limitation the size of embedding for each token is kept fixed and the effect of changing no. of hidden units is studied and the results are shown in the figures below. For both the plots the model was run for 30 iterations.

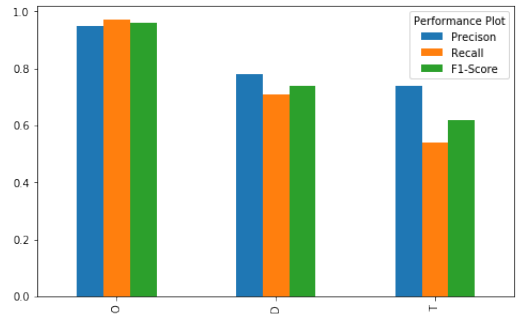

Figure 6:No. of hidden units in each cell = 100. F1-score 0.77.

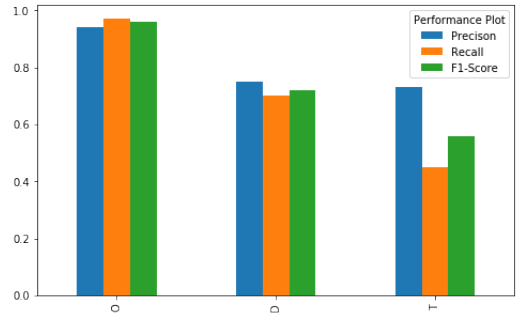

Figure 7:No. of hidden units in each cell = 50. F1-score 0.74.

The below table summarizes the results on the test data for the best model

	Precision	**Recall**	**F1-Score**
O	0.95	0.97	0.96
D	0.78	0.71	0.74
T	0.74	0.54	0.62
Average	0.82	0.75	**0.77**

T2: Table to show Classification Report on Test data on Bi-GRU model

5 Observations and Discussions

The above results clearly depict the difficulties involved in the task of NER. Both our models been quite accurate for label 'O', but have clearly struggled for the Named Entity labels 'D' and 'T' both for the task of chunking and assigning appropriate tags. Doing a comparative analysis, CRF is edging out the Bi-GRU model slightly in the F1-score in the above experiments. But it is worth noting that while CRF is making use of variety a variety features our Bi-GRU model is only taking the embed-

ding vectors. Moreover, since in Bi-GRU models we have to train a lot of parameters for effective training a lot of training samples are required. And hence we can expect that on a larger dataset Bi-GRU model would be more effective. An interesting point is that CRF gives us the opportunity to use domain specific features while such things can't be incorporated directly in deep models.

The performance of our model with respect to tag 'O' seems almost independent of any feature set or hyper-parameter tuning. But for the other labels 'D' and 'T' which are more crucial for our application the increased no. of features and tuned hyper-parameters show a significant positive impact on the F1-scores. Which is encouraging to see.

6 Conclusion

This assignment gave us an opportunity to work with structural prediction models for sequence tagging. CRF is a very versatile model but requires hand crafted features. This can be considered as a limitation as well as strength of the formulation as it allows us to incorporate domain specific features. On the other hand, deep sequence tagging problems learns task specific representation. But it is limited by the amount of labelled data we have which in such sequence labelling task can be expensive to create.

.

7 Github Link

All the code files can be found at the below link:
https://github.com/229Swapnil/NLU_NER_Assignment3