Natural Language Understanding
Assignment-3
NER System for Diseases and Treatments

Karan Malhotra
M.Tech (Systems Engineering)
kmkaran212@gmail.com

1 Introduction

Named Entity recognition is a process in which an algorithm takes a string of text as input and annotate it with suitable pre-defined categories(such as people,place,organization etc.) This approach is used in information extraction very widely as it can automatically scan various articles and reveal which are the major people, organizations, and places discussed in them. It find various applications in domains like knowledge extraction,machine translation and speech recognition.

There are two sub-parts to this problem

- To detect the named entities .
- To annotate entities with correct NER tag.

NER is very important element in medical text mining and text analysis.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. This task is generally more challenging than common names recognition due to ambiguous naming conventions and long length strings of medical entities.

The given task is being solved using two approaches . First one is using the Conditional Random field(CRF) which is a graph based discriminative model. Several sets of features were tried for this model and the top performing ones were selected. Second is a Recurrent neural network based deep sequence tagging model.
This report is being structured as follows-: Section-2 consists the implementation details of both the models. Section-3 includes feature analysis for CRF model.In Section-4 the evaluation results are presented and the section-5 consists of observation and conclusions.

2 Model Details

Both the models take a text corpus as input and provides a label(O,D,T) to every token present in the corpus. The two models and their implementation details are explained below.

2.1 Conditional Random Fields(CRF)

A conditional random field is simply a conditional distribution with an associated graphical structure.These models fall into the sequence modeling family which takes context into account and predict sequence of labels for an input sequence.Mainly CRF finds application in POS tagging, Named entity recognition,Shallow parsing,gene finding and an alternative to Hidden Markov Model(Hmm) in some applications. Unlike HMM's here corresponding to each token in input sequence we can give different kind of features as well as effectively incorporate both past and future contexts, therefore CRFs are considered to be the State of the art for structure 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.
Third, relational Markov networks are a type of general CRF in which the graphical structure and parameter tying are determined by an SQL-like syntax. Finally, Markov logic networks are a type of probabilistic logic in which there are parameters for each rst-order rule in a knowledge base.

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. The following figure shows a simple linear chain CRF model.

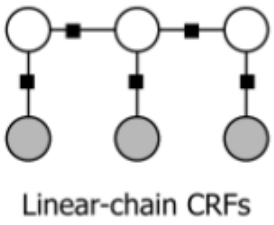

Fig 1:Simple Linear Chain Model
***An Introduction to Conditional Random Fields for Relational Learning(Charles Sutton)**

Sklearn-crfsuite supports a dictionary type feature format, so for every token in training data it is represented by a dictionary including several features like - postag ,word context,cluster Id etc.

2.2 RNN based deep sequence tagging model

RNN based models are also significantly used in case of sequence modeling problems as they are able to capture the context properly. As the NER requires capturing the context from both the directions (left and right) therefore a bidirectional gated recurrent unit is selected. The gated unit provides the solution to the vanishing gradient problem. For this model there are two important hyperparameters which needs to be tuned:
1. Size of state vector of Bi-GRU.
2. Dimension 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 .
So , the model was evaluated for different sets of features and results were analyzed to select the top most among them.The following are the features on which experiments were performed:

- **Word context:** The Current word and its contexts are very useful in recognition tasks. A window of 3 words is being used to capture the context effect for every word.So the feature for a particular word consists of the combination of features of its both side adjacent word's and its own features.

- **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.The last and first three characters of the word were 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 Trisomy 13 so this feature is also selected.Both of them are Boolean features.

- **POS-tag:** Part of speech information also plays an important role in NER, and this can be the feature which differentiate between treatment and diseases. Pos-tagger from nltk library was used to assign tags to tokens in train dataset.

- **Cluster Id using Word2vec:** Word2vec was used to convert the given tokens into a embedding vectors. These embedding vectors were then clustered into three clusters using K-means clustering and after that every token was assigned a cluster Id(0-2). This feature was used on the basis of an assumption that a disease-token will have embedding feature similar to the other disease-tokens than compared to the treatment-tokens. This feature provided a significant improvement in model's performance(Results in upcoming sections).

- **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 correspond-

ing 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 more than two synsets.
Eg: ”angioplasty”,label -T, No of synsetes: 1
”reduction” ,label -O, No of synsets: 3

- **Word-lemma:** The lemma of the word can provide useful information about the root word from which the word is derived. This feature was not much significant for the dataset set provided as generally the diseases and treatments does not have common root words.

- **Word-length:** The word length also provides some information about the word as the average length of diseases are generally more than common words (which has label as O).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 Evaluation Results

As the data provided is unbalanced the model's true performance can't be evaluated using only accuracy measure. Therefore an evaluation measure called **F1-score** which is the weighted harmonic mean of **Precision and Recall** is used. Precision is the percentage of the correct annotations and recall is the percentage of the total NEs that are successfully annotated.

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 and a unweighted average is taken to calculate overall F1-score of all three labels.

4.1 Hyper Parameter tuning:

The data set given was divided into 80:20 train-test split and the 10 fold cross validation was used to search for best value of hyper parameters on the training data. The hyper parameters were c1 and c2 which corresponds to the coefficients of L1 and L2 regularization respectively in the CRF model. The Following table shows the variation of c1 and c2 with their effect with 10 fold cross validation on training data:

C1	C2	F1-Score(cross-Val)
0.1	0.1	0.76
0.05	0.1	0.76
0.1	0.05	0.77
0.02	**0.3**	**0.79**
1	1	0.73
1	10	0.74

T1:Table to show Cross-Validation results while tuning c1 and c2

For the BiGru model the hyperparameter was the number of units in the gru cell (state vector dimension).Various values of the vector were tried and the best results were achieved with a value of 80.

4.2 Performance comparison of Feature Sets

The features were incrementally added and the evaluation measures values were recorded. The average of F1-score corresponding to all three classes is also calculated for comparison.The training data was further splitted into a development set for this experiment.

1.Features:Word-context,Capitalization and digit info,prefix and suffix.
Overall F1-score=0.75

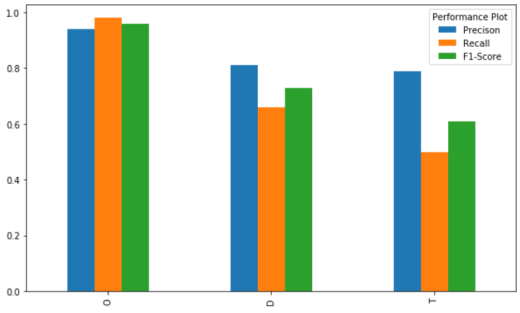

Fig 1:Classification Performance for feature Set 1

2.Features:Word-context,Capitalization,digit-info,prefix and suffix,postag.
Overall F1-score=0.77

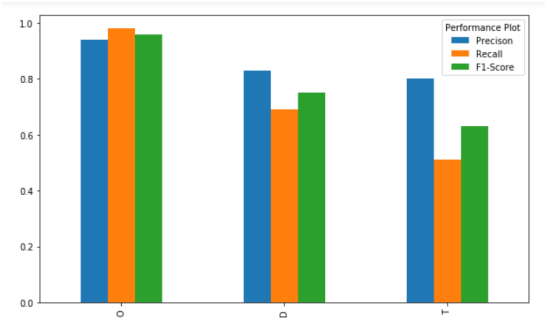

Fig 2:Classification Performance for feature Set 2

3.Features:Word-context,Capitalization,digit-info,prefix and suffix,postag,number-of-context,Word-lemma
Overall F1-score=0.76

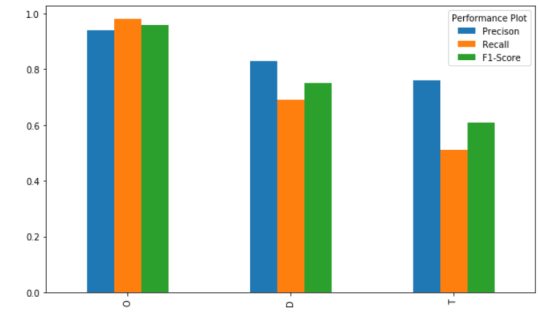

Fig 3:Classification Performance for feature Set 3

4.Features:Word-context,Capitalization,digit-info,prefix and suffix,postag,number-of-context, Cluster-Id,Word-len .
Overall F1-score=0.80

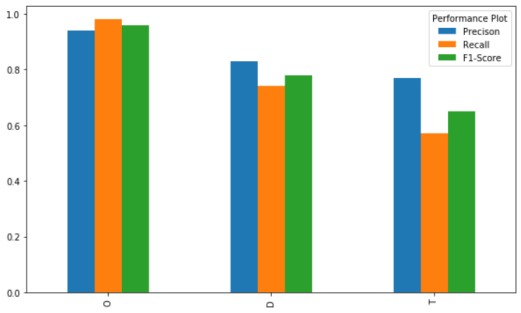

Fig 4:Classification Performance for feature Set 4

The analysis of above results helps to select the best features for the training of the CRF model.

After the hyper-parameter tuning and best feature selection process the CRF model was trained and the performance was evaluated on the test data. The Table below summarize the classification results on the test data.

	Precision	**Recall**	**F1-Score**
O	0.95	0.97	0.96
D	0.82	0.74	0.78
T	0.77	0.57	0.65
Average	0.85	0.76	**0.79**

T2: Table to show Classification Report on Test data(CRF)

The Bi-GRU deep sequence Tagger was trained on the same data set and evaluated on the same test data set as used for CRF model.The state dimension was the hyper parameter in this case and it was tuned on dev set with optimal value was coming to be **80**.The Table below shows the classification report of this model.

	Precision	**Recall**	**F1-Score**
O	0.94	0.97	0.96
D	0.77	0.70	0.73
T	0.74	0.51	0.60
Average	0.82	0.73	**0.77**

T3: Table to show Classification Report on Test data(Bi-GRU)

5 Observations and Conclusions

1. The Accuracy for all subsets of features in CRF model was coming around 92-93% , and variation could not be properly observed as class-O labels tokens were the maximum and were classified accurately for all feature settings.So measures like F-1 score,precision helped to see the variation in

the performance of CRF model with change in features.

2. It was observed that when the features like POStag and kmeans clustering ID were added then the performance of the CRF model was boosted a bit. Adding POStag feature changed F1-score from **0.73 to 0.77** which shows that giving a POStag helped the model to differentiate between diseases and treatment

3. When the feature were made using wordnet(no of synsets) and using K means clustering with help of Word2vec then the F1-score increased to **0.80** on the dev-set .This observation showed that the initial clustering of the tokens worked as a significant feature as treatments and diseases tokens were somewhat clustered in different clusters.

4. Features like Word lemma and word length does not seems to provide some increase in performance and were discarded while classifying labels for the test data..

5. The Bi-GRU model also performed well but was not enough to compete with CRF's performance.This observation supported the argument that CRFs are used in state of the art technologies for NER more than the RNNs and LSTMs as they can work with rich amount of features and contexts.

6. Increasing the number of units in GRU cell was not providing better results as the dataset was small and model was over-fitting . Dropouts were also not giving any significant performance increment.

6 Github Link

The code for these models are uploaded in the github page:
https://github.com/karanMal/NER_system