NLP Task

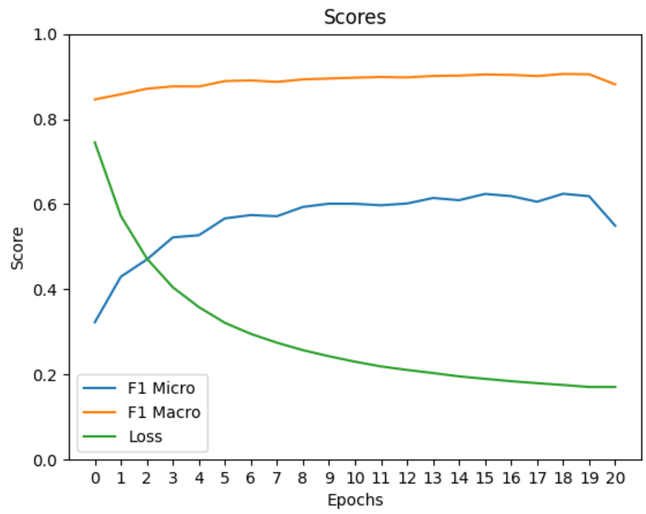
Sewin Tariverdian, 2903040

TECHNISCHE UNIVERSITÄT DARMSTADT

WiSe 2021/2022

1 Evaluation

1.1 Report Scores



Epochs	F1 Macro	F1 Micro	Loss
1	0.846	0.322	0.745
2	0.858	0.429	0.572
3	0.871	0.470	0.472
4	0.876	0.522	0.404
5	0.876	0.527	0.358
6	0.889	0.566	0.321
7	0.890	0.574	0.295
8	0.887	0.571	0.274
9	0.893	0.593	0.257
10	0.895	0.601	0.242
11	0.897	0.600	0.230
12	0.898	0.597	0.219
13	0.897	0.601	0.210
14	0.901	0.614	0.203
15	0.902	0.609	0.195
16	0.904	0.624	0.189
17	0.903	0.619	0.184
18	0.900	0.605	0.179
19	0.905	0.624	0.175
20	0.905	0.618	0.170
test set	0.881	0.549	0.170

Figure 1: Scores

The macro-averaged and micro-averaged F1 scores on the dev set for 20 epochs are shown in the figure above. We can spectate, that they differ significantly on absolute values. The macro-averaged score starts with a rating of 0.85 and increases to an all-time-high of 0.90. The micro-averaged score achieves approx 0.32 before training and achieves around 0.61 after training.

The high score from the macro-averaged score could be a product of a high occurrence of "O"-tags. These are easier to detect for the model. This shifts the global score up.

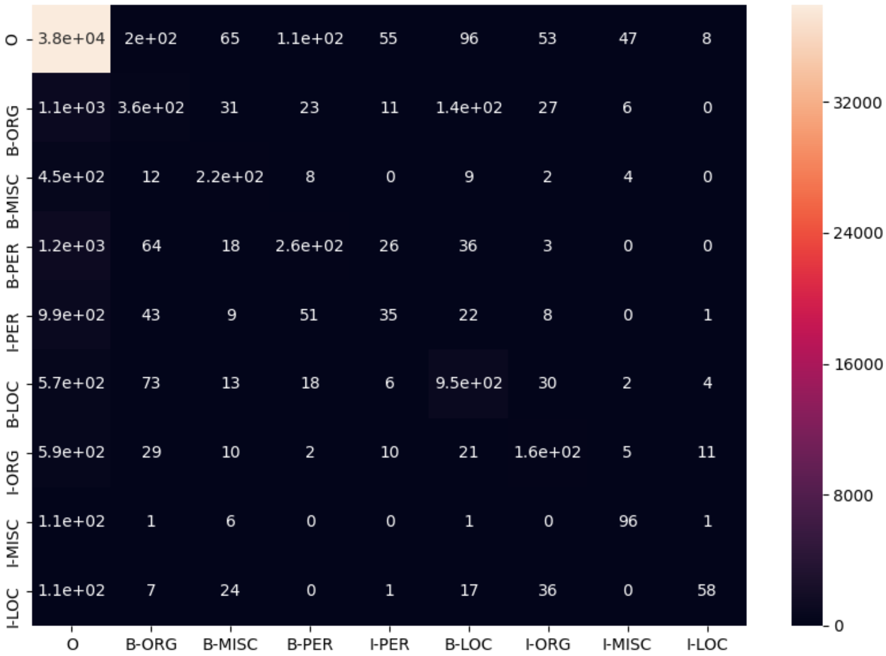


Figure 2: Confusion Matrix after 4 epochs [Truth labels on y-axis, predicted labels on x-axis]

Our suspicion is proven if we look up the confusion matrix after some iterations. The O-tag dominates the data situation. The micro-average score is more considerable of class-specific values. This way the micro-averaged score is more suited for this task.

1.2 Fails

To get a better understanding of our outputs and the situations in which it seems to fail, we take a look again at the confusion matrix. But this time we mute the correct classified "O"-tags so other elements wont get polluted by the high amount of "O"-tags in the data-set.

Figure 3: Confusion Matrix after 19 epochs without "O"-tag [Truth labels on y-axis, predicted labels on x-axis]

In a perfect model we would expect a bright diagonal coloring surrounded by dark fields. We can observe that the model is inclined to assign the "O"-tag on labels in which it is not sure. This seems to happen very often. But also we observe that "I-PER" and "B-PER" are often mixed up.

Example:

Input	BOXING	-	**SCHULZ**	DEFEATS	RIBALTA	IN	**IBF**	**HEAVYWEIGHT**	**FIGHT**	**.**
Truth	O	O	**B-PER**	O	B-PER	O	**B-ORG**	**O**	**O**	O
Predicted	O	O	**O**	O	I-PER	O	**O**	**I-PER**	**B-PER**	O

Figure 4: Example of miss classified data

1.3 Improvement

Summing up the main issues are miss-classification of "O"-tags and the dis-ambiguity of names or similar. This could be caused by over-fitting. To strengthen the influence of neighboring words i would suggest the use of probabilistic models[1]. Also the use of a pre-trained transformer could raise the accuracy.

Generally a primitive sequence tagger can be improved by concepts listed below:

- Augmenting tagger with character-level features
- Add more hidden layer
- Find more suited hyperparameters
- Early stopping with Checkpointer

References

[1] John D. Lafferty, Andrew McCallum, Fernando Pereira (2001) *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*