E1 246 - Natural Language Understanding
Assignment 3

Sachin Mittal (14539)
sachinmittal@iisc.ac.in

Abstract

The aim of NER is to classify words into some predefined categories. In this assignment I built an NER system for diseases and treatments. The input of the model is a set of tokenized sentences and the output is a label for each token in the sentence. Labels are D, T or O signifying disease, treatment or other.

1 Dataset

The format of each line in the training dataset is token label. There is one token per line followed by a space and its label. Blank lines indicate the end of a sentence. It has a total of 3655 sentences.

- **Preprocessing**- As mentioned above, Blank lines are end of the sentences. I read each sentence as a list of (word,tag) pairs. we have 3655 sentences containing 11311 different words with 3 different tags.

- **Data split** I have divided data into 80-20 percent as train and test respectively and further divided training data as 10% for validation data.

2 Implementation Details

I have used hybrid approach combining a bidirectional LSTM model and a CRF model.

- **Bidirectional LSTM** The idea of Bidirectional Recurrent Neural Networks (RNNs) is straightforward. It involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side, then providing the input sequence as-is as input to the first layer and providing a reversed copy of the input sequence to the second.

- **CRF**: We are given a input sequence $x = (x_1, \ldots, x_m)$, i.e. the words of a sentence and a sequence of output states $s = (s_1, \ldots, s_m)$, i.e. the named entity tags. In conditional random fields we modeled the conditional probability of the output state sequence give a input sequence.

- **A CRF on top of Bi-LSTM**: In this Assignment I used combined Bi-LSTM network and a CRF network to form a Bi-LSTM-CRF model, This network can efficiently use past input features via a LSTM layer and sentence level tag information via a CRF layer. A CRF layer is represented by lines which connect consecutive output layers. Bidirectional LSTM layer considers the previous input features and obtain sentence level tag information from the CRF layer. Therefore, the output is an optimal tag sequence instead of mutually independent tags.

  I have used **keras-contrib** package to achieve this goal.

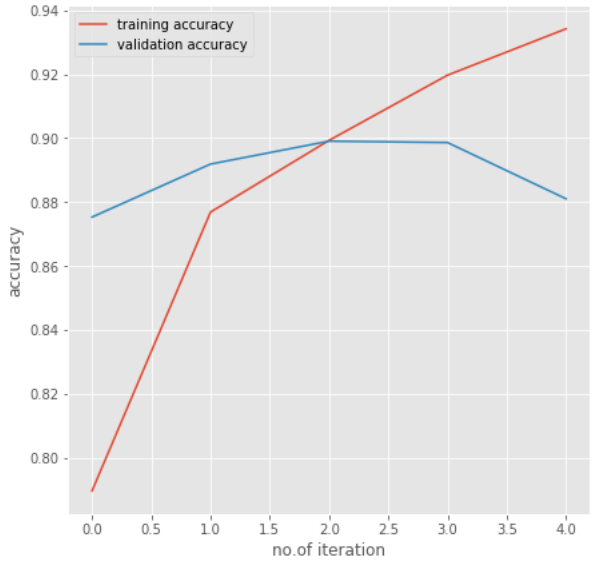
3 Training procedure

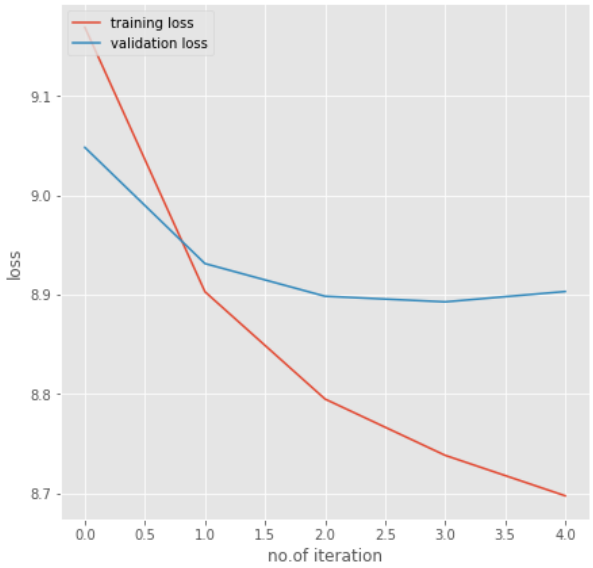
I have taken maximum length (number of words) of each sentence to 75 In each epoch, I divided the whole training data to batches and process one batch at a time. Each batch contains a list of sentences which is determined by the parameter of batch size. In my experiment, I used batch size of 32 which means to include sentences whose total length is no greater than 32.

4 Evaluation Metric

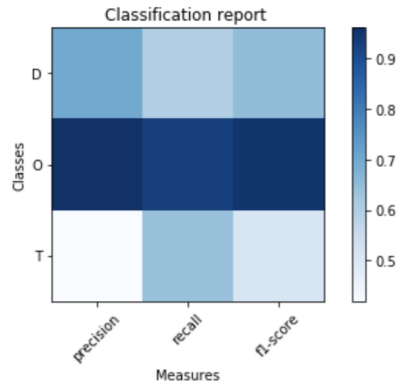
Evaluation metrics for classification tasks are Accuracy, F1 Measure, Precision and Recall.

5 Results





	precision	recall	f1-score	support
D	0.70	0.60	0.65	1049
O	0.96	0.93	0.95	10840
T	0.42	0.64	0.51	772
avg / total	0.90	0.89	0.89	12661



Predicted / True	D	O	T	All
D	633	213	203	1049
O	227	10119	494	10840
T	45	230	497	772
All	905	10562	1194	12661

Above graphs are self explanatory and shows that how different metric values are changes with each iteration.

classification report and confusion matrix are on test.txt which is 20% data of complete file.

6 Github Link

https://goo.gl/cfy9Sv