NLP & ML Assignment 3: Named Entity Recognition

Galieva Rufina

1 Link to solution in Google Colab

`https://colab.research.google.com/drive/1w6jNtHCIX3BUA-_dX14Ui0z7Ob2Aiyb8`
Please note that model takes a lot of time to train.

2 Problem statement

The goal of this assignment is to implement named entities tagger for social media data in English language.

3 Datasets description

We take Broad Twitter Corpus [1] as data for our experiments as suggested in the assignment. Data contains seven unique tags 23,749 unique words and 7,642 unique sentences. Data is divided on 6 sections, named *a,b,e,f,g,h.*This data is in the main interest for us.

Another larger dataset that we are using for pretraining the model is Annotated Corpus for Named Entity Recognition taken from [5]. This dataset contains 47,959 sentences, 35,178 words and 17 tags.

3.1 Data preprocessing

At the beginning we remove all non words from the data obtaining 21,649 unique word in case of Broad Twitter Corpus and 29,903 words in case of Annotated Corpus. Before passing the data to the model we replace words with their unique ids and same we apply for tags, adding <start> and <end> tags at the beginning and at the end of the sentence respectively. As suggested in [1] for the Broad Twitter Corpus dataset we take first half of h section as test data, the rest of the data is considered training data 30% of which is validation set. In case of Annotated Corpus we randomly split data on train (70%) and test (30%), 30% of train set is taken for validation.

4 Model

As a base o the model we take the one provided in PyTorch tutorial MAKING DYNAMIC DECISIONS AND THE BI-LSTM CRF [2] which is based on research [6]. We decided to

go with this model because according to the [3] LSTM-CRF is a state-of-the-art approach to named entity recognition. More on Conditional Random Field Model (CRF) theory can be found at [6].

Why Bidirectional LSTM? That is how model will use both past and future input features.

Why CRF? That is how model can use sentence level tag information. The point is to focus on sentence level instead of individual positions, this is called Conditional Random Fields models [7]. In this case the inputs and outputs are directly connected.

The overall **Bidirectional LSTM-CRF** model architecture is illustrated in Figure 2.

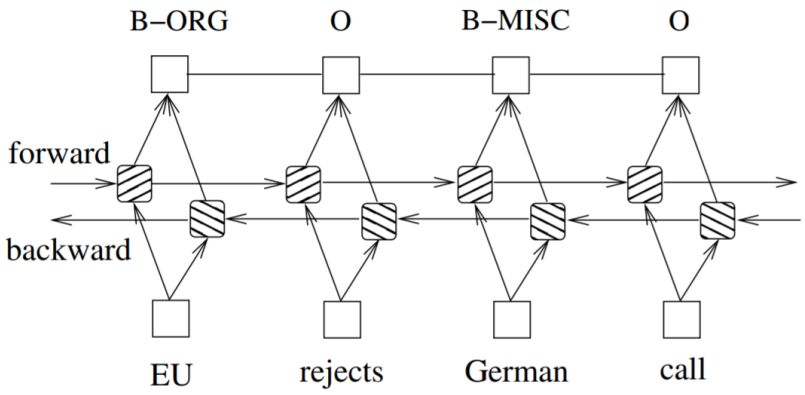

Figure 1: A BI-LSTM-CRF model. Picture from [6].

4.1 Evaluation metrics

We evaluate our model using F1-score reported for each tag as well as overall F1-score computed averaged by entities F1-scores.

4.2 Model training

For the implementations we choose PyTorch tool as provided in the tutorial, because PyTorch allow us creating dynamic graphs and do not waste computations on padding.

We take the following stages in our training pipeline:

- **Model 1** Train model and tune parameters directly on Broad Twitter Corpus, to see if model can learn tagging from this dataset directly.
- **Model 2** Train model and tune parameters on Annotated Corpus, which is larger.
- **Model 3** Use pretrained model from previous point as a starting point for training on Broad Twitter Corpus, e.g. transfer learning. We use lstm parameters of previous model, change last fully connected layer and embedding layer.

Model remained same in all cases and the best obtained parameters are reported in Table 1. We trained all models using stochastic gradient descent to minimize negative log likelihood as in [3]. Training process for model 1 can be found in Figure 2 and Figure 3. We do not report model training for model 2 because due to large dataset and training in colab after training one epoch model was saved and loaded again to train another epoch.

Parameter	Value
Embedding dimention	100
Learning rate	0.01
LSTM hidden dimention	32

Table 1: Best obtained parameters using manual search. Please note, that we might obtain better parameters if conducted more experiments, but due to time constraints and time that model takes to train could not do more.

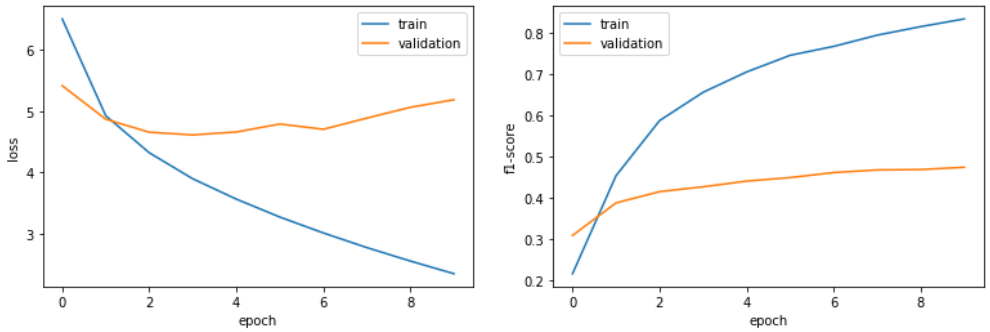

Figure 2: Training process for model 1, e.g. trained only on Broad Twitter Corpus. Left: changes in training/validation loss. Right: changes in training/validation macro f1-scores

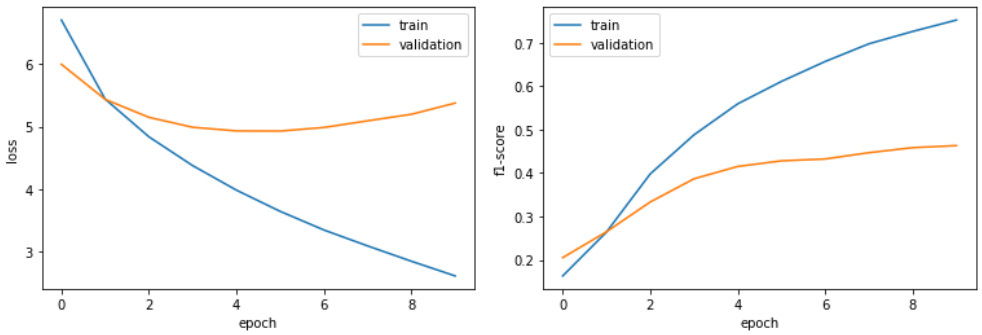

Figure 3: Training process for model 3, e.g. trained on Broad Twitter Corpus using transfer learning. Left: changes in training/validation loss. Right: changes in training/validation macro f1-scores

4.3 Results

The resulting score of three models are presented in tables below.

Metric	Train set	Validation set	Test set
Macro F-1 score	83	47,5	37
Weighted F-1 score	97	89	89
I-PER	89	45	50
B-PER	85	51	53
B-LOC	85	44	48
O	99	95	95
I-ORG	77	32	0
B-ORG	69	33	14
I-LOC	80	33	0

Table 2: Model scores on Broad Twitter Corpus without Transfer learning after training for 10 epochs.

5 Common mistakes & Common non mistakes

To analyze the mistakes we report confusion matrix and normalized confusion matrix based on validation and test sets in Figure 4.

As we can see from confusion matrix the most common mistake is that model predicts tag *O* instead of true tag. As well as often predicts wrong tag for true tag *O*.

We can see that model is able to recognize tag *O* very well, which is obvious, since it has a lot of data for that tag. Also it is good to see that model can recognize *B-LOC, B-PER, I-LOC, I-PER* tags pretty well. It is sad to see that model struggles to recognize *B-ORG, I-ORG* tags.

6 Conclusion

The main challenge of this is task is imbalanced data, model can learn tag *O* pretty well, but struggle with other tags. According to results model show promising results and might show better performance, for that more experiments with parameters are required.

We can note that Bidirectional LSTM-CRF gives not bad results on Broad Twitter Corpus even without pretraining the model, that means model can learn some dependencies. But using pretrained model we obtain slightly worse results in terms of macro F1-score and weighted F1-score. We can only explain this kind of behaviour by different nature of the datasets and first model being not enough pretrained.

Metric	Train set	Validation set	Test set
Macro F-1 score	54	47	6
Weighted F-1 score	96	94	37
O: 0	99	98	1
B-org: 1	70	57	10
I-geo: 2	80	65	0
I-org: 3	76	54	41
B-nat: 4	2	0	0
B-per: 5	73	61	11
B-tim: 6	91	87	0
I-gpe: 7	59	57	0
B-eve: 8	37	32	0
I-eve: 9	0	0 0	
I-art: 10	0	0	0
I-nat: 11	0	0	0
I-tim: 12	65	48	0
B-geo: 13	88	80	0
B-gpe: 14	95	93	0
I-per: 15	82	70	0
B-art: 16	0	0	0

Table 3: Model scores on Annotated Corpus after training for 3 epochs.

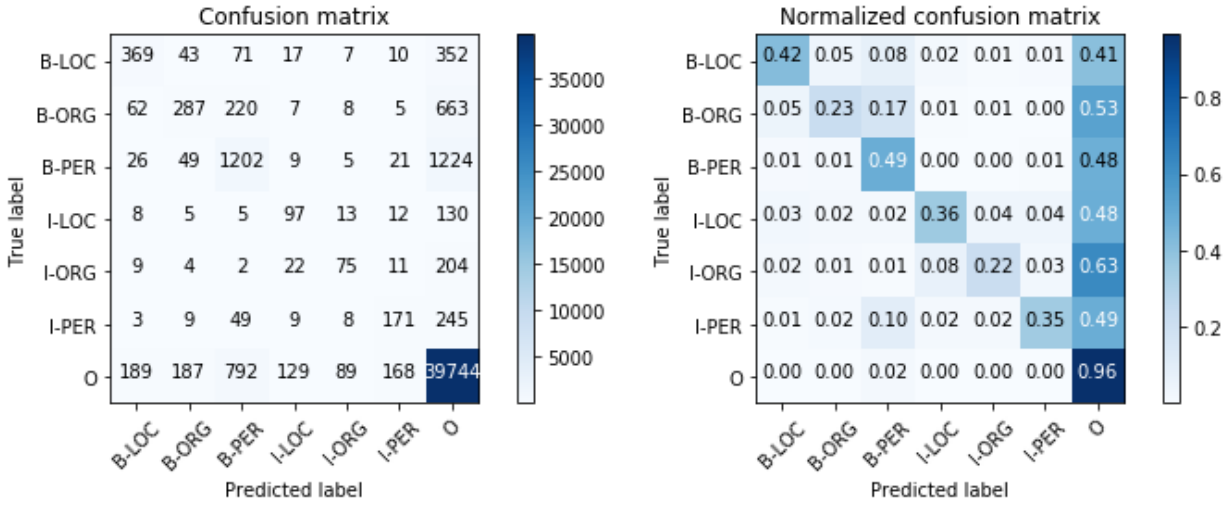

Figure 4: The confusion matrix of the model 3 on Broad Twitter Corpus. Left: Non normalized confusion matrix. Right: Normalized confusin matrix.

Metric	Train set	Validation set	Test set
Macro F-1 score	75	45	34
Weighted F-1 score	95	88	88
I-PER	80	39	22
B-PER	73	47	55
B-LOC	73	47	55
O	98	95	95
I-ORG	68	29	0
B-ORG	57	30	11
I-LOC	67	29	0

Table 4: Model scores on Broad Twitter Corpus with Transfer learning + training for 10 epochs.

References

[1] Broad Twitter Corpus: A Diverse Named Entity Recognition Resource. Leon Derczynski, Kalina Bontcheva, and Ian Roberts. Proceedings of COLING, pages 1169-1179 2016.

[2] PyTorch: MAKING DYNAMIC DECISIONS AND THE BI-LSTM CRF. https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html

[3] Sequence Tagging With A LSTM-CRF. https://www.depends-on-the-definition.com/sequence-tagging-lstm-crf/

[4] Conditional Random Fields Explained. Aditya Prasad. https://towardsdatascience.com/conditional-random-fields-explained-e5b8256da776

[5] Annotated Corpus for Named Entity Recognition. https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus

[6] Huang, Zhiheng, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015).

[7] J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of ICML.