ID2223 Scalable Machine Learning and Deep Learning

Project Report

Joseph Attieh Pulasthi Harasgama

Introduction

The past decade has seen numerous waves of technological advancements that can be noticed by the significant increase in the use of smart devices and the internet. Unfortunately, some people have been exploiting the anonymity provided by the internet, as more people are being victims of cyberbullying and verbal abuse. To solve this issue, a lot of researchers have worked on Profanity Detection. Since this is a worldwide problem immune to language and cultural differences, it is crucial to be able to scale these systems to support multiple languages. In this project, we propose to explore Multilingual Profanity Detection through two solutions: the first solution relies on multilingual embeddings and a classification layer to perform the classification. The second solution tries to perform Knowledge Distillation to transfer knowledge from a language to another.

Datasets

To perform Multilingual Profanity Detection, we need datasets that provide textual texts and whether this text exhibits some hate/offense in it. To do so, we looked for datasets with binary labels (hate/no hate). We should note that some of these datasets were labelled by multiple experts and had more than one hate label. In this case, we labelled them through a voting process. Since we wanted to explore multilingual profanity detection, we procured datasets in different languages. Most of the datasets have distinct training and testing sets. In case the dataset has no clear split, we performed stratified splitting of 70/30 (to make sure that we have an equal amount of hateful and non hateful text in each of the sets). Furthermore, we have noticed an extreme imbalance between hateful and non-hateful data. Therefore, we performed oversampling/undersampling to try to solve this issue. The datasets procured are shown in the table in Appendix 1.

Methods

To perform multilingual profanity detection, we propose the following two solutions:

- Solution 1: This solution proposes to use a multilingual neural network-based model that is trained to generate multilingual representations. Two of the most famous approaches are mBERT [12] and XLM-RoBERTa [13]. mBERT is a BERT model that is trained on a text of multiple languages from Wikipedia. It handles the language imbalance problem of Wikipedia by oversampling small languages and undersampling large languages during training. XLM-RoBERTa is a RoBERTa model trained on a big multilingual dataset.
  The solution proposed is to fine-tune one/multiple models presented above on the task of Profanity Detection and evaluate it in different languages.

Therefore, we experimented with the following architectures:

- Model 1: This model is a mBert model with a classification layer of 2 neurons (it accepts the output of the CLS token of the BERT model, as it englobes some context). We also used a dropout of 0.1. The activation function is the Softmax function and the loss used is the cross entropy loss. To train the model, we used the AdamW optimizer with a loss of 2e-5. At each iteration, we split the training set into training and validation sets (70/30). Furthermore, we used the linear warmup scheduler for the learning rate. This model was implemented using mBert from Hugging Face and adding the layer using PyTorch.
- Model 2: This model is a XLM-R model with two neurons. It uses the same setup as the previous model (the only difference is that the first model's last layer was implemented by us, while this model is `XLMRobertaForSequenceClassification`.

- Solution 2: This solution proposes to perform Knowledge Distillation for Profanity Detection. Knowledge distillation will allow us to transfer the knowledge that we have learned from one language to another one by allowing a student model to learn from a teacher model. The teacher model is a Profanity Detector that performs well on a certain language. The student model is a model that is trained to perform profanity detection for a language that is different from the language used in the teacher model. This model tries to learn how to perform Profanity Detection by relying on a parallel corpus/dictionary/knowledge source that maps the language of the teacher to the language of the student, and by optimizing a loss function between its output and the output of the teacher model (ground truth).

  We experiment with the simplest form of knowledge distillation, by using the output of the last layer, instead of using the output of an embedding/inner layer. This requires us to have a parallel corpus to help us transfer the knowledge. We choose to distill the knowledge from French to Italian (the choice was made since the models of Solution 1 performed well in French, which can be seen in Experiments 1 and 2). Therefore, we proceed by translating the Italian dataset IHSC to French. This was done by iterating the file and calling the Google Cloud Translation API to get the translation for each input. We used Google Translate as we could not find datasets with parallel corpora for profanity detection. We chose not to experiment with other languages due to time restrictions.

  The setup is as follows:

  - Teacher model: the teacher model used is Model 1 that was fine tuned on all training sets (Experiment 2). The teacher model receives a text in French and outputs two values corresponding to Profanity / No Profanity.
  - Student model: the student model is also of Model 1 but is not fine tuned at all. It receives an Italian text and should predict values that are similar to the French version of this text fed to the teacher model.
  - The training is performed as follows: First, the French version of the text is fed to the teacher model. The output of the model is the ground truth. Second, we compute the output of the student model on the French and Italian texts. Then, we compute the mean square error loss between the outcome of each of these texts with the outcome of the teacher model. We add the loss and we update the

weights of the student model. We train for 20 epochs and with the same optimizers and scheduler as Solution 1.

Experiments and Results

Experiment 1

We proceed by fine tuning Model 1 on the training set of a selected dataset from a certain language. Then, we compute the accuracy and F-score measures on all test sets that we have. This would give us an idea on whether learning profanity detection in one language can help profanity detection in another language. The results of experiment 1 can be visualized in the notebook. We present the F-scores of the models that we have trained in Appendix 1 (the results of the accuracy were quite the same and do not give as much insight as the F-score).
Looking at the results, we can observe the following:

- Fine tuning on the Arabic and Turkish training sets result in relatively acceptable detection for both languages (F-score> 0.6), but does not result in good profanity detection for the other languages (F-score <0.2).
- Fine tuning on the German and Danish datasets is not good for profanity detection for either languages (F-score ~0.4-0.5), or for the other languages (F-score <0.15).
- Fine tuning on the first Spanish dataset is resulting in an F-score of 0.6 for the dataset. The second Spanish dataset performs very poorly on all sets.
- Fine tuning on the Portuguese dataset produces an F-score of 0.5 on the test set of this dataset and F-scores revolving around 0.4 for Arabic, French and Spanish. Fine tuning on Italian produces an F-score of 0.4 for the Italian test set, and F-scores revolving around 0.3 for Arabic and Turkish. Furthermore, fine tuning on the French dataset produces an F-score higher than 0.8 for the French test set, and F-scores that are consistent for all datasets (around 0.3). That would mean that learning profanity detection for one language also helps other languages. One can also attribute this observation to the fact that profanity in some languages is quite syntactically similar (for example, Portuguese is similar to French and Spanish, etc.).

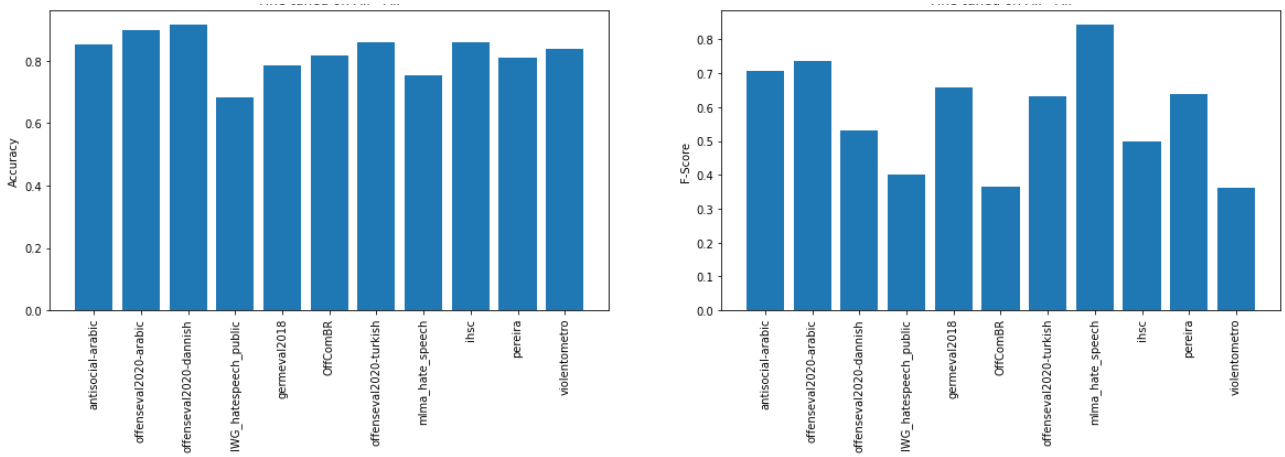
We can clearly see that the performance is quite dependent on the dataset. We can notice that fine tuning on a certain training set is resulting in a peak of F-score on the testing set of this dataset. We can infer that different versions of Model 1 were able to generalize on unseen data from the same domain and distribution, but failed to generalize with data from different domains (language) and different distributions (dataset). Furthermore, this can be attributed to the degree of profanity that varies between different datasets (some datasets exhibit minor hate text while others are more major).
To improve the results, we could train on more languages/datasets, so that the model could capture the generalizable/true knowledge needed for profanity detection. Since we are using mBert, this will help capture the profanity detection jointly with all languages (since the layer is finetuned on all languages instead of only one). This is tested in Experiment 2. To improve the results, we could also explore multiple methods to prevent overfitting to the local language/profanity artifacts, such as increasing the dropout or making the architecture more

complex (instead of a linear layer, we could use an LSTM layer for example). This would help the model generalize better.

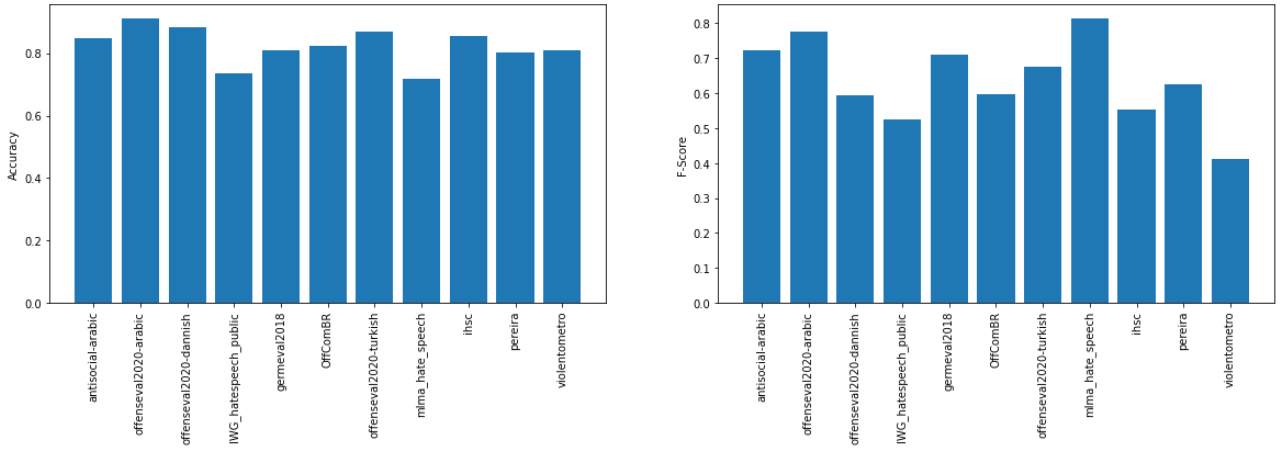
Experiment 2

We proceed by fine tuning both Model 1 and Model 2 on all the training sets of all languages. Then, we test the model on all testing sets. The results of the second experiment are shown below:

mBert + classification layer fine tuned on all training sets



XLM-R + classification layer fine tuned on all training sets

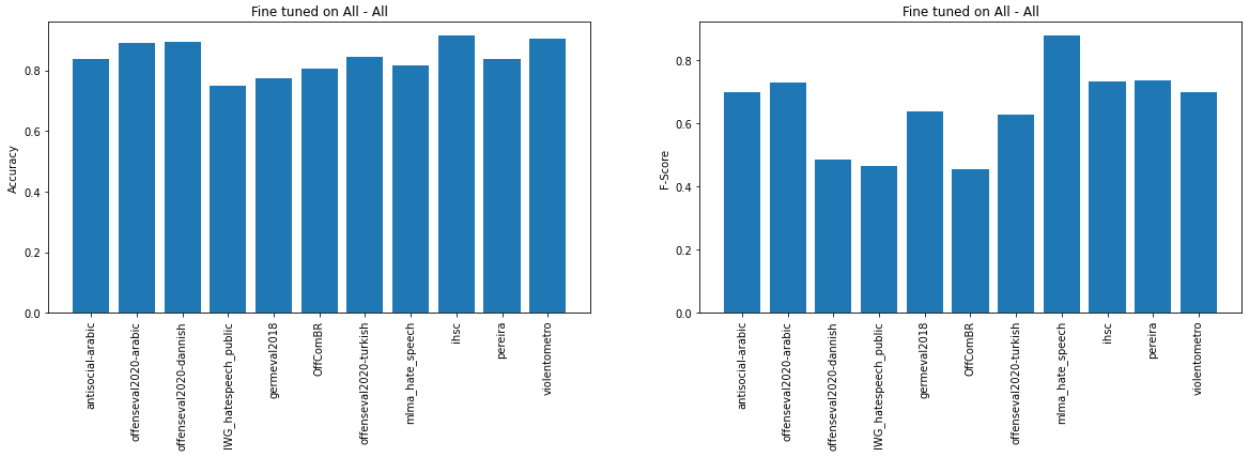


As we can observe, fine-tuning the models on all datasets resulted in much better scores compared to Experiment 1 for all the languages. This can be clearly attributed to the fact that we included more training pairs from different languages, which help generalize the profanity detector to multiple languages (the knowledge captured is generalized over different domains/distributions). Both mBert and XLM-R have similar trends in both accuracy and F-scores. This makes sense as we are using the same datasets and the same training procedure. A better approach would be to sample identically from each language so that the training set is more balanced in terms of different languages (it is balanced in training/testing set distribution, but not between different sets). However, we can notice that the XLM-R performs slightly better, as expected (as XLM-R shows better results in general than mBERT).

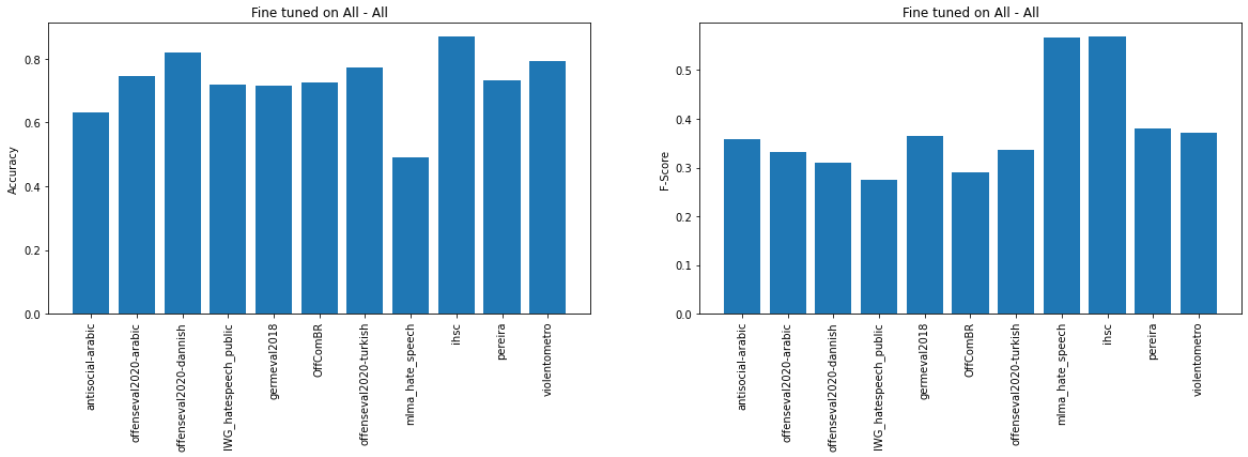
Experiment 3

We perform the knowledge distillation (from French to Italian) and we compute the accuracy and F-score on all testing sets. We perform the experiment with the mBert model from Experiment 2 as the teacher model. As for the dataset used, we employed the dataset IHSC and its translated version in French (translation was done using Google Translate). We are trying to teach a Model 1 (not fine tuned on any dataset) how to perform profanity checking through knowledge distillation. Results are shown below:

Teacher model (untrained)



Student model trained



As we can see, the model was able to learn how to perform profanity detection from the teacher model by relying on the knowledge distillation to the Italian language. We chose to transfer the knowledge from French to Italian since the teacher model was performing best for French as seen in the graph. We should mention that increasing the number of epochs for training would have resulted in better results.

Looking at the outcomes, we can see that the student model learned a distilled knowledge for the Italian language (since it was partly minimizing the loss between the French teacher outcome and the target Italian outcome). As we can notice, the student has learned knowledge that is similar to the teacher, as the F-scores of both models follow the same trend, with the difference that the teacher model has higher values. This can be explained as follow:

- Firstly, the student model was trained on machine translated language, which might be quite different from what is in the French testing set and which might not be similar to human language. The machine translation model offered by Google might not be of the same domain as the test set we have, and might be biased (might translate the data without maintaining the same degree of profanity). This can explain why the improvement for the Italian language was not as drastic as expected.
- Secondly, we can attribute the difference of performance to the fact that knowledge distillation on the last layer might not be the best solution for us since we have two neurons. We could use other approaches that rely on distilling hidden layers.
- Thirdly, we can see that Italian and French have very close F-scores on the test sets. This is clearly a result of the joint loss function that minimizes two MSE scores. Better results and a better optimization would have happened if we trained over more epochs (we only performed the training with 20 epochs). Furthermore, we set the batch size to be 32 and the maximum length for the BERT model tokenizer to be 64 (this was done for all experiments). Different parameters would have resulted in better results.

Finally, we should note that the student model is much better than the mBert model that was fine-tuned for the italian dataset (Appendix 1). Furthermore, the score on other languages is better, which means that the knowledge captured from distillation is generalizable (to some extent) to other languages. The student model learned from the teacher model, but did not outperform it in our current setup.

How to run the code

The steps below can be followed to replicate the results of the project. All datasets, saved models can be accessed via this Google Drive link.

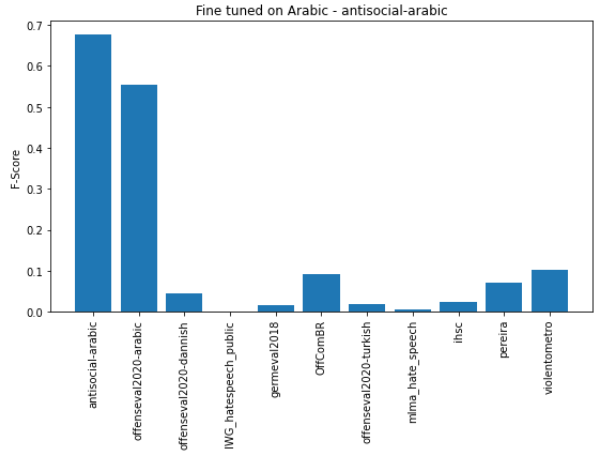
1. Import the attached notebook to Google Colab.
2. Add a shortcut to your Google drive from the shared folder by right-clicking on the folder name -> Add Shortcut to Drive.
3. Execute the notebook from the top.

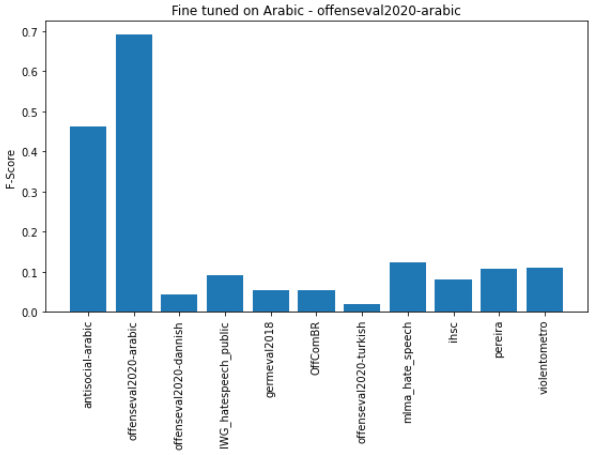
Appendix 1

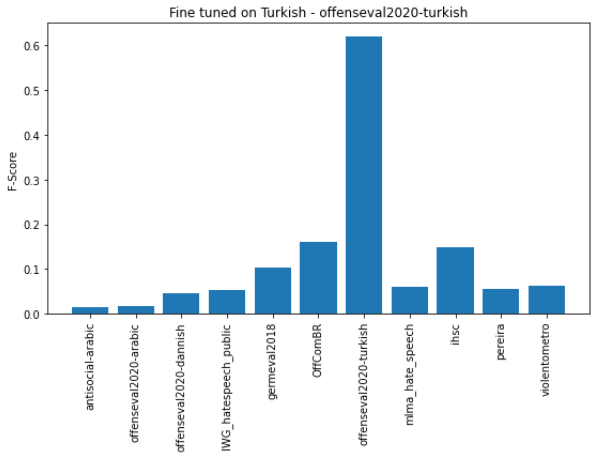
These are the datasets used:

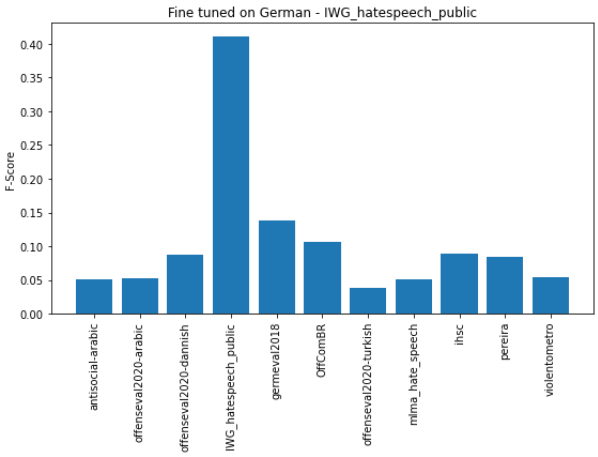
Language	Dataset	Size of Dataset	Percentage Abusiveness	Platform
Arabic	Antisocial-arabic [1]	11,268	0.25	Youtube
	Offenseval2020-arabic [2]	9,666	0.20	Twitter
Danish	Offenseval2020-danish [3]	3,289	0.13	Reddit/Facebook
French	Hate_speech_mlma [4]	4,014	0.80	Twitter
German	Germeval2018 [5]	5,009	0.34	Twitter
	IWG_hatespeech_public [6]	469	0.22	Twitter
Italian	IHSC [7]	5,221	0.16	Twitter
Portuguese	OffComBR [8]	1,033	0.20	g1.globo.com
Spanish	Pereira et al [9]	6,000	0.26	Twitter
	Violentometro-online [10]	1,959	0.15	Facebook
Turkish	Offenseval2020-turkish [11]	34,792	0.19	Twitter

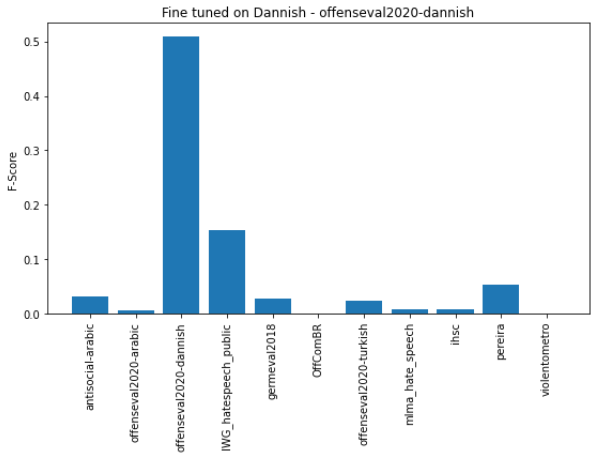
Appendix 2

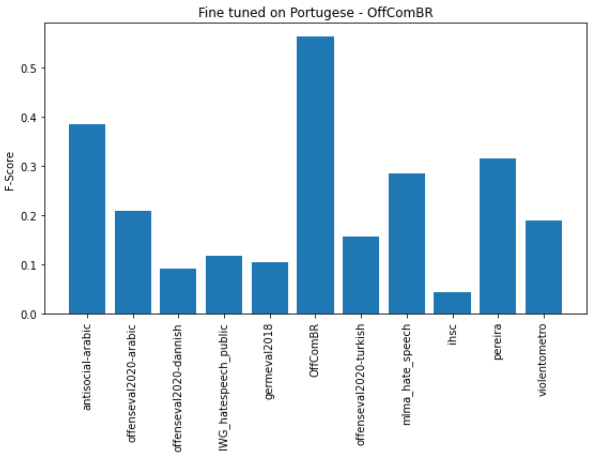

Fine tuned on Arabic - antisocial-arabic
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro

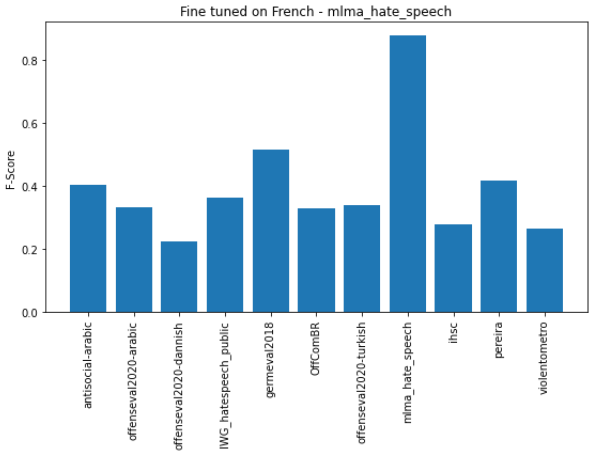

Fine tuned on Arabic - offenseval2020-arabic
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro

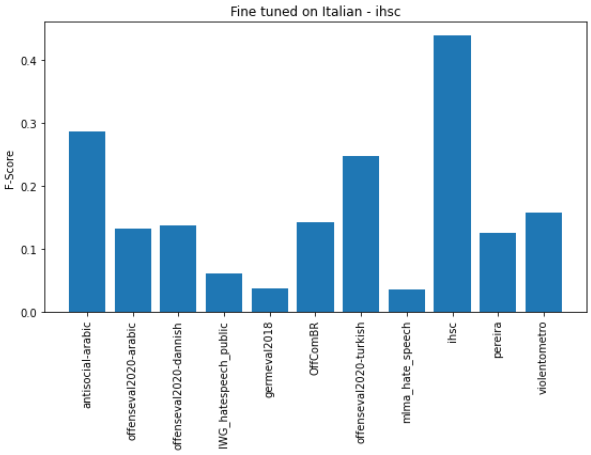

Fine tuned on Turkish - offenseval2020-turkish
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro

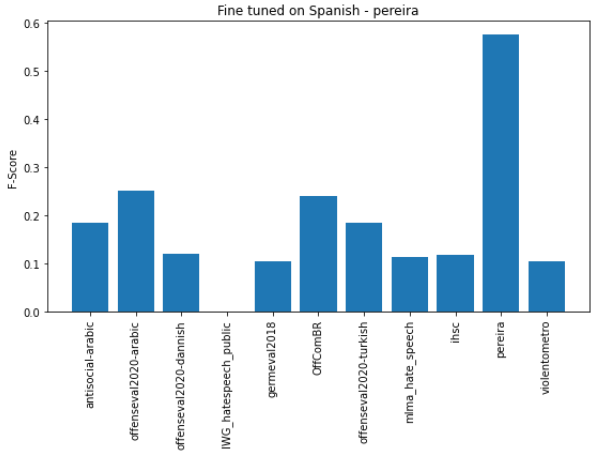

Fine tuned on German - IWG_hatespeech_public
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro

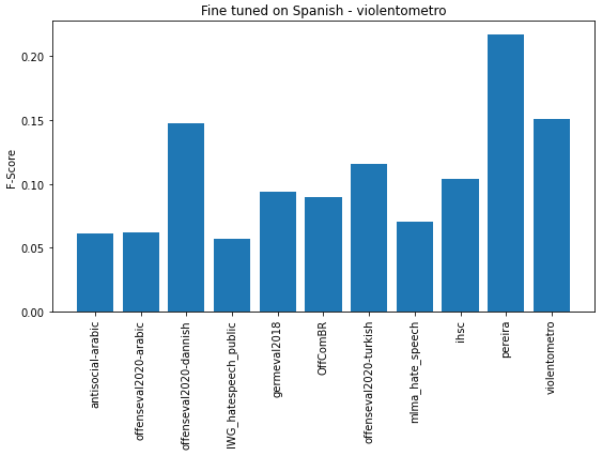

Fine tuned on Dannish - offenseval2020-dannish
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro


Fine tuned on Portugese - OffComBR
F-Score
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro


Fine tuned on French - mlma_hate_speech
F-Score
0.0
0.2
0.4
0.6
0.8
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro


Fine tuned on Italian - ihsc
F-Score
0.0
0.1
0.2
0.3
0.4
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro


Fine tuned on Spanish - pereira
F-Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro


Fine tuned on Spanish - violentometro
F-Score
0.00
0.05
0.10
0.15
0.20
antisocial-arabic
offenseval2020-arabic
offenseval2020-dannish
IWG_hatespeech_public
germeval2018
OffComBR
offenseval2020-turkish
mlma_hate_speech
ihsc
pereira
violentometro

References

[1] A. Alakrot, L. Murray and N. Nikolov, "Dataset Construction for the Detection of Anti-Social Behaviour in Online Communication in Arabic", Procedia Computer Science, vol. 142, pp. 174-181, 2018.

[2] H. Mubarak, A. Rashed, K. Darwish, Y. Samih and A. Abdelali, "Arabic Offensive Language on Twitter: Analysis and Experiments", arXiv preprint arXiv:2004.02192, 2020.

[3] G. Sigurbergsson and L. Derczynski, "Offensive Language and Hate Speech Detection for Danish", Proceedings of the 12th Language Resources and Evaluation Conference, 2020.

[4] N. Ousidhoum, Z. Lin, H. Zhang, Y. Song and D. Yeung, "Multilingual and Multi-Aspect Hate Speech Analysis", Proceedings of EMNLP, 2019.

[5] M. Wiegand, M. Siegel and J. Ruppenhofer, "Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language", in GermEval 2018 Shared Task on the Identification of Offensive Language, Vienna, Austria, 2018.

[6] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky and M. Wojatzki. “Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis”. 2017.

[7] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti and M. Stranisci. “An Italian Twitter Corpus of Hate Speech against Immigrants”. in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan: European Language Resources Association (ELRA). 2018.

[8] R. de Pelle and V. Moreira. “Offensive Comments in the Brazilian Web: A Dataset and Baseline Results”. in: VI Brazilian Workshop on Social Network Analysis and Mining. SBC. 2017.

[9] J. Pereira Kohatsu, L. Quijano-Sanchez, F. Liberatore and M. Camacho-Collados, "HaterNet a system for detecting and analyzing hate speech in Twitter", 2019.

[10] violentometro-online-team, "violentometro-online", GitHub, 2021. [Online]. Available: https://github.com/violentometro-online-team/violentometro-online.

[11] C. Çöltekin. “A Corpus of Turkish Offensive Language on Social Media”. In: Proceedings of the 12th International Conference on Language Resources and Evaluation. 2020

[12] J. Devlin, M. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[13] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” 2020.