University *of Ljubljana*
Faculty *of Computer and Information Science*

Cross-lingual offensive language identification

Maj Šavli, Blaž Rupnik and Leon Premk

Abstract

Rise in social media usage on the internet in past years has also brought some negative consequences. Research shows that offensive language and cyberbullying is a growing issue on platforms like Facebook, Twitter etc. In this project we researched some approaches for creating models that can automatically detect offensive language and hate speech. These models can be split into two groups, those that use neural networks and traditional approaches which don't. For these we only did classification on english dataset and generated some features and then applied models on these features. Results for all models show that predicting using these approaches can be quite accurate but we have to note that used dataset was a bit imbalanced so we must take these results with a grain of salt. We also tried transfering models to slovene language using multilanguage deep learning models such as BERT and XLM. These models allow training on data in different languages as target language but as can be seen perform worse on cross language classification than using them on single language.

Keywords

nlp, machine learning, hate speech

Advisors: Slavko Žitnik

Introduction

Since the outburst of social media, freedom of speech has allowed anyone to share their opinion on internet. While that allows people to make a change, it can also have negative consequences. Offensive language or hate speech has become a constant on online forums [1]. Best definition of online hate speech we can use are hateful messages (posts on social platforms, comments on news articles) directed against an individual or a group of individuals based on their identity. Because of these messages the group can be viewed as undesirable which warrants hostility towards them.

That's why automatic offensive language detection is highly required task. Some solutions for hate speech detection already exist but most are for english language. In this project we tackled offensive language classification using traditional machine learning approaches and state-of-the-art models such as mBERT and XLM-R on english datasets ([2, 3, 4, 5]) and later transfered our models to Slovenian language and test them on Slovenian datasets.

Related work

With rapid growth of information on internet, automatic tools for detecting hate speech are in huge demand. Earlier implementations of offensive language detection were based on basic machine learning classifiers such as naive bayes and SVM. By increasing hardware capabilities in recent years deep learning methods became became the new state-of-the-art outperforming previous methods by large margin.

Pitenis et al. [6] used deep learning methods to detect offensive language in Greek Tweeter posts. In another work Rizwan et al. [7] proposed their Convolutional Neural Network n-gram to detect hate speech on dataset containing Roman Urdu tweets. Ranasinghe et al. [8] used different state-of-the-art natural language processing methods such as BERT and XLM to detect offensive language in Bengali, Hindi and Spanish social media posts. In OffensEval 2020 [9] competitors were detecting offensive language, categorizing it based on offense type and identifying toward whom offense was targeted. Datasets were in English, Arabic, Danish, Greek and Turkish language. Most teams used pre-trained Transformers such as BERT [10] and it's variations like RoBERTa [11], or AL-BERT [12]. Other Transformers, most notably GPT-2 [13], were also used for classification. Word embeddings were mostly done by BERT or RoBERTa and BERT's multilingual variant mBERT [10].

Methods

Data

English dataset consist of five datasets from different sources. They mainly consist of social media posts on Twitter [4], Reddit and Gab [3], from White supremacist forum [2] aswell as comments from Fox news [5]. The first Slovenian dataset was manually acquired from Facebook, 24UR, and some other sites. The second Slovene dataset was acquired from [14], which is a Slovenian Twitter hate speech dataset. In the following paragraphs we present the data more thoroughly.

Fox news comments dataset consists of 1528 annotated comments from Fox news website. Each of the comments is labeled as either non-hate or hate. The class distibution can be seen in the image 1.

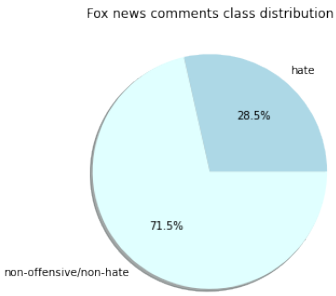

Figure 1. Fox news comments class distribution

Gab and reddit datasets consist of 21776 (Reddit) and 21774 (Gab) user comments. In this datasets, the comments are labeled as non-hate/hate speech, the same as the previous dataset. We can see that the class distributions, seen in the image 2 differ for the two datasets, Gab's being more balanced that Reddit's.

Twitter datasets can be divided into English and Slovenian one. English twitter dataset contains 24784 tweets and Slovenian dataset contains 17729 tweets. The two datasets are multi class meaning each tweet is labeled as one of the three possible labels. Both English and Slovenian tweets are labeled the same, hate/offensive/neither. Looking at the class distribution, seen in image 3, we can see the datasets are balanced very differently. Majority tweets in English dataset are labeled as offensive, where majority of tweets in the Slovenian dataset is labeled as non-offensive.

White supremacist forum dataset is another non-hate/hate dataset and contains 10834 comments from posts of white supremacist forum website. Looking at the class distribution, seen in image 4, we can quickly see the majority class, non-offensive, is larger than in previous datasets.

Slovenian test data is a small manually acquired dataset for testing the performance of transformer models and contains 96 comments. The class distribution is in this case, as seen in image 5 nicely balanced, with three classes: non-offensive/offensive/hate.

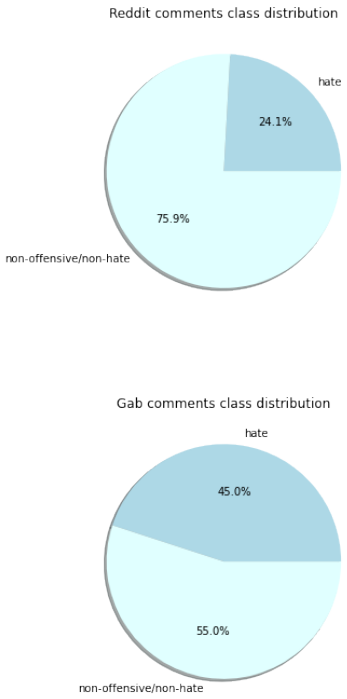

Figure 2. Reddit and Gab comments class distribution

Data preprocessing

In many cases, especially in text classification or translation text prepocessing can improve the accuracy of algorithms and models. Before input text, presented in natural form, can be passed to machine learning algorithms, it needs to be cleaned of unnecessary words. This way, the machine learning model can focus and learn on the words, which hold the useful information.

The first step was removing all the extra whitespaces and tabs, since they don't hold any information. Second step was changing all characters to lowercase characters, which helps the whole process of text processing and it can be beneficial in other ways, such as parsing. We want every word to be normalized and in original form and not in form of contractions, so the third step was to extend all the recognized contractions. The fourth step was to remove all special characters, since such characters add no value to text understanding and are considered as noise. Removing numbers can sometimes be useful, sometimes not. It depends on what problem you are solving. Since we deal with classification of hateful and offensive language we decided that numbers do not play a major role and can be removed as well. The last and a very import step was stemming. Stemming is process of reducing every word to its root. This is done by removing unneccessary characers, most often a suffix. The stemmers, which we

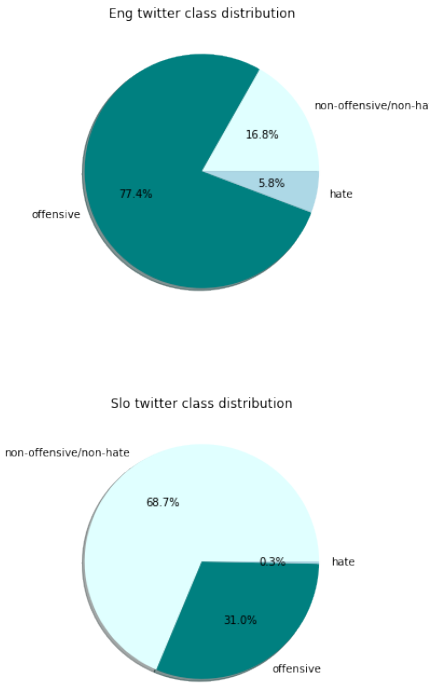

Figure 3. Twitter class distribution

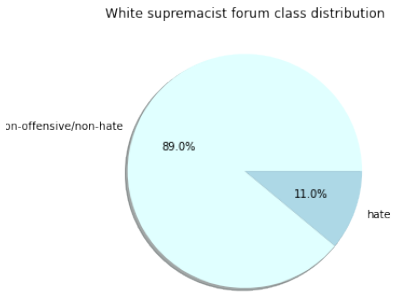

Figure 4. White supremacist forum class distribution

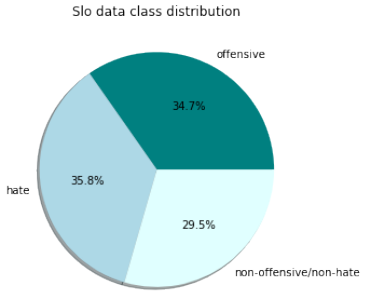

Figure 5. Slo test data class distribution

used, are two of the most known stemming models, Porter and Snowball. The models, however, have flaws and do not stem every word correctly and removing or changing information from text.

Classification using traditional approaches

For classifying offensive language on english datasets we also tried using a few traditional approaches, meaning neural networks are not involved, so statistical models like logistic regression, support vector machine and random forest which constructs multitude of decision trees that are used for classification. In two subsections we first describe how we constructed features from specific dataset and then we give some results for classification of offensive language.

Extracting features

We seperated extracting features into three different steps. First one was defining sentiment score for each sample in the data. For this we used a list of known english hate words, which we retrieved from Hatebase repository [15]. So for each sample we calculated what percentage of words are hate words from this hate words list.

Second step was extracting bigrams from the whole dataset and adding each bigram as a seperate column and defining binary value, meaning value is yes if sample has correlated bigram else no. Bigram is a sequence of two adjacent words. Since number of columns would be huge if used all bigrams we decided to set lower limit to 0.5% which means that we ignore terms that appear in less than 0.5% samples. Before performing this we also stemmed the data since many charged words can have different derivations.

Last set of features we produced are so called tf-idf features. Their purpose is to reflect how important a word is in the whole dataset. We extracted 50 terms with the biggest tf-idf weight and again used them as a seperate columns where values tell us if given sample contains correlated term. Weight of a term is determined by term frequency (tf) that tells us how many times a term occurs in one particular sample and inverse document frequency (idf) which tells us in how many samples the terms appears. When we performed this on our Twitter dataset we found out that big number of the 50 terms are slurs.

Concrete examples of these features can be seen in table 1.

Table 1. Used feature names and examples in our preprocessed dataset

Feature	Feature examples
Sentiment score	0.66 (means two thirds of words from the tweet can also be found in the hate words list)
Bigram	”act like”, ”ass bi*ch” (two most common bigrams in Twitter dataset - 2 adjacent words)
TF-IDF	”act”, ”always” (two terms with biggest weight - occur many times within small number of tweets, we selected 50 terms with biggest weight)

Classification

Before starting classification we created one feature dataset that consists of all three feature subsets described in the previous section. For evaluation of the models we used k-fold validation. Specifically we used 5 folds and for each metric that we observe we then calculated the mean and used it in the results table.

In table 2 we can see results for four different models in classifying if tweet is offensive in the given Twitter dataset. As the last one we also added the dummy classifier which always picks the majority class in the dataset so that it can be used as baseline for evaluating performance of the other models.

Next we also trained models for predicting three possible classes. In this case we didn't use fold validation due to longetivity. Here we splitted the data into train and test with fixed random seed so the results can be easily replicated. Evaluation results can be seen in table 3. We explain results more in next section.

Results

When evaluating results we can use the dummy model as our minimum that we try to improve. For binary classification of offensive tweets we can see that all models used had accuracy of over 90 percent. Here we have to note that classes are imbalanced and because of this we also included the F1 macro score. This metric also accounts for aggregation of class volume which is why the scores are lower than for regular F score.

As for the best performing model we could select eXtreme Gradient Boosting by a small margin. Extreme here means that this type of boosting uses more advanced types of regularization than regular gradient boosting, so consequently model generalization is better.

Since here we made binary predictions we decided to also visualize ROC curve 6 for all of used models. Dummy is here of an outlier due to its definition but we can see that all other models have very similiar sensitivity and have good performance.

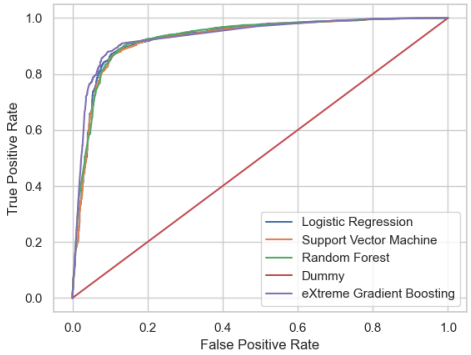

Figure 6. ROC curve for binary classification of all traditional models used.

We also performed multiclass prediction on the same set with the same models. As expected the evaluation scores here are lower than for the binary classification. Performance was again similiar for all models except dummy with extreme gradient boosting being slightly better than the others.

Classification with BERT and XLM

Since deep neural networks are curently achieving best results in machine learning we decided to use BERT and XLM for our task. We tried two variations of BERT. First being multilanguage pre-trained model mBERT, that is trained for 104 languages and second CroSloEngual BERT [16] that is trained on Croatian, Slovenian, and English corpora. For XLM we used XLM-RoBERTa that was said to outperform mBERT on a variety of cross-lingual benchmarks. Our classification head consists of three convolutional layers followed by one hidden layer with 128 parameters and final output layer with softmax activation function.

Model training

Our models were trained with couple different settings. In first scenario we trained our models on English data and tested them on English data. This approach produced best results, since corporas of all three models consist primarily of English language. Then we trained our models on both English and Slovenian data and tested them on test set that combines both languages. This setting produced slightly worse results, but was still much better than last two settings. In the last two settings we primarily targeted classification of Slovenian texts, that's why our test data consisted only of Slovenian texts. This texts were hand picked to balance out distribution of classes. Since classes in training data were heavily unbalanced, we decided to trim down majority class, that represented more than 68% of all data.

Results

Table 4 shows how models trained on English data performed on classification task with English texts. We can see that

Table 2. Classification results on English Twitter dataset when classifying if tweet is offensive

Model	Precision	Recall	F score	F1 macro score
Logistic Regression	0.93	0.93	0.93	0.84
Support Vector Machine	0.93	0.93	0.93	0.84
eXtreme Gradient Boost	0.96	0.90	0.93	0.86
Random Forest	0.93	0.93	0.93	0.85
Dummy	0.77	1.00	0.87	0.44

Table 3. Classification results on English Twitter dataset when classifying all three classes

Model	Precision	Recall	F score	F1 macro score
Logistic Regression	0.88	0.89	0.88	0.68
Support Vector Machine	0.87	0.89	0.87	0.65
eXtreme Gradient Boost	0.89	0.88	0.88	0.70
Random Forest	0.87	0.88	0.87	0.66
Dummy	0.60	0.77	0.68	0.29

models were able to classify all classes quite accurately. We can also see that both BERT models outperformed XLM. Combining models didn't prove to be an effective way to increase classification performance. Performance of mixed training and test sets is shown in table 5. We can see that accuracy is a bit worse than previous setting. Combining all three models into an ensemble improved performance by a few percent. When testing this model on Slovenian data only, performance dropped even more as can be seen in table 6. Models were able to detect two out of three classes, since class 2 is not frequent in Slovenian data. Best performing model was mBERT, outperforming all other models in all views except class 1 recall, where it was beaten by XLMr, meaning that XLMr predicted correctly highest percentage of all examples labeled with class 1. Performance of models trained on Slovenian data only is shown in table 7. Results are somewhat better than training with both languages, especially class 1 recall and class 0 precision.

After testing our models on data with same origin as training data, we decided to test our data on hand-picked samples. Results are shown in tables 8 and 9. Models trained on both languages were able to detect class 1 much better. Models trained on Slovenian data only had much higher class 1 precision, meaning higher percentage of examples classified with class 1 actually belonged to class 1. On the other hand XLMr trained on mixed language dataset had highest class 1 recall, meaning it was able to classfy correctly higher percentage of class 1 examples. It was also the only model to detect class 2 in this test set, even though it's class 2 recall was pretty low.

Conclusion

Traditional approaches

As seen in results we got high accuracy on all classifiers in comparison to the baseline one where we simply select the class with highest frequency. F1 macro score is lower than other measures which is expected due to class distribution of the used dataset.

When checking tweets for which classification was a false positive (so tweet marked as offensive even though it wasn't) we discovered that many of these use some kind of a dialect so a lot of features are skewed because of this. We figure that a possible solution would be to preprocess data in a way that we also transform dialect words based on some kind of mapping or with a help of a dictionary.

Another pattern we saw at some of false positives is that they contain curse words but are not actually offensive. For fixing this we figure that we could improve our sentiment score feature that checks for amount of curse words in a tweet since we don't know the context of this curses. We could improve this with checking words adjacent to curses if they might have positive meaning.

Deep learning

Models trained on both languages seemed to performed slightly better, especially on hand-picked Slovenian test data, since it was able to detect two out of three classes. Models were also much more accurate on English test data, which makes sense since majority of pre trained models corpus consists of English texts. English is also much simpler to understand containing less dialects, which are a huge part of Slovenian language. Slovenian offensive language and hate speech also seems to be much milder than english counterpart, making it hard to detect. Even models trained and tested on Slovenian data performed at best on pair with models trained on both languages, which only proves the fact Slovenian offensive language and hate speech detection can already be a hard task for "intra-language" models let alone "cross-language" models.

References

[1] Konrad Rudnicki and Stefan Steiger. Online hate speech - introduction into motivational causes, effects and regulatory contexts. 08 2020.

[2] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white

Table 4. Classification results on English train data and English test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.87	0.86	0.94	0.71	0.93	0.90	0.63	0.90	0.92	0.67
XLMr	0.80	0.73	0.95	0.72	0.97	0.75	0.51	0.83	0.84	0.60
CSE-BERT	0.85	0.83	0.93	0.71	0.94	0.87	0.59	0.88	0.90	0.64
Ensemble	0.86	0.82	0.95	0.73	0.96	0.87	0.59	0.88	0.91	0.65

Table 5. Classification results on English + Slovenian train data and English + Slovenian test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.80	0.75	0.92	0.69	0.92	0.74	0.61	0.83	0.82	0.65
XLMr	0.62	0.70	0.73	0.34	0.46	0.79	0.61	0.56	0.76	0.44
CSE-BERT	0.79	0.73	0.91	0.71	0.93	0.71	0.59	0.82	0.80	0.64
Ensemble	0.80	0.75	0.92	0.71	0.94	0.73	0.62	0.84	0.82	0.66

Table 6. Classification results on English + Slovenian train data and Slovenian test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.62	0.61	0.66	0	0.88	0.29	0	0.72	0.40	0
XLMr	0.54	0.60	0.51	0	0.54	0.54	0	0.57	0.52	0
CSE-BERT	0.58	0.57	0.63	0	0.94	0.14	0	0.71	0.22	0
Ensemble	0.60	0.59	0.68	0	0.93	0.18	0	0.72	0.29	0

Table 7. Classification results on Slovenian train data and Slovenian test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.60	0.67	0.50	0	0.65	0.53	0	0.66	0.52	0
XLMr	0.64	0.69	0.57	0	0.73	0.52	0	0.71	0.54	0
CSE-BERT	0.61	0.68	0.52	0	0.65	0.55	0	0.67	0.53	0
Ensemble	0.65	0.70	0.58	0	0.73	0.54	0	0.72	0.56	0

Table 8. Classification results on English + Slovenian train data and hand-picked Slovenian test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.26	0.28	0.17	0	0.82	0.06	0	0.41	0.09	0
XLMr	0.40	0.35	0.44	0.6	0.68	0.49	0.09	0.46	0.46	0.15
CSE-BERT	0.31	0.31	0.25	0	1	0.03	0	0.47	0.05	0
Ensemble	0.31	0.30	0.33	0	1	0.03	0	0.47	0.06	0

Table 9. Classification results on Slovenian train data and hand-picked Slovenian test data

		Precision			Recall			F score		
Model	CA	0	1	2	0	1	2	0	1	2
mBERT	0.31	0.29	0.35	0	0.71	0.27	0	0.41	0.31	0
XLMr	0.33	0.31	0.50	0	0.96	0.12	0	0.47	0.20	0
CSE-BERT	0.30	0.28	0.35	0	0.75	0.21	0	0.41	0.26	0
Ensemble	0.32	0.29	0.46	0	0.86	0.18	0	0.44	0.26	0

supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[3] Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. A benchmark dataset for learning to intervene in online hate speech. *arXiv preprint arXiv:1909.04251*, 2019.

[4] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

[5] Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266, Varna, Bulgaria, September 2017. INCOMA Ltd.

[6] Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. Offensive language identification in greek. *arXiv preprint arXiv:2003.07459*, 2020.

[7] Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. Hate-speech and offensive language detection in roman urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522, 2020.

[8] Tharindu Ranasinghe and Marcos Zampieri. Multilingual offensive language identification with cross-lingual embeddings. *arXiv preprint arXiv:2010.05324*, 2020.

[9] Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[14] Petra Kralj Novak, Igor Mozetič, and Nikola Ljubešić. Slovenian twitter hate speech dataset IMSyPP-sl, 2021. Slovenian language resource repository CLARIN.SI.

[15] Hatebase. https://hatebase.org/.

[16] M. Ulčar and M. Robnik-Šikonja. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In P Sojka, I Kopeček, K Pala, and A Horák, editors, *Text, Speech, and Dialogue TSD 2020*, volume 12284 of *Lecture Notes in Computer Science*. Springer, 2020.