# Report on the NER project

By Tatjana Cucic

I will start by reporting everything I did on this project and after that, I will share my impressions, as well.

## Milestones 1 and 2

### Simple RNN

The first thing I did for the project was try to play with the parameters and make the Simple RNN model work. I tried different activation functions like ReLU, sigmoid and tanh, as well as various optimizers like Adam, Adamax, Adadelta, RMSprop, etc. I tried training with various combinations of optimizers and activation functions, as well as learning rates. I also experimented with batch sizes and number of epochs, but all results were very similar. The only thing that actually made a difference was lowering the learning rate, so I started experimenting with a learning rate of 1e-10 to 1e-4 and 1e-4 gave the “best result” of around 0.38 f-score.

However, once I ran the predictions, they were not very good, which was kind of expected. The tags were various, but mostly inaccurate.

Also, I had an idea because RNNs have short memory that I should shorten our training dataset by eliminating very long sentences. Our longest sentence was 160 something long, so I decided to cut the longest sentence into half. However, the results did not improve much.

### LSTM

After that, I tried dabbling with LSTM and that was just horrible. I still don’t know why the training of the LSTM gave such bad results. At first, it was giving a nan value for the loss and an f-score of around 0.9. that made me very happy, but then after looking at the predictions, I realized that the score was that high because all predicted values were the same (nan). I thought the reason for this might have been because at some point there may have been division by zero which would give nan values as a result, so I tried to change hyperparameters. I also tried many combinations of optimizers, activation functions, batch sizes, learning rates and hidden layer sizes until I reached a “good” result regarding the loss. The loss was no longer nan and it was actually declining during training down to a small value (around 0.04). However, the precision, recall and f score were all zeros, but sadly it was the “best” I could do with this model.

I did not know why the results were that bad, so I tried to reread the data using Pandas Dataframes and encountered numerous other problems. The files were not parsed well, so “\t” actually stuck around in many random places throughout the data. Once I removed those strange strings from “words” and “tags” in the corresponding places, I still had the longest sentence length of around 12 thousand. After inspecting the data further, I realized it was not enough to parse the sentences by dots, question marks or exclamation points, but some other strange characters like {\, /, #, [,], etc}, so I eventually gave up on rereading the data and went on with BERT.

I ran the predictions with this horrible model, as well and it gave the same tag for every word, but at least they were not nan values which I considered a “success” after all the nan results I got beforehand.

### BERT

I followed the instructions given in the lectures for implementing BERT, used a 12-layer pretrained BERT and it worked very well on our data. I trained it for 3 epochs and it reached an accuracy of around 0.95. The GDrive folder contains all 3 models with a follow-up image containing a graph of the training and/or some predictions, as well. I was very excited about BERT results, so I added them to the image, as well.

I saved all 3 models that I trained together with the images made in paint to help you grasp the results, since I was unsure about which outputs will stay in the notebook by the time we finish the project. Since the models are too large for GIT, I uploaded them on my Google Drive: <https://drive.google.com/drive/folders/15nZK7b4HSr771lP4kvdUbgn4o4UFYfEV?usp=sharing>

## Milestone 3

Unfortunately, I did not manage to do anything regarding Milestone 3, even though I think it would have been interesting to test the models on a new text and analyze the results, but I did not have enough time or good enough models (except BERT) to try it.

## Impressions

This project was very difficult and challenging for me. It may seem easy, but I encountered so many difficulties throughout it. The first attempt to finish it on time failed miserably, because the last day of the deadline I realized that our longest sentence is 17 thousand words long which made my computer crash constantly, as well as Colab (and I thought it was normal because of a large dataset).

After that, it became a bit easier, but even more frustrating because I could not get a functioning LSTM model and I tried very hard. It was very time-consuming and unsuccessful.

Also, I am an exchange student with a background in pure mathematics and a desire to be a data scientist. The project was very interesting, but my coding skills or my knowledge of deep learning probably did not fulfill the prerequisites for this course.