Natural Language Processing 2022
Homework 1
Named Entity Recognition

Mert YILDIZ
Sapienza University of Rome

1 Introduction

In this homework, identification and categorization task of the key information(entities) in text has been done which is known as Named entity recognition (NER). An entity in here can be any word or series of words that consistently refers to the same thing. For the sake of the task, every detected entity is classified into one of 13 predetermined categories. The categories are Person (PER), Location (LOC), Corporation (CORP), O , etc. and there are B,I,O prefixes where B- stands to indicate that the tag is the beginning of a chunk, I- stands to indicate that the tag is inside a chunk and O- indicates that a token belongs to NO entity or chunk. Therefore the categories are as follows: B-PER, B-LOC,…, I-PER, I-LOC, … , and O. For this task, as we were not allowed to use transformer based architecture, I have used a variety of Long Short-Term Memory (LSTM) based models for sequence tagging. To get better performance, I have used Bidirectional LSTM (BiLSTM), since knowing what words immediately follow and precede a word in a sentence, it effectively increase the amount of information available to the network and improving the context available to the algorithm. And, I have seen as expected that BiLSTM was way better compare to simple baseline LSTM.

2 Methods & Models

As previously mentioned, LSTM based models has been used for this task. To start with, I have preprocessed the given data and applied the baseline model. For me the baseline model was LSTM that starts with an embedding layer then pass the embedded tokens to the LSTM layer with only 1 number of layer, a last linear layer and log softmax or softmax to return the scores for each of 13 categories to be considered. The reason I have used log softmax but not directly softmax was, that I have used Negative Log Likelihood Loss function with log softmax and the Cross entropy loss with softmax. This is particularly useful when you have an unbalanced training set (PyTorch, 2019). With 300 embedding dimensions, 300 hidden dimension, learning rate of 0.1 and training for only 5 epochs the accuracy was 0.88 and the F1 score was 0.57 which is a good starting point without any tuning. Then I have tried to add more complexity to the model step by step and see what affects the performance in a good way. The first thing I have done was to increase the number of layers. As we know, the deeper the model the better the performance. Then I have seen that the model was over-fitting. To decrease the overfitting, I have used the dropout as regularization and it helped.
Another approach was to use BiLSTM. I have started with a simple BiLSTM model with one layer and no dropout. Then I have used more layers and used dropout again to decrease overfitting. The last approach was to use bidirectional LSTM with Conditional Random Field (CRF) layer (BiLSTM-CRF) which is known as one of the best approach that has been used for sequence tagging. In this approach, instead of log-softmax or softmax we use CRF layer. CRF predict a sequence based on the probabilities of transitioning from one label to another and the past scores. The score of a sentence is defined as the sum of the previous score, the transition probabilities and the network output. The transition probabilities are the bigrams scores giving information of the neighboring words.

3 Experiments & Results

Once I have checked the dataset, I have seen that the number of entities for different categories were unbalanced at all. The most frequent one is “O” and it is obvious that all models got the best accuracy and F1 scores for this category. On the other hand the models were not able to classify the entities of the least frequent category with a good performance. Also, the dataset were small but one might say that with a big enough data and at least more balanced entities by categories, all models performance would propel.

3.1 LSTM

The starting point was to get the best accuracy and F1 score with LSTM model. I have created several LSTM models by increasing the number of layers to 2 and 3, generating random embeddings for different dimensions, using Glove and Word2Vec pretrained embeddings. I have seen that 3 layers were overfitting too much and dropout was not helping enough with small portion and while I have increased the dropout amount performance affected in bad way which I think were causing information loss. Therefore, I have chosen the num_layer = 2 as the best. Pretrained embeddings were out performing the randomly generated embeddings. Word2Vec pretrained embeddings were not good enough since the number of existing words were less compared to Glove pretrained embeddings. Therefore, I have experienced that Glove pretrained embeddings with 300 dimensions performed the best. Then I have fine tunned the dropout proportion and the best was 0.2. To tune the learning rate, I have started with learning rate 0.001 and 0.01 but it was not converging fast. So, I have decided to start with learning rate 0.1 and update the learning rate step by step to 0.01 and 0.001 after getting under 0.1 loss. The best accuracy and F1 score I had is as written in the Table 1.

3.2 Bidirectional LSTM

As previously stated, since knowing what words immediately follow and precede a word in a sentence effectively increase the amount of information available to the network and it improves the context available to the algorithm the second approach was Bidirectional LSTM. I have built several models as in LSTM section with different num_layers, pretrained embeddings and fine tunned the dropout and learning rate. The best model was again with pretrained 300 dimensions Glove embeddings, num_layers = 2, dropout = 0.2 and learning rate = 0.001. This approach out performed LSTM models. The results can be seen in Table 1.

3.3 BiLSTM – CRF

The last approach was BiLSTM with CRF layer. I have built and run only one model which has 2 layers and I have finetuned the dropout proportion and learning rate. The best configuration for BiLSTM-CRF was again with dropout = 0.2 and I have started with learning rate = 0.1 then I have decreased it to 0.01 and 0.001 after some epochs. I had the best F1 score with this model compared to LSTM and BiLSTM.

Model	Accuracy	Macro F1
Baseline(LSTM)	0.88	0.57
Baseline(BiLSTM)	0.90	0.65
Pretrained W2V LSTM	0.88	0.57
Pretrained W2V BiLSTM	0.91	0.68
Pretrained Glove LSTM	0.90	0.63
Pretrained Glove BiLSTM	0.92	0.71
BiLSTM - CRF	0.93	0.73

Table 1. Accuracy and F1 Scores of Models Based on Dev Data

4 Conclusions

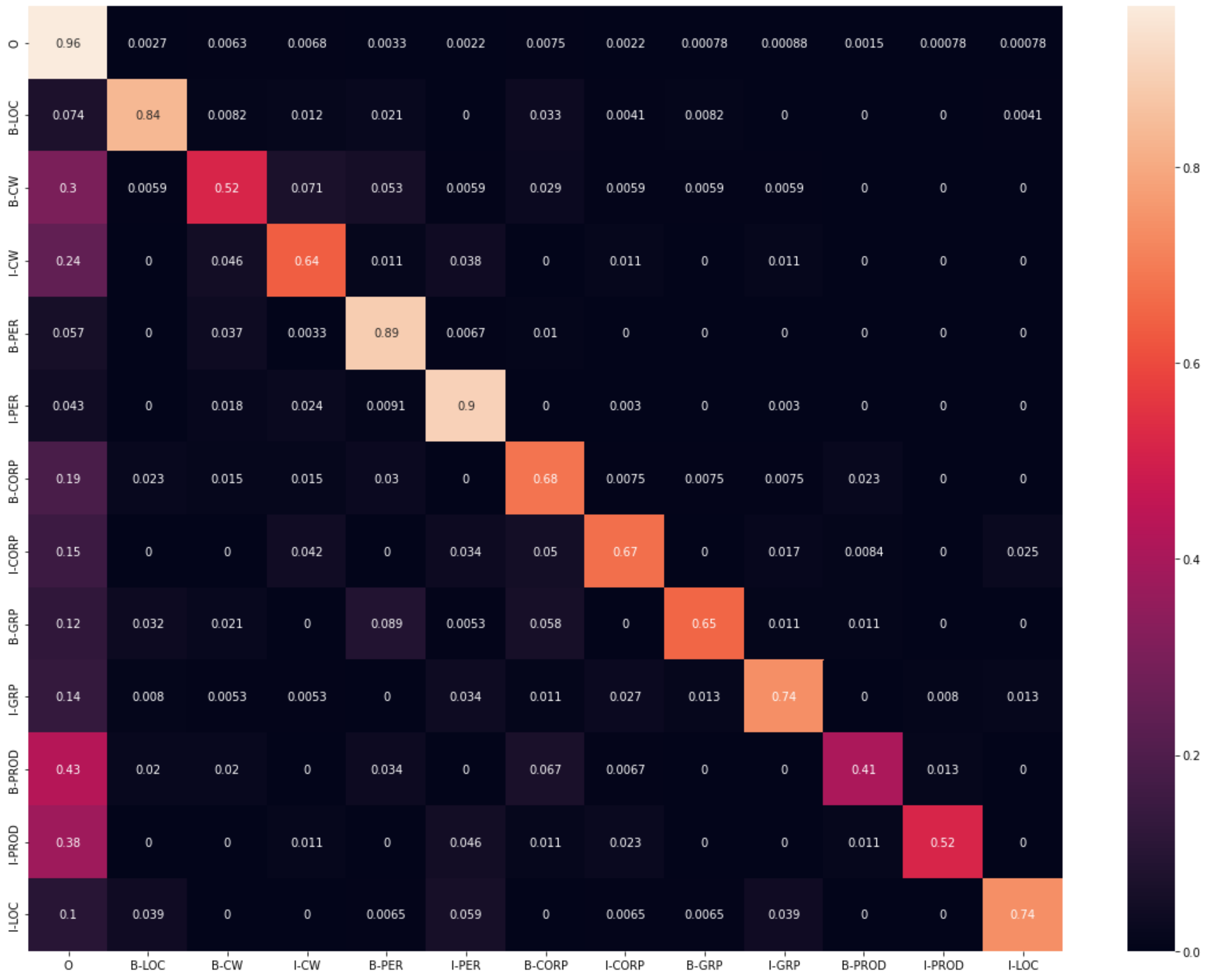
Named Entity Recognition is an important NLP task that has been improved nowadays. In short, I have tried to tag sequences in this homework. Mostly used model are RNNs and LSTM models for this task. I have applied a various LSTM models and had an insight to see how this task can be done. As I have mentioned in the previous sections, Bidirectional LSTM model out perform the LSTM model and BiLSTM with CRF layer outperform BiLSTM model itself.

5 References

NLLLoss - PyTorch 1.11.0 documentation. (n.d.). Retrieved April 19, 2022, from https://pytorch.org/docs/stable/generated/torch.nn.NLLLoss.html

6 Appendices

Confusion Matrix of the best model of the best configuration



	O	B-LOC	B-CW	I-CW	B-PER	I-PER	B-CORP	I-CORP	B-GRP	I-GRP	B-PROD	I-PROD	I-LOC
O	0.96	0.0027	0.0063	0.0068	0.0033	0.0022	0.0075	0.0022	0.00078	0.00088	0.0015	0.00078	0.00078
B-LOC	0.074	0.84	0.0082	0.012	0.021	0	0.033	0.0041	0.0082	0	0	0	0.0041
B-CW	0.3	0.0059	0.52	0.071	0.053	0.0059	0.029	0.0059	0.0059	0.0059	0	0	0
I-CW	0.24	0	0.046	0.64	0.011	0.038	0	0.011	0	0.011	0	0	0
B-PER	0.057	0	0.037	0.0033	0.89	0.0067	0.01	0	0	0	0	0	0
I-PER	0.043	0	0.018	0.024	0.0091	0.9	0	0.003	0	0.003	0	0	0
B-CORP	0.19	0.023	0.015	0.015	0.03	0	0.68	0.0075	0.0075	0.0075	0.023	0	0
I-CORP	0.15	0	0	0.042	0	0.034	0.05	0.67	0	0.017	0.0084	0	0.025
B-GRP	0.12	0.032	0.021	0	0.089	0.0053	0.058	0	0.65	0.011	0.011	0	0
I-GRP	0.14	0.008	0.0053	0.0053	0	0.034	0.011	0.027	0.013	0.74	0	0.008	0.013
B-PROD	0.43	0.02	0.02	0	0.034	0	0.067	0.0067	0	0	0.41	0.013	0
I-PROD	0.38	0	0	0.011	0	0.046	0.011	0.023	0	0	0.011	0.52	0
I-LOC	0.1	0.039	0	0	0.0065	0.059	0	0.0065	0.0065	0.039	0	0	0.74