1. Introduction
2. Task Description
3. Dataset

3.5 roberta

1. Approach
2. Results and discussion
3. Conclusion

Abstract

Transformers models like XLM-RoBERTa can today perform challenging classification tasks. This capability is particularly crucial in applications such as sentiment analysis, opinion mining, and personalized content recommendation. In this project we perform supervised learning on the XLM-RoBERTa to train it to do subjectivity detection in multiple languages and we measure it’s performance relative to other models. The approach consists of training a language separately and training on all languages simultaneously.

## 2. Task description

The goal of task 2 at Clef2024 is binary classification task of a sentence or a paragraph from a news article, to distinguish whether it is subjective or objective.

On the technical side, the task is supposed to be done on the XLM-RoBERTa-Large transformer model, with the baseline comparison done on a different model like GPT4, Mistral or Llama2.

The datasets are given as individual languages like English, German, Arabic, Bulgarian, Italian and as a multilingual set that mixes those languages.

The task will be achieved my developing and training a model capable of performing the identification of subjectivity.

## 2. Datasets

Given datasets incorporate sentences taken from news articles labeled as either objective or subjective sentences. There are 3 columns in a dataset:

-SentenceId – ID of the sentence

-Sentence – the string holding the sentence

-Label – marking if the sentence is objective (OBJ) or subjective (SUBJ)

Datasets are divided into subtasks of individual languages, and combined subtask of all languages. Each subtask has a train set, a development (dev) set and a development-test (dev-test) set.

English subtask contains:

830 train sentences, 352 OBJ, 298 SUBJ

219 dev sentences, 106 OBJ, 113 SUBJ

243 dev-test sentences, 116 OBJ, 127 SUBJ

subtask-2A-italian

train: 1613 sentences, 1231 OBJ, 382 SUBJ

dev: 227 sentences, 167 OBJ, 60 SUBJ

dev-test - 440 sentences, 323 OBJ, 117 SUBJ

german

train: 800 sentences, 492 OBJ, 308 SUBJ

dev: 200 sentences, 123 OBJ, 77 SUBJ

dev-test - 291 sentences, 194 OBJ, 97 SUBJ

bulgarian

train: 733 sentences, 382 OBJ, 312 SUBJ

dev: 106 sentences, 59 OBJ, 47 SUBJ

dev-test - 208 sentences, 116 OBJ, 92 SUBJ

arabic

train: 1185 sentences, 905 OBJ, 280 SUBJ

dev: 297 sentences, 227 OBJ, 70 SUBJ

dev-test - 445 sentences, 363 OBJ, 82 SUBJ

multilingual24

train: 4428 sentences, 3160 OBJ, 1268 SUBJ

dev: 40 sentences, 40 OBJ, 0 SUBJ

dev-test - 400 sentences, 200 OBJ, 200 SUBJ

The data had to be preprocessed by removing some of the special signs in sentences, which were creating trouble in reading the file. Then, the labels in the data frames were binarized, with the OBJ label being replaced with 0, and SUBJ replaced with 1. The further preprocessing was done by the tokenizer, to make the data feedable to the transformer model.

Multilingual 2024 dev set has a low number of sentences, all being objective. This could cause problems, and to fix this we have taken some sentences from train set and move them to the dev set instead.

## Approach

In this chapter we go over the planned approach for the task. We have trained Roberta on individual language datasets, and also trained it separately on the multilanguage dataset.

XLM-RoBERTa – TODO

The task approach has followed those steps:

1. Selecting values for model parameterization: we tried to find most suitable values for epochs, learning rate and other parameters used in a training the model. We have fine-tuned the model with from 2 to 8 epochs. Validation was done with the validation dataset.
2. Training the model – in this step the model was trained on the train dataset and validation dataset with labels, which is a supervised learning method. The data was encoded into tokens by the RoBERTa tokenizer.
3. Making predictions – predictions are made on the dev-test set, on which we calculate the accuracy of the model and other parameters like precision and f1-score
4. Reevaluating the parameters – comparison is made between achieved results and previous results to evaluate the parameters and the approach.

## Experimentation

TODO 2 approaches, changed parameters (learning speed, drop speed, epochs), sampling

At first, we have started the work on the English dataset as it was relatively small and could be trained quickly. We have started with a low number of epochs, which is how many times does the model go over the dataset during training, starting with 2 epochs and going up 1 epoch with each run, up to 6-8 epochs, depending on the result. Usually the performance of predicting improved with the epoch number, although at some point, usually at epochs above 5 or 6, the model’s performance would get worse and it would make wrong predictions.

TODO pics with epoch performances

At some higher number of epochs we could also see signs of overfitting, where the train loss kept going down, but the validation loss started to go up.

Later on when we were sure that the model training pipeline works, we moved to other languages and the multilingual dataset. We have noticed that applying the same parameters for every language would not necessarily result in similar performance.

TODO pics with different lang performances

Learning rate controls how fast the model learns during the training, in our case its based on feedback it gets from the label classification. We have experimented with changing the learning rate in our runs with different values like: 5e^(-5), 3e^(-5), 5e^(-6), 1e^(-5), 5e^(-4). During the experimentation we tried to minimize the learning rate, and thus find the optimal value of it, while keeping the accuracy as high as possible.

## Results

In this section, we go over the results obtained and we compare them against the baseline results, achieved with GPT4 classifier.

Figures

Epoch f1 score

Increasing val loss

25 0.690100 0.709527

50 0.645700 0.737487

75 0.641400 0.691304

100 0.571000 0.780878

125 0.571900 0.854978