Assignment 3: Named Entity Recognition system for Diseases and Treatments

Suraj Panwar
MTech DESE
SR: 14644
surajpanwar@iisc.ac.in

1 Introduction

This report presents NER tagging algorithms for Diseases and Treatments, along with their performance comparison and discussion about the useful features to be extracted from the dataset along with a ablation study on different combinations of features with a inference on the optimal features to be extracted for good performance.

The models used here are:

1. MALLET based NER system.
2. LSTM based NER system.
3. SKLEARN CRF based NER system.

The models other than the MALLET based NER system are coded in python and their performance is compared to the performance of a standard package such as MALLET.

2 Separation of Datasets

The data separation is done as follows:

1. For Mallet the data is divided into 3 sets Training set, Test set and Validation set in the ratio of 70:20:10 for system implementation.
2. For LSTM based Neural sequence tagger system the data is divided into 80:20 ratio into Training and Test dataset for training and testing the model.
3. For SKLEARN based CRF system the 10-fold cross validation is preferred over the train test split as the process can be repeated multiple times and the results between different folds can be compared for model performance.

NOTE: Even though the model splitting is done in different manner in different systems the performance can be appropriately compared as the random seed is kept same in all the programs and hence the splitting is uniform in all cases and thus the performance of the models can be compared on a common ground.

2.1 Corpus and Basic Preprocessing

The Corpus given to us consists of 68175 entries of tokens with 11312 unique tokens which are presented along side with one of the unique tags, i.e. ”D”, ”O” and ”T”, where the tags stand for Disease, Others and Treatment the main objective of the assignment is to develop a reliable method to predict the appropriate tag given a sentence.

The sentence are separated by a next line, however as the Pandas module is not able to recognize the separation in the data, the data is modeled in such a way that the sentence endings are marked with a ”/n” token which is identified by the module and then is removed from the model as we go further along the model to maintain the sanctity of the sentence order. This is the only operation manually done on the dataset.

3 NER Systems implementation

The NER system is implemented using the following algorithms, this section provides a brief overview of the models and their workings.

3.1 MALLET

The Sequence Tagging toolkit is used to predict the tags for the tokens by using CRF based models, it is a Java based script and is used for sequence tagging. The data is trained for 500 iterations with a 2 and 4 order chain for different feature sets and the test is computed on the 20% of test data.

The features selected to be used in the MALLET for NER tagging are as follows:

3.2 Glove Embedding

The Glove embedding are a general purpose trained embeddings on a large corpus, as our corpus used here is small it makes more sense to utilize general trained embeddings to avoid over fitting on the data. The Glove embeddings are present in 50, 100 and 300 length embeddings these are used to make the embeddings for the dataset, as we can in the result the results obtained are improved when using Glove embeddings.

3.3 POS tags

Parts of speech tags are made for the dataset using the NLTK library and assigning the POS tags using the ”pos_tag” function which assigns tags to the dataset based on general corpus trained rules, these are then used in conjugation with other features for NER in MALLET.

3.4 Capitalization

Tags for presence of Capitalized letter in a word are mentioned in this feature, this has been achieved using Regular Expression. The feature is calculated as the presence of a capital letter provides good basis for capital noun and they usually have a good chance of belonging to either ”D” or ”T”.

3.5 Inference

The table shows the accuracy with different components of features present in the training and test dataset the result shows that the all of the above features computed above are relevant in the context of NER and combination of all gives the best accuracy.

4 Neural Network based NER system

The NER system made by the Neural method is a BiLSTM based model with crf̀loss_function and crfàccuracy metric which allows to employ a neural method model compatible with a CRF model, the sentence are padded to the largest sentence and the features taken are only the POS tag and the word itself the network employs a 20 dimension network which is taken as the input in the system. The output accuracy is calculated on both the training and test dataset.

5 SKLEARN CRF based NER model

The model is constructed using the SKLEARN CRF-suite which holds the CRF library. The model is based on the ’lbfgs’ algorithm and the features provided are the POS tags along with the original words and the desired tokens. The output is computed on the 10 fold cross-validation and the results have been reported.

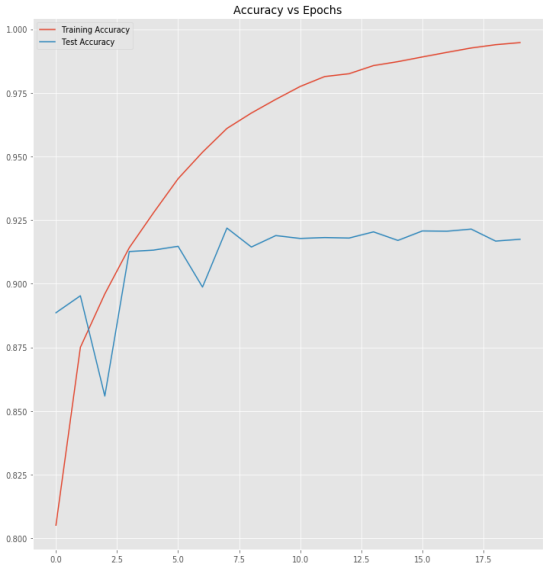

Figure 1: Accuracy vs Epoch plot for LSTM

Features	Order-2	Order-4
POS only	84.67%	85.81%
Capital only	81.82%	80.99%
Glove only	86.78%	87.62%
POS + Capital	84.91%	82.51%
Glove + POS	88.78%	88.51%
Glove + Capital	88.52%	89.01%
Glove + POS + Capital	89.72%	89.68%

Table 1: Test accuracy for different feature combinations using MALLET after 50 iterations.

Model	Train Acc.	Test Acc.
MALLET	89.72%	89.68%
SKLEARN CRF	93.00%	91.00%
LSTM based net	99.50%	93.42%

Table 2: Test accuracy for different feature combinations using MALLET after 50 iterations.

6 Comparison between Models

The results are evident that the LSTM based model far surpasses the conventional CRF models in both training and test accuracy. The Neural model trained as a LSTM with CRF loss function gives best of both worlds with LSTM based models in-

	precision	recall	f1-score	support
D	0.78	0.62	0.69	4889
O	0.94	0.98	0.96	59465
T	0.68	0.44	0.53	3821
avg / total	0.91	0.92	0.91	68175

Figure 2: Cross Validation result for SKLEARN CRF

corporating the CRF functions

From the MALLET results we can see that the features play little role in the performance of the model and a basic model is bound to perform at par with the model with the a large number of features. This can be attributed to the large number of the ”O”(Other) category of labels which shadow the slight improvement of the model discrimination.

The data is thus clearly evident towards the efficiency of the LSTM base Neural model towards the performance on the task of NER.

6.1 Limitations

The sentence labeling problems in this scale is mostly data constrained as the corpus given to us was a relatively small corpus the accuracy achieved was low however the accuracy can be further increased by increasing the size of the dataset.

Also, like in this dataset having a large portion of tags belonging to one category, i.e in this case ”O” causes the model to incline towards simpler algorithms which are biased towards the frequently assigned labels such as in this case. having multiple labels can solve this issue to a large extent.

6.2 Results

From the above discussion we can see the improved performance of the LSTM based models compared to the conventional CRF’s. This can be validated with the substantial increase in the accuracy on both training and test set.

The accuracy of the NER can be increased further if the sentence generation is trained on a larger portion of the train corpus, however that is likely to cause over fitting of the model.

7 Git hub Resource

The code for all the models have been uploaded on git hub with a read me file containing the instructions.

The files have been uploaded at:

https://github.com/Suraj-Panwar/NLU_Assignment_3.git