Multilingual subjectivity in News Articles, Group 15

JAKUB MROZ, StudId 260703, j.mroz@stud.uis.no
MUHAMMAD FAHAD NAWAZ RANA, StudID 277542, 277542@uis.no

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.

ACM Reference Format:
Jakub Mroz and Muhammad Fahad Nawaz Rana. 2024. Multilingual subjectivity in News Articles, Group 15. *ACM Trans. Graph.* 37, 4, Article 111 (August 2024), 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 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.

Subjectivity detection enables researchers, analysts, and organizations to gain insights into the diverse perspectives present in articles, facilitating more nuanced understanding and decision-making in various domains.

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.

Authors' Contact Information: Jakub Mroz, StudId 260703, j.mroz@stud.uis.no; Muhammad Fahad Nawaz Rana, StudID 277542, 277542@uis.no.

ACM 1557-7368/2024/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

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

2.1 Datasets overview

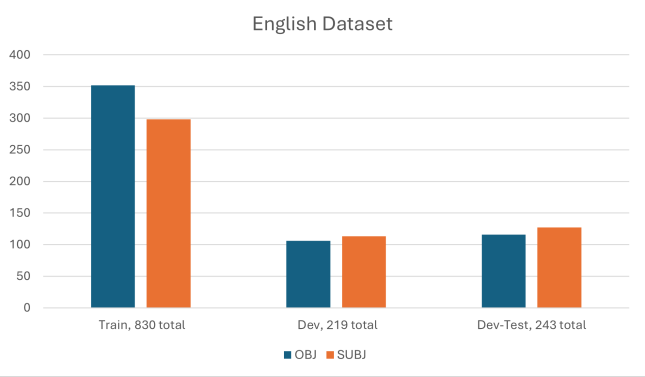

Fig. 1. English dataset

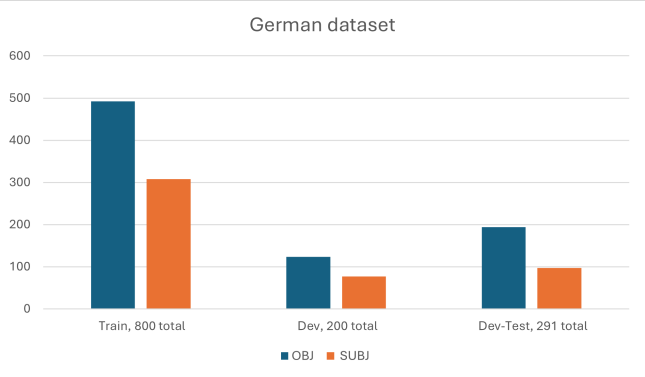

Fig. 2. German dataset

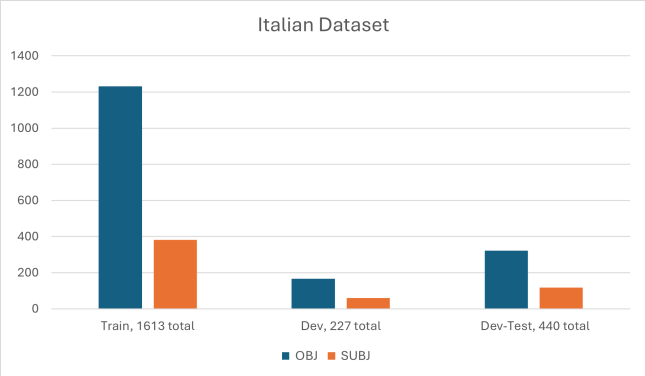

Fig. 3. Italian dataset

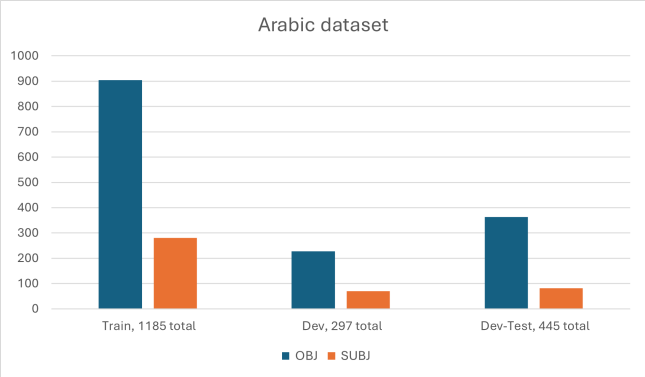

Fig. 4. Arabic dataset

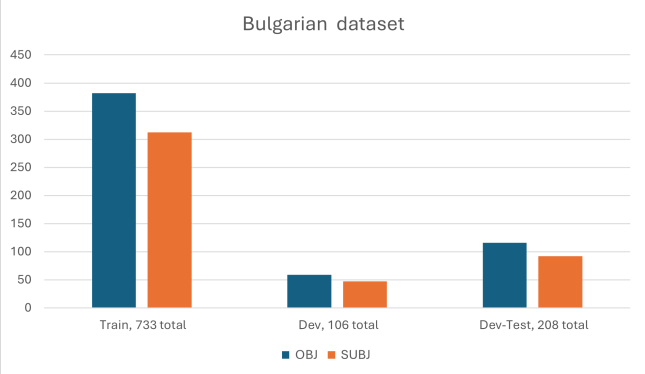

Fig. 5. Bulgarian dataset

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.

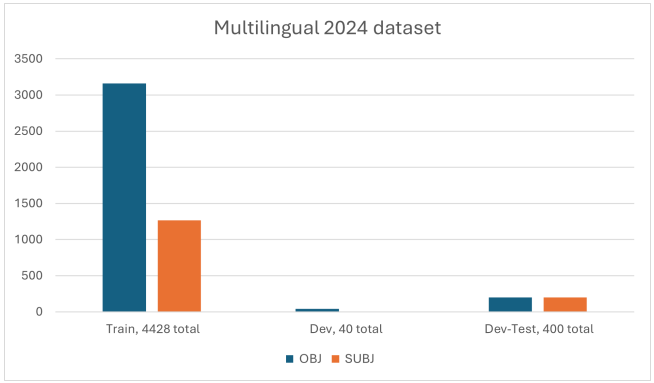

Fig. 6. Multilingual dataset

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.

3 XLM-ROBERTA

XLM-RoBERTa is a state-of-the-art language model that combines the benefits of cross-lingual pretraining and RoBERTa. It is designed for natural language understanding tasks. It can work on variety of languages. It has already been used in classification tasks like detecting hate speech or sexism, it works well in those scenarios thanks to its cross-language capabilities

4 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.

The task approach has followed those steps:

- 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.
- 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. Labels were binarized into 1s and 0s, and the data was encoded into tokens by the RoBERTa tokenizer.
- 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
- Reevaluating the parameters – comparison is made between achieved results and previous results to evaluate the parameters and the approach.

5 EXPERIMENTATION

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. This is shown in the Figure 7.

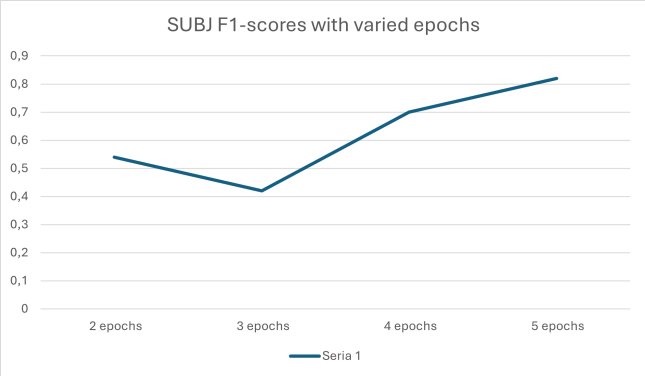

Fig. 7. F1 scores on subjectivity with varied number of epochs during training

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. This is shown in the Figure 8.

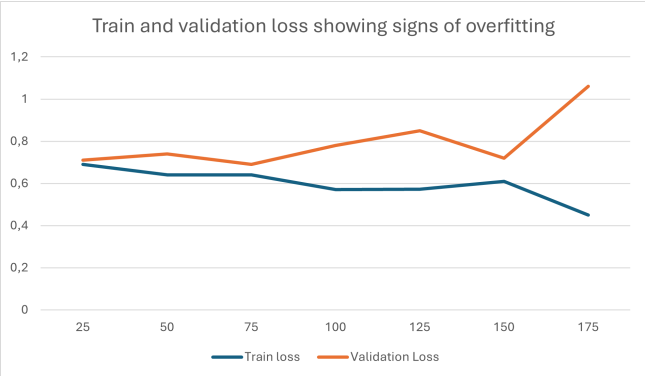

Fig. 8. Signs of overfitting

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. This can be seen in figures 9 and 10, where same parameters were applied to the model, and yet the resulting F1-score on subjective sentences was much different.

```
Accuracy for RoBERTa: 0.7283950617283951
Classification Report:
              precision    recall  f1-score   support

         OBJ       0.66      0.91      0.76       116
        SUBJ       0.87      0.57      0.69       127

    accuracy                           0.73       243
   macro avg       0.76      0.74      0.72       243
weighted avg       0.77      0.73      0.72       243
```

Fig. 9. English results 4 epochs 5e-5 learning rate

```
Accuracy for RoBERTa: 0.8359550561797753
Classification Report:
              precision    recall  f1-score   support

         OBJ       0.84      0.98      0.91       363
        SUBJ       0.70      0.20      0.30        82

    accuracy                           0.84       445
   macro avg       0.77      0.59      0.61       445
weighted avg       0.82      0.84      0.80       445
```

Fig. 10. Arabic results 4 epochs 5e-5 learning rate

5.1 Learning rate

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. Higher rates may expedite convergence but risk overshooting optimal solutions, while lower rates may enhance generalization.

We have also experimented with dropout probability: Adjusted between 0.1, 0.2, and 0.5 to regulate model regularization. Higher probabilities introduce more randomness, aiding in preventing overfitting and improving generalization.

6 TECHNICAL PROBLEMS AND LIMITATIONS

The Bulgarian dataset and the Italian dataset caused the model to crash during training due to insufficient memory on our systems. This was probably caused by incorrect batching,

but we were not able to fix the issue on time. This is the reason we could not get results on the Bulgarian and Italian subtask.

The multilingual dataset was bigger in size than individual language datasets, which cause the training time to also be increased relatively to the number of rows. To save time, we tried sampling it down to half of it, and we still got similar results when compared with the full dataset.

7 RESULTS AND DISCUSSION

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

The result of prediction are calculated with Scikit-learn implementation of Accuracy Score and Classification Report. The Classification report shows metrics like precision, recall, f1-score and accuracy. Precision measures accuracy of positive predictions. High precision means low false-positive rate. Recall measures the ability of a model to find all the relevant cases within a dataset. F1-score is a combination of those 2, as both are important for classification. Accuracy is the ratio of the total number of correct predictions and the total number of predictions.

The scores are calculated by comparing the labels of test sets against the predictions done by the model. The best scores between languages we were able to achieve was about 80%, while averaging about 75%. The most accurate scores were achieved at 3 - 5 epochs.

Accuracy for RoBERTa: 0.7942386831275721
Classification Report:

	precision	recall	f1-score	support
OBJ	0.78	0.78	0.78	116
SUBJ	0.80	0.80	0.80	127
accuracy			0.79	243
macro avg	0.79	0.79	0.79	243
weighted avg	0.79	0.79	0.79	243

Fig. 11. Best English results

In the figure 15 there is a PRC curve which maps recall and precision. The ideal model's curve goes all the way to the upper right corner, which would mean the model makes predictions perfectly.

The accuracy of around 80% and F1-score around 0.8 are not perfect scores. Our baseline for the comparison was GPT4 classifier which achieved lower scores, of up to 60%. This difference can be caused by not focusing enough time in optimizing the GPT model. The only outlier was the arabic subtask where we could either get the precision high and recall low, or the opposite. It may be a language specific issue.

Accuracy for RoBERTa: 0.8075601374570447
Classification Report:

	precision	recall	f1-score	support
OBJ	0.83	0.89	0.86	194
SUBJ	0.75	0.64	0.69	97
accuracy			0.81	291
macro avg	0.79	0.77	0.77	291
weighted avg	0.80	0.81	0.80	291

Fig. 12. Best German results

Accuracy for RoBERTa: 0.8044943820224719
Classification Report:

	precision	recall	f1-score	support
OBJ	0.94	0.81	0.87	363
SUBJ	0.48	0.78	0.60	82
accuracy			0.80	445
macro avg	0.71	0.80	0.73	445
weighted avg	0.86	0.80	0.82	445

Fig. 13. Best Arabic results

Accuracy for RoBERTa: 0.765
Classification Report:

	precision	recall	f1-score	support
OBJ	0.79	0.72	0.75	200
SUBJ	0.74	0.81	0.78	200
accuracy			0.77	400
macro avg	0.77	0.77	0.76	400
weighted avg	0.77	0.77	0.76	400

Fig. 14. Best Multilingual results

The results show that the sentences contain clues on which the model can be trained and then predict subjectivity in a given sentence, although achieved accuracy is definitely not perfect. This could be caused by factors like too much variation in subjective sentences, causing underfitting in our model. We think that

In conclusion, both RoBERTa and ChatGPT-4 models demonstrated effectiveness for subjectivity classification, with RoBERTa exhibiting more consistent performance. However, ChatGPT-4 shows potential and may require further optimization. Parameter experimentation underscored the importance of tuning hyperparameters to optimize the model's performance.

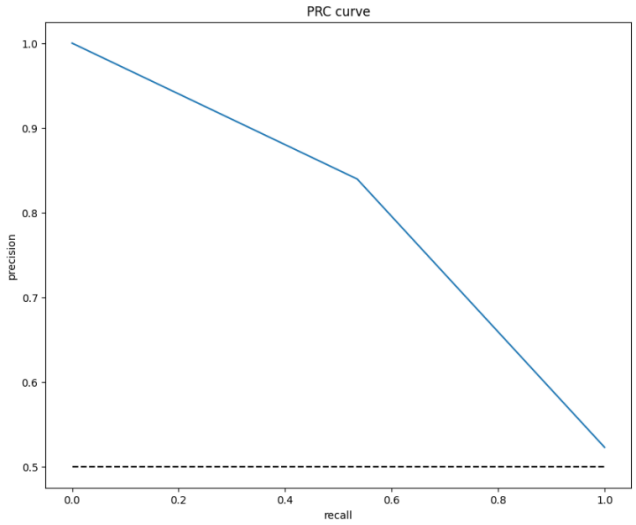

Fig. 15. PRC curve

8 CONTRIBUTIONS

In this section we name the responsibilities and task divided between the group members.

Jakub Mroz was responsible for:

- Researching the boilerplate code for training RoBERTa
- Creating code for training the model
- Experimenting with epochs and learning rate
- Analyzing datasets and results
- Report sections 1, 2, 3, 4, 5, 6, 7

Muhammad Fahad Nawaz Rana was responsible for:

- Parameter tuning
- Creating code for baseline comparison against GPT
- Report sections 1, 3, 5, 7

REFERENCES

A ONLINE RESOURCES

Github Repository of the project: https://github.com/JakubMroz4/subjectivity-detection