Advanced Natural Language Processing Project: Hate Speech Detection.

Group 15:
Carsten Gieshoff (i6258391),
Zhaolin Li (i6229455),
David Pomerenke (i6254308)

March 23, 2021

Abstract

We improve recent work on the automatic detection of hate speech on Twitter. Using the approach and data from Davidson, Warmsley, et al. 2017[1] as a baseline, we suggest small improvements with regards to tokenization and weighting. By implementing a BERT and BERTweet model for hate speech detection, we improve the overall accuracy from 84% to 91%, and the F1-score for the hate speech class from 44% to 48%. Without using hate speech related training data from these languages, we transfer our model to Italian, Spanish, and Turkish, and evaluate the approach on a multilingual toxic comment dataset.[2]

[1] Cf. https://github.com/t-davidson/hate-speech-and-offensive-language

[2] You can find our code repository at https://github.com/carstengieshoff/HS-Detection-Project.

1 Introduction

Social media platforms such as facebook and twitter have had a lot of success since their establishment 15 years ago. The ease with which humans can stay in contact, share their ideas and find people with common interests regardless of their location has led to huge amounts of textual and visual information on these platforms.

A negative side effect with this huge amount of information is the distribution of hateful and harmful content targeted at individuals, ethnic minorities and other groups. The sheer amount of tweets that are produced on a daily basis makes it an impossible tasks to identify and filter out these tweets manually. Consequently, the task of automatic detection of harmful or hateful content has obtained a lot of attention recently.

Work in this field faces several challenges. There are several different definitions of what constitutes hate speech, caused by human biases and personal sentiments. Another problem is that whether a utterance is considered to be hate speech might depend on (non-available) meta data such as who the author and who the addressee is. Moreover, annotated data is expensive to obtain and is also biased by the respective understanding of hate speech of the annotators. We delimit our work to the scope of Davidson, Warmsley, et al. 2017, who try to more robustly distinguish between hate speech, offensive language (e.g. lyrics containing slur terms) and not offensive language at all. Thus, we do not investigate different definitions of hate speech, but try to improve on the classifiers used in Davidson, Warmsley, et al. 2017. To that end, the classifier used in Davidson, Warmsley, et al. will be analyzed with the authors own criticism of it, and with a closer analysis of their code. Based on this analysis and identified shortcomings new methods will be proposed and explored. Among these methods the best ones will be compared in terms of the accuracy, precision and hate speech specific F1-score. This way, we both evaluate overall performance of the considered classifiers and evaluate their performance on the most important class: hate speech.

Further, the current work has been focused on monolingual hate speech detection, mostly English. This project aims to develop an approach for multilingual hate speech detection. As providing annotated training data for in multiple languages is not practical, we focuses on multilingual detection with a training dataset only in English.

The following research questions will be addressed during the project:

- How to increase robustness in the detection of hate speech?
- Can we expand the model towards multilingual detection without annotated data in the second language?

2 Related Work

2.1 Hate Speech Detection

Automatic hate speech detection is a form on text classification, a very active field of research with many contributions. The development of this research branch is closely related to the overall advances in Natural Language Processing (NLP) in recent years, which is mainly driven by Deep Learning, e.g. Devlin et al. 2018. However, classical Machine Learning approaches have been very successful in this regime, too Ikonomakis et al. 2005.

Schmidt and Wiegand 2017 provides a survey over common approaches in automatic hate speech detection and related tasks. They highlight the potency of machine learning classifiers build on simple lexical features such as bag of words. They also summarize that recent work built on character n-grams has proven to be more successful than token n-grams, arguing that this might be caused by the nature of the task specific data, which is prone to miss-spelling that occurs often on social media platforms. It should be noted though, that this paper does not include novel approaches like transformers Vaswani et al. 2017 that have shaped the course of application an research in NLP significantly in the last years.
A more novel summary which also takes into account neural approaches is given in MacAvaney et al. 2019. In this paper the authors propose an interpretable mSVM model that tries to be more efficient on detecting hate speech based on subtle combinations of features, which might not be detected by classical regularized classifiers. Moreover, the authors provide a benchmark BERT model showing the competitiveness of more classical approaches.

Our work is motivated by the findings and data provided in Davidson, Warmsley, et al. 2017. In this work, the authors build a data set of tweets, which are manually labelled int three categories: 'hate speech', 'offensive language' and 'neither'. The authors emphasise the importance to distinguish between the large amount of offensive language and actual hate speech. Davidson, Warmsley, et al. build and analyze a classifier on their data set. They also provide insights into the shortcomings of both their classifier and their manually annotated data.

Davidson, Bhattacharya, et al. 2019 investigate further the bias of hate speech classifiers as introduced by the data sets these classifiers are trained on. They investigate reveal datasets from this field by training and analysing classifiers on these datasets. They show that these biases remain even after accounting partially for demographic-dependent slang. This provides an important insight into the problems of hate speech detection and highlights the responsibility of developing such methods for deployment.

2.2 Multilingual detection

Using a single model to detect hate speech for multilingual case has been been a popular topic recently. Huang et al. 2020 published a multilingual Twitter corpus for the task of hate speech detection. Aluru et al. 2020 and Chiril et al. 2019 built multilingual models by fine-tuning deep learning models, such as BERT, with multilingual hate speech datasets. However, due to the lack of annotated data in terms of language and domain, model fine-tuning is not always a practical approach.

Multilingual Translation has gained a lot improvement recently. Tang et al. 2020 developed a multilingual model that translate directly between any pair of 50 languages with affordable accuracy. Benefit from that, translating the data from the source language into English then detect hate speech with a detection model trained with English data becomes a feasible solution for multilingual hate speech detection.

3 Approach

In this section we restate the model proposed in Davidson, Warmsley, et al., replicate the model and evaluate it on a test-set. Then, we address its shortcomings as stated by Davidson, Warmsley, et al. and possible improvements that were identified by inspecting their code.
We then propose several approaches that are aimed at improving on the identifies shortcomings of this model. The experimental results of these approaches are stated in Section 5.

3.1 The Current Model

This section shortly outlines the model proposed in Davidson, Warmsley, et al. 2017 for automated hate speech detection, states their results and re-summarizes the flaws of the model as stated by Davidson, Warmsley, et al.

3.1.1 Setup Current Model

In this section we shortly review the model used in Davidson, Warmsley, et al. 2017. The authors first create a range of features including tdidf-matrix, POS-tags (on a char basis), sentiment scores and Flesch-Kincaid Grade Level and Flesch Reading Ease scores to assess quality of each tweet. Then, the dimensionality of the data is reduced significantly by using a meta-transformer that only considers features that were relevant to L1-regularized Logistic regression. On these features then several machine learning classifiers such as Naive Bayes, SVM and Logistic Regression are trained and evaluated. Among those approaches L2-regularized Logistic Regression was the most successful. Their final evaluation is run on all the available data, not on a specific test set. We rerun their experiments with minimal necessary changes (different solvers in fitting models due to different software versions) and evaluate the final model on 10% of the data which was held out for testing alone. The confusion matrix evaluated only on this test data can be seen in Figure 1. Table 1 shows accuracy, avg. precision and the hate speech specific F1-score of the replicated model as evaluated on the test data.

	Replicated Model
Accuracy	0.84
Weighted Precision	0.85
HS F1-score	0.44

Table 1: Metrics of replicated model.

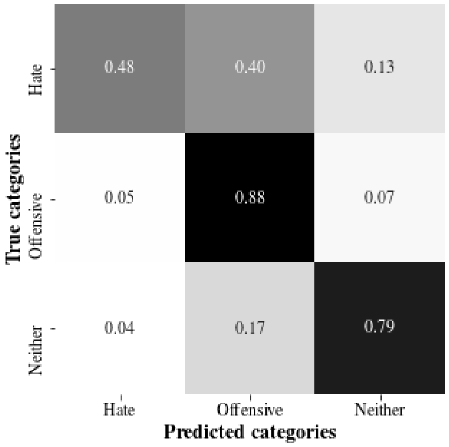


Figure 1: Confusion matrix of replicated model (Train-Test split used)
This can be replicated using the JupyterNotebook 'ReplicationWithTrainTestSplit.ipynb'. Different solvers were used Further parameter choices can also be deducted from the notebooks. Evaluation of the model was done on 10% held out test set. Further metrics can also be obtained from 'ReplicationWithTrainTestSplit.ipynb'.

3.1.2 Flaws of Current Model

Analyzing the model, the authors of Davidson, Warmsley, et al. 2017 claim the following properties of their model:

- Much misclassification of hate speech class: the '[...]model is biased towards classifying tweets as less hateful[...]' (see Results in Davidson, Warmsley, et al. 2017)
- Too much reliance on lexical features: Tweets that are most likely to be considered hate speech contain multiple slur terms from the lexicon or mention topics of relevance
- Types of racism that are less frequent in the training data tend to go undetected

The first problem might be addressed by putting more weight on hate speech instances in the training data. To tackle the heavy reliance on lexical features, more syntactic and contextual features might be used. To address the different types of racism, data augmentation might be used by replacing racist or homophobic slur terms by sexist ones and then applying a paraphrasing model. This last part unfortunately will not be addressed here.

3.2 New Approaches

3.2.1 Different Features

We explore the effects of choosing different features for the classical ML models chosen in Davidson, Warmsley, et al. 2017. Davidson, Warmsley, et al. 2017 use char based uni-, bi and trigram tokenization. We explore word based tokenization based on the tokenizers designed for tweets provided by the nltk library[3]. While both approaches have been used in related literature, word based tokenization might yield more useful POS-features compared to the current approach (c.f. code from Davidson, Warmsley, et al.). We use this approach trying to make the model pay more attention to syntax.

3.2.2 Different Machine Learning Models

The final model of Davidson, Warmsley, et al. 2017 uses logistic regression for classification. They have have searched for good parameters, and also tried out a support vector machine classifier, but not published these intermediate results. We replicate these intermediate results using their code. Additionally, we try out other machine learning models: A naive Bayes classifier, a stochastic gradient descent (SGD) learner for support vector machines, and an ensemble method that tries to combine the best of all the aforementioned approaches. We evaluate these models on three different features extracted from the tweet dataset: The original features generated by Davidson, Warmsley, et al. 2017, the selected features by the authors using logistic regression, and the features that we generate ourselves (see subsubsection 3.2.1).

3.2.3 Aggressive Hate Speech Weighting

The models used in Davidson, Warmsley, et al. 2017 use different weights for the different classes in their loss functions to account for the unbalanced dataset. We explore the behaviour of models that were trained on loss functions that put even more emphasis on the the hate-speech class. Let $n = n_1 + n_2 + n_3$ be the weight the size of the data set and the amount of instances from each class respectively. In the scope of Davidson, Warmsley, et al. 2017 the weights are chosen as $\omega_i = \frac{n}{3n_i}$ such that equal balance is used for all classes. Note that in this setting ω_0 the weight of the hate speech class is the largest. For different reasonable choices of E (e.g. $E = 1, 1.25, 1.5$) we use the weights

$$\tilde{\omega}_i := \frac{\omega_i^E}{\omega_0^E + \omega_1^E + \omega_2^E} \qquad \text{for } i = 0, 1, 2 \tag{1}$$

to put more emphasis on the hate speech tweets.

[3] https://www.nltk.org/

3.2.4 SOTA Approach: BERT-Classifier

Addressing the issues of syntactic consideration and addressee consideration in the model in Davidson, Warmsley, et al. 2017 we also implement a classifier built from a fully connected layer on top of a BERT-model. We used the AdamW optimizer from huggingface [4] to fine tune the existing BERT-model for 4 epochs.

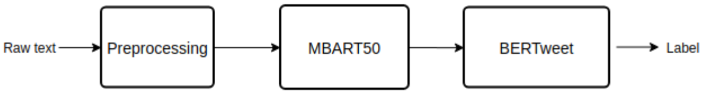
3.2.5 SOTA Approach: RoBERTa / BERTweet

In Section 3.2.4 we suggested a BERT-based classifier trying to build a classifier that might make use of syntax of tweets and that might be able to better grasp the small nuances that distinguish offensive language and hate speech. We aim to advance a little further in this direction by using BERTweet (Nguyen et al. 2020) instead of a classical BERT model.
BERTweet (Nguyen et al. 2020) has BERT architecture but was pre-trained in the more novel RoBERTa (Liu et al. 2019) fashion on a large corpus of tweets. As RoBERTa type models have shown a lot of success on downstream tasks and due to the tweet specific language model, this setup is very promising. (While BERT has been used in this field – see MacAvaney et al. 2019 –, BERTweet has not been applied yet to this task to the best of our knowledge. Architecture, training setup and hyper parameters have been chosen as in the BERT setup.

3.3 Multilingual detection

Multilingual hate speech detection consists of two steps: multilingual translation and hate speech detection. The workflow is shown in Figure 2. Multilingual translations work as translating the source language into English with the model MBART50 developed by Tang et al. 2020. The model has been proven to perform well at translating directly between any pair of 50 languages. Before feeding the MBART50, the raw tweets get processed by removing abnormal characters and symbols. With the translation, the pre-trained model BERTweet (see 3.2.5) is applied to classify hate speech.

Figure 2: Multilingual hate speech detection workflow



Besides, a state-of-the-art multilingual hate speech detection model, named detoxify, from Hanu and Unitary team 2020 is selected as reference. The detoxify

[4] https://huggingface.co/

model is fine-tuned from the XLM-RoBERTa model (Conneau et al. 2020) with the hate speech dataset covering 7 difference languages. Both the developed and the detoxify model will be evaluated with a dataset in Spanish, Italian and Turkish languages. These three languages are covered by the detoxify training dataset. The reason for selecting the detoxify is that it has SOAT performance in multilingual hate speech detection and has comparable performance to the BERTweet model.

4 Data

4.1 Hate speech detection

The data we use was created by Davidson, Warmsley, et al. 2017. This dataset is considered a major benchmark dataset by the survey on hate speech detection by MacAvaney et al. 2019. The authors scraped twitter, searching for tweets that contain keywords that are found in *HateBase*, a manually compiled dictionary of likely hate speech terms. These tweets were then manually classified by humans (usually three humans per tweet) on a crowdworking platform. The three categories were hate speech, (other) offensive language, and neither of them. The result is 24802 tweets, where for every tweet and category the number of humans that voted for this category is given. Majority rule is used to establish the final category for each tweet.

The authors highlight two major problems with the data set: First, only 5% of the data have been labeled as hate speech, so the size of the categories is very uneven. Second, after manually inspecting some exemplary labels made by the crowdworkers, the authors found that some of them are labeled badly. Particularly, the authors observe the following properties of the data:

- Mislabeling as hate speech due to occurrence of slur terms without hateful context.
- Mislabeling hate speech as offensive or neither because they contain rare slur, or because of subtle hate speech or not addressing someone directly.
- Racist and homophobic tweets are more likely to be considered as hate speech by annotators than sexist tweets.

These problems cannot be tackled in the scope of this work. Further improvements might be obtained by:

- Improving the labeling of the given dataset.
- Crawling new data.
- Using other related models (c.f. MacAvaney et al. 2019) to generate additional features.

4.2 Multilingual detection

The Multilingual Toxic Comment Classification dataset from Jigsaw 2020 is selected for multilingual hate speech detection. The dataset has 8000 instances consisting of comment text, language id and toxicity. The dataset consists of three languages: Turkish, Spanish and Italian, which has 3000, 2500, 2500 instances respectively. The whole dataset is used for evaluation as all models are pre-trained.

5 Experiments & Results

All our experiments can be replicated using our published code.[5]

5.1 Experiments using Different Features

As described in Section 3.2.1 we compared different tokenization options. To that end L2-regularized Logistic Regression following the setup from Section 3.1.1 was run using different tokenizers. While the original approach in Davidson, Warmsley, et al. 2017 uses char-based tokenization we used word-based tokenization using the TweetTokenizer from nltk library[6]. Note that slightly different parameters were used in this char-based approach compared to the original model (c.f. section 3.1.1). We also conducted this experiment using different maximal length for n-grams, and different frequency filters for tokens to be added to the vocabulary. As the results among different options in these regimes did show only small difference, we only state the experiment closest to the original setup from Davidson, Warmsley, et al. 2017 here.

Table 2 shows weighted avg. precision, accuracy and hate speech specific F1-score of the resulting classifiers. Notably, word-based performs slightly better than char-based, especially on the hate speech class of interest, as can be seen by the increased F1-score.

	char-based	word-based
accuracy	0.8721	0.8778
weighted percision	0.8841	0.8948
HS F1-score	0.42	0.526

Table 2: Char-based vs Word-based tokenization
This can be replicated using the JupyterNotebook 'ComparingTokenization.ipynb'. Here word-based is the 'casusal_std' that can be chosen in the notebook. Further parameter choices can also be deducted from this notebook. Evaluation of both classifiers was done on the same 10% held out test set.

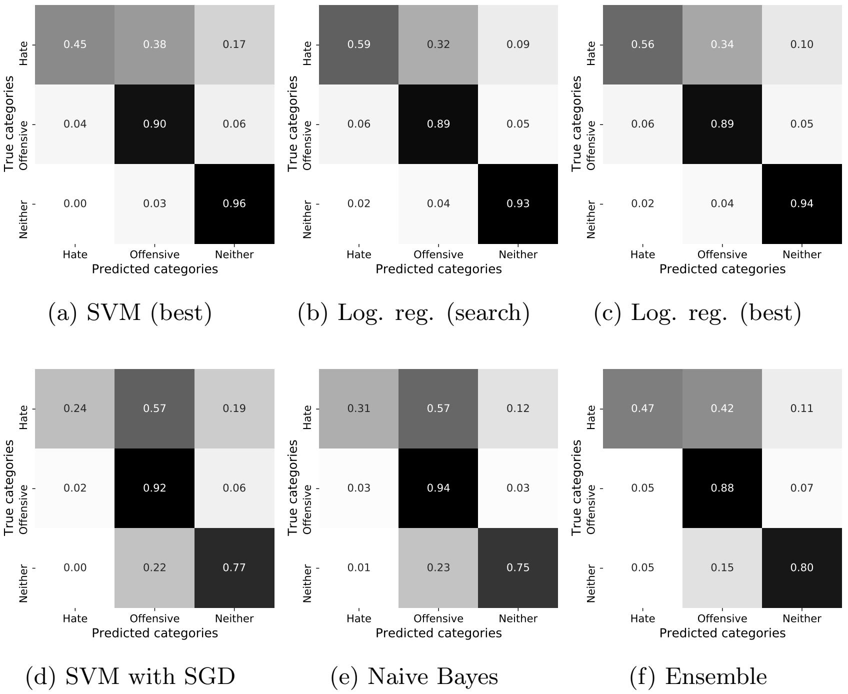
5.2 Experiments using different machine learning models

We report the results from these experiments in Figure 3. Here we interpret these figures.

Davidson, Warmsley, et al. 2017 have identified a best model for a support vector machine classifier and for a logistic regression classifier, respectively, and we can find the parameters in the code. We modify their code by applying train-test split. This results in lower accuracy. Moreover, their 'best' logistic

[5] https://github.com/carstengieshoff/HS-Detection-Project

[6] https://www.nltk.org/_modules/nltk/tokenize/casual.html



(a) SVM (best)

(b) Log. reg. (search)

(c) Log. reg. (best)

(d) SVM with SGD

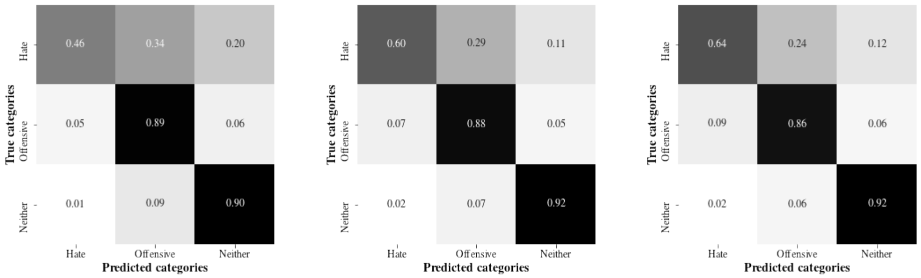
(e) Naive Bayes

(f) Ensemble

Figure 3: Confusion matrices for different machine learning models. Top row: Replication of their models, with replicated selected features, modified by applying train-test split. Bottom row: Additional machine learning models for comparison, with the same parameters. See subsection 5.2 for more information.

regression model (c) is now outperformed by a model (b) using different parameter settings from the grid search they use to find the best model. Their finding that logistic regression works better than the support vector machine can be verified.

Motivated by some initially found improvements with regards to the F1-score of the hate speech class, we also implement an SVM classifier that uses a stochastic gradient descent learner (d). Our final result, however, is that this method is actually worse than the simple support vector machine classifier by the authors. We also investigate a naive Bayes classifier (e), which is also inferior to the approaches by the authors. Finally, we combine all these approaches using an ensemble voting classifier, that combines the predictions made by the other classifiers.



(a) $E = 1$

(b) $E = 1.25$

(c) $E = 1.5$

Figure 4: Confusion matrices for Logistic regression with different weigths

	E=1	E=1.25	E=1.5
accuracy	0.8778	0.8645	0.8540
weighted precision	0.8948	0.8878	0.8884
HS F1-score	0.526	0.4876	0.4773

Table 3: Influence of weights on different metrics

This can be replicated using the JupyterNotebook 'AggressiveWeighting.ipynb'. Further parameter choices can also be deducted from this notebook. Evaluation of both all scenarios was done on the same 10% held out test set.

5.3 Experiments using Aggressive Weighting

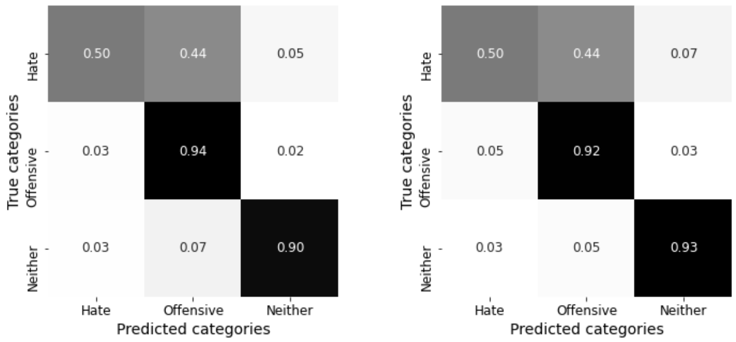
For our experiments on more aggressive weighting we used the setup from the best model (i.e. tokenization etc.) from Section 5.1. The training of the Logistic Regression were done with weights as in (1) for $E = 1$ (standard, same model as Section 5.1) and compared to $E = 1.25, 1.5$. We also tried using these weights for the data dimensionality reduction, but this proved harmful rather than helpful. Thus, new weights were only applied to the loss function for training the Logistic Regression.

Figure 4 and Table 3 show that more aggressive weighting has benign effects on the recall hate speech class, while it has negative effects the F1 score on this class and weakens classification on the other classes leading to reduced overall accuracy. Still this emphasis on recall of the hate speech class can have it uses in certain scenarios such as preliminary filtering.

5.4 BERT and BERTweet Experiments

As suggested in Sections 3.2.4 and 3.2.5 we implemented two state-of-the-art approaches for hate speech detection. Both consist of a fully connected layer on top of a BERT and BERTweet model respectively. Note that for these experiments less training data was used compared to the other approaches (see

Sections 5.1, 5.3), as a validation set was used. Thus, compared to previously a 90%-10% split this section deals with an 80%-10%-10% split. This should be taken into consideration when comparing these models to the other sections. Experiments were run for different amount of epochs (4, 6) and different batch sizes (16, 32).



(a) Best BERT model confusion matrix

(b) Best BERTweet model confusion matrix

Figure 5: Confusion matrices SOTA models

	BERTweet	BERT
accuracy	0.9	0.9104
weighted precision	0.9102	0.9133
HS F1-score	0.4342	0.4803

Table 4: Metrics of BERT and BERTweet model
This can be replicated using the JupyterNotebook 'BERTClassification.ipynb' and 'BERTweetClassification.ipynb'. BERT: exp=1, Epochs = 4, batch size = 16. BERTweet: exp=1, Epochs = 6, batch size = 32. Further parameter choices can also be deducted from the notebooks. Evaluation of both scenarios was done on the same 10% held out test set.

It should be noted that more intense fine tuning and more expertise in fine tuning of these models might achieve better results.

Table 4 shows that these state-of-the-art models outperform previous approaches while being effectively trained on less data. Unfortunately, their behaviour on the hate speech class (HS F1-score) is comparable to the model from Davidson, Warmsley, et al. 2017 but worse than the word-based logistig regression from Section 5.1.

5.5 Multilingual Detection

Two experiments have been done for multilingual detection: Baseline and Translation & Detection. The dataset in 4.2 has been used. The evaluation is carried out with the confusion metrics.

5.5.1 Detoxify

The pre-trained model Detoxify has been used in this experiment to provide a reference. As shown in 6, for all 8000 instance the Accuracy, Precision, F1-score are 0.89, 0.8 and 0.94 respectively. As 7 shows, the baseline performs better on Turkish than on Italian and Spanish.

Figure 6: Baseline result for all languages. 0 means non-toxic and 1 means toxic.

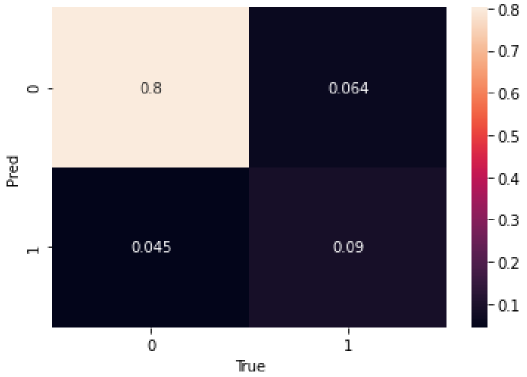
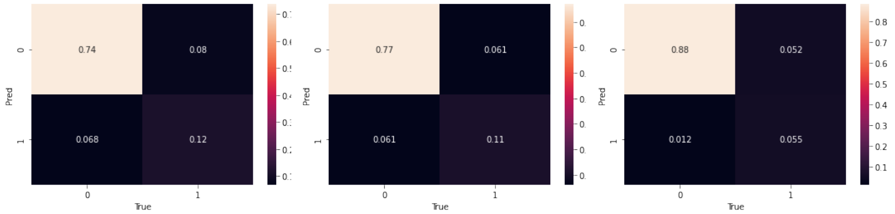


Figure 7: Baseline result for each languages. 0 means non-toxic and 1 means toxic. From left to right: Italian, Spanish and Turkish



5.5.2 Translation & Detection

In this experiment, the raw text gets translated with mBART50. The translation is then fed into BERTweet for hate speech classification. As 3.2.5 says, the BERTweet model has been trained to classify three labels: hate speech, offensive language and neither. In this experiment, hate speech and offensive language are both set as toxic and the neither is set as non-toxic. As shown in 8, the method gets Precision 0.83 which means it performs well at predicting non-toxic text. But the method reaches a high False Positive Rate 0.83 and a high Negative Predictive Value 0.58 which means it performs badly at recognizing toxic comment. The F1-score is 0.92. As 9 shows, there is not much difference between each language.

Figure 8: Multilingual detection results for all languages. 0 means non-toxic and 1 means toxic.

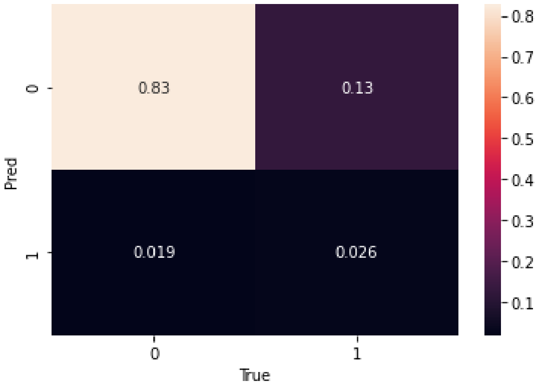
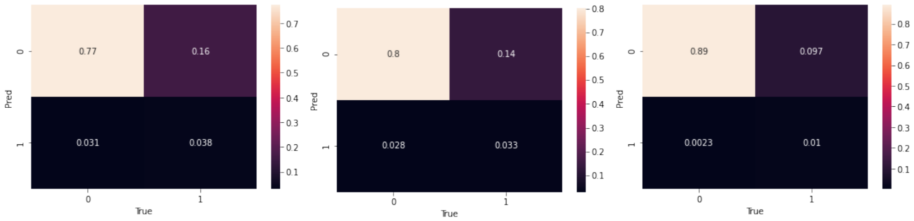


Figure 9: Multilingual detection results for each languages. 0 means non-toxic and 1 means toxic. From left to right: Italian, Spanish and Turkish



Compared with the training dataset of BERTweet, the dataset used in this experiment only has two label: toxic and non-toxic. The previous set both hate speech and offensive language as toxic, but the offensive language maybe not clearly classified in this dataset. Therefore, a follow-up experiment has been done by removing the results which are classified as offensive language. As shown in 10, the precision, False Positive Rate, Negative Predictive Value and F1-score are 0.86, 0.95, 0.45 and 0.93. The results indicates that removing the offensive results doesn't help. As 11 shows, there is not much difference between each language.

Figure 10: Multilingual detection results for all languages with no offensive data. 0 means non-toxic and 1 means toxic.

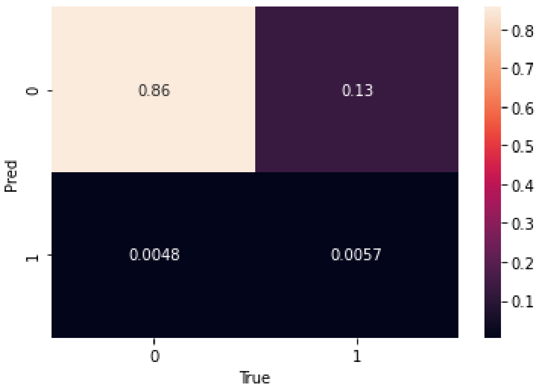
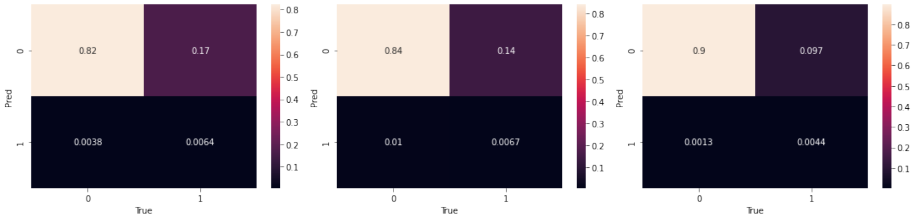


Figure 11: Multilingual detection results for each languages with no offensive data. 0 means non-toxic and 1 means toxic. From left to right: Italian, Spanish and Turkish



5.4 shows, the BERTweet reaches comparable result to the detoxify model. As the results of the above experiments, compared with a multilingual model training with language-specific dataset, the translation and detection methods have a worse overall performance. The hate speech attribute gets lost while translation thus the data is tend to be classified as non-toxic. These experiment indicates that solving multilingual problem by directly translating the source data would loss the problem-related attribute and thus leads to a worse accuracy.

6 Conclusion

In this project, we have achieved three key contributions:

1. We have thoroughly investigated the approach by Davidson, Warmsley, et al. 2017. We have improved the evaluation of the approach by replicating it and applying a train-test split on the data. We have identified that the character-based tokenization for POS-tagging is a weakness of the approach, and that better results can be obtained by using word-based tokenization. By applying aggressive weighting we have improved recall for the hate speech class, which was a problem reported by the authors and can be very important depending on what the classification is supposed to be used for. We have also investigated using other machine learning models, but found that the authors' models perform better.

2. We have successfully implemented two transformer models to better detect hate speech. With the BERT and BERTweet models we could achieve 90% and 91% accuracy. Here, future work would be to further improve recall specifically for the hate speech class, as well as for specific kinds of social discrimination within the hate speech class, by using data augmentation.

3. This project leverages the multilingual translation technique by translating the text from the source language into English and then classify hate speech with a model pretrained with English data. The method is evaluated on a multilingual toxic comment dataset. The performance is compared with a multilingual model trained with annotated data which has comparable performance to the pretrained English model. As the result shows, the method performs well at detecting non-toxic data. But solving the multilingual detection task with translation leads to losing the hate-speech-related attribute, thus the toxic data might be classified as non-toxic.

In summary, we have made substantial progress on both our research questions: We have increased robustness in the detection of hate speech, and we have expanded the model towards multilingual detection without annotated data in the second language.

References

Aluru, S. S., Mathew, B., Saha, P., & Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

Chiril, P., Benamara Zitoune, F., Moriceau, V., Coulomb-Gully, M., & Kumar, A. (2019). Multilingual and multitarget hate speech detection in tweets. *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN) PFIA 2019. Volume II : Articles courts*, 351–360. https://www.aclweb.org/anthology/2019.jeptalnrecital-court.21

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale.

Davidson, T., Bhattacharya, D., & Weber, I. (2019). Racial bias in hate speech and abusive language detection datasets. *arXiv preprint arXiv:1905.12516*.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, *11*(1).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hanu, L., & Unitary team. (2020). Detoxify.

Huang, X., Xing, L., Dernoncourt, F., & Paul, M. J. (2020). Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition.

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, *4*(8), 966–974.

Jigsaw. (2020). Kaggle.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PloS one*, *14*(8), e0221152.

Nguyen, D. Q., Vu, T., & Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 9–14.

Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. *Proceedings of the fifth international workshop on natural language processing for social media*, 1–10.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., & Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.