A Transfer Learning Method for Hate Speech Detection

Eva Šmuc*, Goran Delač*, Marin Šilić*, Klemo Vladimir*
* Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia
{eva.smuc, goran.delac, marin.silic, klemo.vladimir}@fer.hr

***Abstract*—In this work we explore the possibilities of using transfer learning techniques to enhance performance of hate speech detection models by relying on similar linguistic problems (e.g. toxic language detection). Multiple algorithms are trained for similar linguistic tasks on larger datasets, and the obtained models are used for getting predictions on the ETHOS dataset, which we chose as the target dataset of our work. The obtained predictions are used as sole or additional features in the subsequently performed experiments. Multiple algorithms are evaluated, including Logistic Regression, SVM, RidgeClassifier, Decision Tree, Random Forest, AdaBoost, GradBoost, Bagging. Furthermore, multiple textual representations are taken into account including Tf-Idf, Bert embeddings and BERT embeddings combined with the aforementioned additional features. Transformer-based models BERT and DistilBERT are introduced and fine-tuned on ETHOS dataset. All the obtained models are evaluated and the resulting performance metrics are compared to results obtained by the authors of the ETHOS dataset. In order to explore the remaining underlying issues, model-agnostic method LIME is used to obtain explanations for incorrect predictions.**

***Keywords*—transfer learning, hate speech detection, deep learning, feature extraction**

I. INTRODUCTION

In recent years, people have been joining social networks progressively, resulting in a massive increase in interactions on social media. While this type of infrastructure allows constructive conversations, knowledge exchange and *"information dissemination at a fast rate"* [1], it is also vulnerable to suspicious and harmful activities such as fake news, various forms of propaganda, offensive and hate speech. Hate speech is defined by Facebook as a *"direct attack against people – rather than concepts or institutions – on the basis of protected characteristics: race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity and serious disease"*. Presence of hate speech distorts the initial idea of social media platforms as inclusive environments that support constructive conversation and freedom of expression. In May 2016, The European Commission launched The Code of Conduct, together with Facebook, Microsoft, Twitter and YouTube, as a response to rapidly increasing amount of hate speech present online [2]. The Code makes it possible to report content, which is then removed by the company if it violates their hateful conduct policy. In order to respect these regulations, online social media and social networking services are faced with the challenge of detection and removal of hate speech. Due to a large amount of text posted online on daily bases, it is impossible remove the hateful content manually, which motivates further development of automated hate speech detection technologies based on artificial intelligence. One of the main problems in hate speech detection is a lack of available labeled data for training. Even when labeled, most of the datasets lack a transparent and verified annotation process. However, when similar tasks are considered, there is a substantial amount of available data for toxic, offensive and abusive language detection. In this work, the relationship between hate speech and similar tasks is explored through application of transfer learning techniques.

II. RELATED WORK

A. Similar Tasks

Hate speech classification is a complex task, sometimes not clearly distinguished from other forms of abusive language. The focus of work presented in this paper is binary classification performed on the ETHOS dataset comments. Similar tasks to hate speech classification involve toxic, offensive and abusive text classification, considering both binary and multi-class approaches. Most of the work up to 2020 concerning hate speech classification was focused on feature-based classifiers and advantages of ensemble methods ([3], [4], [5]), while recent research concentrates on usage of deep,transformer-based models such as Bert ([6], [7], [8]) and GPT-3 [9]. Research involving Bert includes retraining the model on specific datasets [7] and experimenting with different embeddings [6] in order to improve performance metrics and achieve new state-of-art results. In [5], a detailed analysis of common challenges in Natural Language Processing is provided. Specifically, difficulties in existing datasets for toxic and similar language detection. Despite focusing on toxic language, authors argue that *"even though each field uses different definitions for their classification, similar methods can often be applied to different tasks"* and show that same methods they use in their work can *"effectively be applied to a hate speech detection task"*.

B. Work Related to ETHOS Dataset

ETHOS binary and multi-label versions of the dataset are introduced in [10]. A detailed description of the annotation process is provided, as well as performance of selected models on binary hate speech classification. In

[6], authors analyze the performance of hate speech detection classifier on ETHOS by replacing or integrating the word embeddings (fastText and GloVe) with static BERT embeddings - it was observed that the neural network performed better with static BERT embeddings. In terms of metrics, an improvement of specificity was achieved when compared to fine-tuned BERT. In [9] classification of ETHOS dataset comments is performed using GPT-3. The paper focuses on exploring the ability of GPT-3 to detect hate speech, with and without providing the model with examples. The main idea was to explore if language models like GPT-3 can be used in the future to help prevent the production of offensive and hateful language produced by humans online.

III. TARGET DATASET: ETHOS

For the purpose of this work, binary ETHOS dataset was selected, due to its detailed description of the annotation process which ensured labels balance and diversity. Another important feature of ETHOS dataset collection process is usage of verified annotators.

A. Creation

The key asset presented in [10] is a balanced dataset creation, which purpose is to overcome the issue of existing hate speech datasets such as biased and imbalanced labels. The data consists of Youtube comments collected through Hatebusters and Reddit comments collected through Public Reddit Data Repository. Initially, data labels are predicted by an SVM classifier provided by Hatebusters. In the next phase, comments in the $[.4, .6]$ probability range are manually annotated since their classification is less certain, while the comments in the ranges $[.0, .1] \cup [.9, .1.0]$ are only examined in order to detect possible misclassification. Only certain comments are selected depending on their label and content, to achieve balance and diversity. Data is validated via Figure Eight Data Labeling Platform. Finally, the annotated data is examined manually to prevent any possible misclassification.

B. Overview

Dataset used in this work is a binary version of ETHOS dataset "Ethos_Binary.csv" that contains 998 comments and two classes, hate and non-hate. More precisely, it contains a label *'isHate'*, a decimal number in the range [0,1] which marks the absence or presence of hate in the comment. For every comment, N annotators voted for the set labels. P is the sum of positive votes, which is divided by the number of annotators N, to normalize the *'isHate'* value to the [0,1] range. For the purpose of using algorithms in binary scope, the authors of ETHOS dataset propose binarization of values to the {0,1} classes for each label ('isHate' >= 0.5: label = 1 Else: label = 0). Rounding of values results in 433 samples classified as hate and 565 samples classified as non-hate. The measure of dataset balance can be calculated using the Shannon entropy [6], where n is the number of examples in the dataset, k the number of classes and c_i the size of class i. The Shannon entropy approaches 0 for unbalanced datasets and log(k) for balanced datasets. For ETHOS dataset $Balance = 0.986$, indicating a balanced dataset.

$$Balance = \frac{H}{log(k)} = \frac{-\sum_{i=1}^{k} \frac{c_i}{n} log(\frac{c_i}{n})}{log(k)} \quad (1)$$

IV. EXPLORATORY DATA ANALYSIS

Exploratory data analysis is performed in order to get data insight and summarize the most important characteristics of the dataset. The additional purpose of the analysis is to detect possible differences in the comments belonging to opposing classes. *Perspective API* [11] is used to calculate toxicity and profanity probability scores of the comments. The score indicates how likely it is that a reader would perceive the comment as containing the given attribute. The *nltk* library is used to get sentiment scores for all comments. To get the number of difficult words per comment, *textstat* library is used.

Toxicity The majority of 'hate speech' comments fall in the range between 0.5 and 1, while the majority of 'not hate speech' comments have toxicity probability between 0 and 0.45.

Profanity Comments labeled as 'hate speech' are in the range between 0.5 and 1, but the majority have a profanity score higher than 0.9. Most of the 'not hate speech' comments have probability range between 0 and 0.25.

Sentiment score Sentiment scores are scaled to range [0,1]. The difference in 'hate speech' and 'not hate speech' comments sentiment is not as clear as in the case of toxicity and profanity, but it is still noticeable.

Difficult words The number of difficult words, further defined by *textstat* as rare or atypical words, are more likely to be found in comments labeled as 'hate speech', according to the analysis.

The difference in attribute mean values for both classes is shown in Table I. Higher means for profanity, toxicity and word difficulty are observed in 'hate speech' comments. Lower mean for sentiment score in 'hate speech' class indicates a more negative sentiment of comments in this class than in the comments labeled as 'not hate speech'.

Label	profanity	toxicity	sentiment score	word difficulty
'not hate speech' (0)	0.2931	0.3336	0.4911	2.7557
'hate speech' (1)	0.6113	0.7197	0.3492	3.1778

TABLE I: Mean value of attributes for 'not hate speech' and 'hate speech' comments

V. PROBLEM DESCRIPTION AND PROPOSED APPROACH

A. Problem Description

Throughout this work, several different combinations of text representations and algorithms are tested in order to compare the performance results on the ETHOS dataset.

Transfer learning paradigm is explored by using predictions made by models trained for similar tasks (offensive, abusive and toxic language detection) as additional or sole features used to train the binary hate speech classifier. Transformer-based models are fine-tuned on the same dataset and their performance is compared to other models. Although hate speech and similar forms are not clearly distinguishable [7], we explore how usage of other (and usually larger) datasets built for similar tasks can possibly improve performance of hate speech detection models.

B. Proposed Approach

The proposed transfer learning approach is to obtain new, numerical features, and train models using those features.

Feature set 1 is a set of features obtained by using machine learning models trained on related datasets. Floating point features in *Feature set 1* represent probabilities for positive, 'isHate' class, obtained by using pre-trained models for prediction of class probabilities for ETHOS comments.

Creation of Feature set 1 consists of selecting the larger datasets for similar tasks to hate speech, selecting machine learning algorithms and pre-trained models, and training those models on selected datasets. ETHOS is a binary dataset, so other binary datasets were needed to obtain the probabilities for classes 'non-hate' and 'hate'. Due to a lack of larger binary datasets, several multi-class ones were chosen (as presented in Table II): Toxic, Movies, Youtube, and one binary dataset - FoxNews. Each non-binary dataset was binarized by separating the neutral class 0 (*neither*, *appropriate*, *non-hate*) from the rest of the classes and assigning class 1 to all other classes.

Dataset	Original classes	New binary classes	Size
Toxic	toxic severe_toxic obscene threat insult identity_hate neither	0: neither 1: (toxic, severe_toxic, obscene, threat, insult, identity_hate)	159571
Fox News	non-hate hate	0: non-hate 1: hate	1528
Movies	hate offensive neither	0: neither 1: (hate, offensive)	3208
Youtube	appropriate inappropriate offensive violent	0: appropriate 1: (inappropriate, offensive, violent)	101569

TABLE II: Selected datasets, their original classes, new binary classes and size

After data preprocessing, each of the four chosen datasets (Toxic, Fox News, Movies and Youtube) contains a column 'comment' labeled as 1 (inappropriate, offensive, violent, toxic, hate etc.) or 0 (neutral, neither, appropriate). Five different machine learning methods were chosen to create models by training (or fine-tuning in case of BERT) on each of the four datasets. One linear approach - Logistic regression, two ensemble methods - Random Forest Classifier and XGBoost, and two transformer-based pre-trained models - BERT and DistilBERT. The result of the described process are 20 = (5 models)*(4 datasets) different models shown in Table III, and *Feature set 1* - predictions of class probabilities for ETHOS comments, provided by these models.

	Toxic	Fox News	Movies	Youtube
Logistic regression (LR)	LR_toxic	LR_fox	LR_movies	LR_youtube
Random Forest Classifier (RFC)	RFC_toxic	RFC_fox	RFC_movies	RFC_youtube
XGBoost (XGB)	XGB_toxic	XGB_fox	XGB_movies	XGB_youtube
BERT	toxic_bert	fox_bert	movies_bert	youtube_bert
DistilBERT	toxic_distil	fox_distil	movies_distil	youtube_distil

TABLE III: Selected ML models and datasets

Feature set 2 is created using toxicity, profanity and sentiment probability, as well as the number of difficult words per comment.

Creation of Feature set 2 is explained in Section IV, along with definitions of each attribute. These attributes accentuate differences in comments labeled with 0 and comments labeled with 1, and can also be used as additional numerical features to achieve improvement in classifier performance. *Feature set 2* is shown in Figure 1.

	comment	label	profanity	toxicity	sentiment_score	word_difficulty
0	You should know women's sports are a joke	1	0.208376	0.448056	0.648495	1
1	You look like Sloth with deeper Down's syndrome	1	0.310944	0.448206	0.681315	2
2	You look like Russian and speak like Indian. Both are disgusting go kill yourself	1	0.693099	0.961763	0.184939	2
3	Women deserve to be abused, I guess.	1	0.464719	0.700786	0.242475	0
4	Women are made for making babies and cooking dinner and nothing else!!!	1	0.221928	0.750965	0.499497	0
...	...	...	...	...	...	...
993	From the midnight sun where the hot springs blow	0	0.153424	0.053735	0.499497	0
994	Don't say I'm not your type	0	0.059359	0.151594	0.499497	0
995	And therefore never send to know for whom the bell tolls	0	0.025942	0.068119	0.499497	1
996	And I can't stand another day	0	0.078510	0.187466	0.499497	0
997	All values, unless otherwise stated, are in US dollars."	0	0.010062	0.018603	0.701802	3

Fig. 1: Feature set 2

VI. Approach Based on BERT

Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) [12] learn the context of a word based on both the words that precede it and that follow it ('bidirectional'), as a contrast to directional models that read the text input either left-to-right or right-to-left. It is often enough to fine-tune BERT to achieve high performance results on smaller datasets. In this work, BERT and DistilBERT are used for classification and regression. Additionally, BERT is used for extracting embeddings.

A. BERT and DistilBERT - Classification and Regression

BERT and DistilBERT classification model is fine-tuned on each of the datasets presented in Section V-B and for single sentence classification on the ETHOS dataset in Experiment B. When fine-tuning a BERT regression model, the 'isHate' score is used as a label and the mean-squared loss function is used instead of cross-entropy loss function. BERT and DistilBERT regression model are used in Experiment B, for performance comparison with the classification models. This is done to emphasize the impact that the rounding of the averaged score has on the accuracy of predictions.

B. BERT Embeddings

BERT can be used solely to extract features from text data. Other algorithms for obtaining vector representations for words, such as GloVe, Fasttext and Word2Ve, have the same function. The key advantage that BERT offers are contextualized word embeddings, while the other mentioned algorithms have a fixed representation of the word regardless of the context. Since BERT embeddings are numerical vectors, it is possible to append other additional features and use BERT embeddings with appended features as a representation. Experiment A is performed by learning regression models with different algorithms and using several representations extending BERT embeddings: BERT embeddings only, BERT embeddings with appended *Feature set 1*, BERT embeddings with appended *Feature set 2* and finally BERT embeddings with appended both *Feature set 1* and *Feature set 2*.

VII. EVALUATION

Nested cross-validation is used as an approach to perform model selection and validation. The k-fold cross-validation for model hyperparameter optimization is nested inside the k-fold cross-validation procedure for model selection, where $k = 10$ is chosen for the outer loop, and $k = 3$ for the inner loop. For evaluation purposes, the same evaluation method and algorithms (but adjusted to work with the new data) are selected as in [10]. This was done in order to make a direct comparison with the performance results presented by the authors, achieved when using feature extraction on ETHOS dataset comments (done by converting the raw comments to a matrix of TF-IDF features using TfidfVectorizer). The obtained results are shown in Table IV.

A. Experiment A: Logistic Regression, SVM, RidgeClassifier, Decision Tree, Random Forest

In Experiment A, Logistic Regression, SVM, RidgeClassifier, Decision Tree and Random Forest based models are optimized for each algorithm using nested cross-validation method. The proposed approach with BERT embeddings and appended *Feature set 1* and *Feature set 2* performs best, with Ridge Classifier producing the highest scores. SVM performs almost the same as Ridge Classifier, but with a significantly less amount of time needed for fitting.

Representation	Method	F1	Precision	Recall	Accuracy	fitTime(sec)
Tf-Idf	LogReg	0.6650	0.6694	0.6707	0.6694	166.2673
	SVM	0.6607	0.6647	0.6670	0.6643	103.2807
	Ridge	0.6547	0.6571	0.6580	0.6624	36.0019
	DTree	0.6104	0.6148	0.6152	0.6181	646.2416
	RandomF	0.6441	0.6469	0.6468	0.6504	1.2263

TABLE IV: ETHOS paper: performance metrics for Experiment A

B. Experiment B: DistilBERT and BERT

In Experiment B, BERT and DistilBERT are fine-tuned on the ETHOS dataset. Best performance metrics is achieved with the BERT regression model (when the model is fine-tuned using 'isHate' as a label and the obtained predictions are rounded).

Representation	Method	F1	Precision	Recall	Accuracy	fitTime(sec)
BERT emb	LogReg	0.7195	0.7220	0.7216	0.7284	231.4352
	SVM	0.7145	0.7191	0.7173	0.7245	2.1786
	Ridge	0.7114	0.7140	0.7140	0.7204	44.7056
	DTree	0.5632	0.5791	0.5792	0.5813	294.2670
	RandomF	0.6380	0.6554	0.6419	0.6603	29.2616
Feature set 1	LogReg	0.7452	0.7469	0.7467	0.7525	0.5334
	SVM	0.7343	0.7357	0.7367	0.7415	1.0674
	Ridge	0.7395	0.7407	0.7416	0.7464	0.3148
	DTree	0.7051	0.7080	0.7097	0.7114	22.9166
	RandomF	0.7283	0.7309	0.7285	0.7375	183.9144
Feature set 2	LogReg	0.7762	0.7780	0.7774	0.7826	1.2931
	SVM	0.7808	0.7831	0.7832	0.7865	3.5810
	Ridge	0.7096	0.7129	0.7171	0.7134	0.2996
	DTree	0.7728	0.7767	0.7823	0.7755	18.5314
	RandomF	0.7673	0.7693	0.7703	0.7735	1.9666
BERT emb	LogReg	0.7592	0.7623	0.7602	0.7675	112.2335
+	SVM	0.7145	0.7191	0.7173	0.7245	2.2121
Features set 1	Ridge	0.7114	0.7140	0.7140	0.7204	46.2407
	DTree	0.5632	0.5791	0.5792	0.5813	286.9777
	RandomF	0.6380	0.6554	0.6419	0.6603	27.2942
BERT emb	LogReg	0.7963	0.7959	0.7989	0.8016	135.8343
+	SVM	0.7984	0.7981	0.8026	0.8026	2.2675
Feature set 2	Ridge	0.7912	0.7920	0.7937	0.7966	28.3758
	DTree	0.7721	0.7823	0.7852	0.7745	300.7481
	RandomF	0.7838	0.7842	0.7885	0.7885	222.4563
Feature set 1	LogReg	0.8098	0.8100	0.8130	0.8146	9.9949
+	SVM	0.8080	0.8084	0.8119	0.8126	46.4875
Feature set 2	Ridge	0.8080	0.8079	0.8118	0.8126	5.4765
	DTree	0.7678	0.7704	0.7754	0.7715	43.8981
	RandomF	0.7917	0.7908	0.7946	0.7966	3.4595
BERT emb	LogReg	0.8070	0.8077	0.8089	0.8126	123.8262
+	**SVM**	**0.8159**	**0.8153**	**0.8184**	**0.8206**	**2.1346**
Feature set 1	**Ridge**	**0.8162**	**0.8154**	**0.8198**	**0.8207**	**35.0411**
+	DTree	0.7721	0.7823	0.7852	0.7745	361.5874
Feature set 2	RandomF	0.7825	0.7822	0.7860	0.7875	38.4884

TABLE V: Performance metrics for Experiment A

Method	F1	Precision	Recall	Accuracy
BERT (classification)	0.7551	0.7756	0.7681	0.7644
BERT (regression)	**0.7893**	**0.7911**	**0.7902**	**0.7956**
DistilBERT (classification)	0.7277	0.7436	0.7304	0.7446
DistilBERT (regression)	0.7804	0.7866	0.7808	0.7886

TABLE VI: Performance metrics for Experiment B

VIII. INTERPRETABILITY

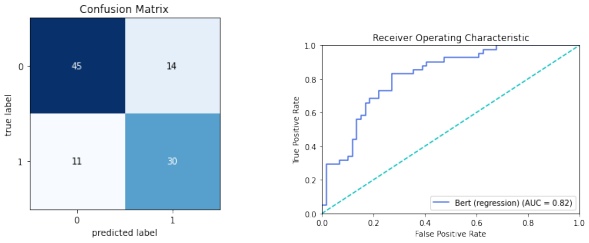
Results of cross-validation provide an overview of model performance over unseen data. This is a process necessary for machine learning model performance estimation, but it doesn't help in understanding model predictions and its decision making path. Models are defined as interpretable *"when humans can understand the reasoning behind predictions and decisions made by the model"* [13]. The purpose of an explanation is to relate feature values of an example to its model prediction in an understandable way for humans. In the case of automated hate speech detection, removing comments with no justifiable explanation can result in violation of corporate technology practices and preferences.

A. Error analysis using LIME

Model-agnostic interpretability techniques enable global and local post hoc model interpretation, independent of implementation. One of the local methods is a local surrogate model, which explains a prediction by replacing complex model with a locally interpretable surrogate model. LIME (Local Interpretable Model-agnostic Explanations), a technique that trains local surrogate models to get explanations for single predictions, was first introduced in [14]. Models from Experiment A do not reach F1

score higher than 0.82, while transformer-based classifiers from Experiment B reach 0.79. To find out the underlying issues, an error analysis is performed using LIME, by providing an explanation for erroneous predictions. Error analysis is based on the one performed in [5].

1) LIME for Text: BERT Regression Model: BERT regression model (the best performing transformer-based model from Experiment B) was tested on the test set consisting of 100 examples. Confusion matrix and the Receiver Operator Characteristic are shown in Figure 2.



(a) Confusion matrix (b) ROC

Fig. 2: BERT Regression Model

False Negatives

- **'Hate speech' comments without swear words or extreme language** Previous research shows that *"classifiers often fail when there are no obvious hateful words present"* ([5], [15]).

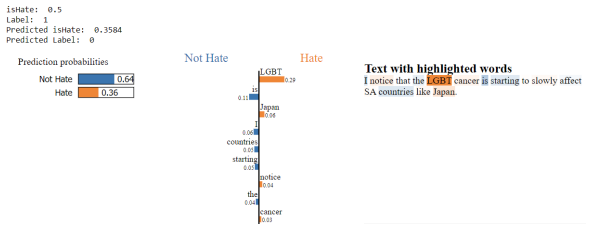


Fig. 3: An example of a comment misclassified as 'not hate speech', because it doesn't contain any hateful vocabulary, swear words or offensive language

- **Questionable 'isHate' score** Out of 11 false negatives, 8 have an original 'isHate' score of 0.5, which indicates that comments with this score are problematic for the classifier. Even though the label of the following comments is 1 when 0.5 is rounded, their content suggests that their 'isHate' score should be higher.

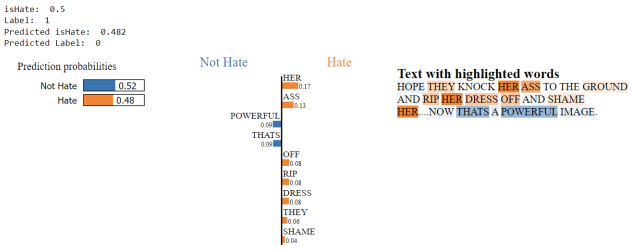


Fig. 4: An example of a comment with a questionable 'isHate' score

False Positives

- **Overrepresented terms in the training set** The classifier learned to use the words "white", "islam", "muslim" (and some others), as features for classifying comments as 'hate speech', which is not always the case. These words alone do not necessarily indicate hate speech. Erroneous predictions that the model makes because of their presence are the result of data distribution in the training set. For example, 40/58 comments containing the word "white" in the training set are comments with label 1, 24/33 comments containing the word "islam" are comments with label 1 etc.

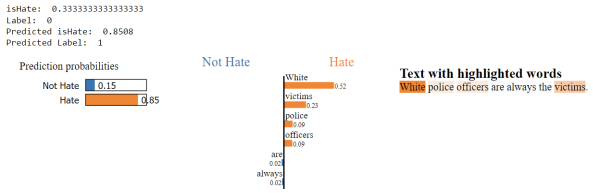
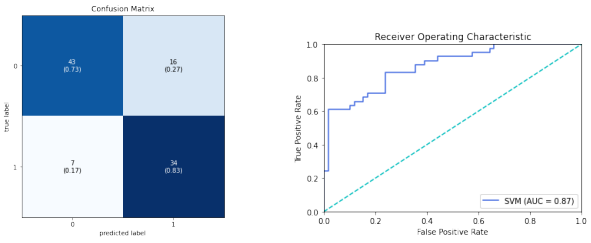


Fig. 5: An example of a comment misclassified as 'hate speech', because of the word "White"

2) LIME for Tabular Data: SVM: For applying LIME for tabular data to get prediction explanations, SVM with *Feature set 1 + Feature set 2* as representation was selected. Even though the Ridge Classifier with BERT embeddings + *Feature set 1 + Feature set 2* as representation had the best overall performance, the best performing model with *Feature set 1 +Feature set 2* as representation was chosen because these features are named and both their meaning and creation process are explained in previous sections. BERT word embeddings have 768 unnamed features with no clear meaning and can't contribute to explanatory analysis. The selected SVM model was tested on the same test set consisting of 100 examples. Confusion matrix and the Receiver Operator Characteristic are shown in Figure 6.



(a) Confusion matrix (b) ROC

Fig. 6: SVM

In LIME for tabular data, floating point numbers represent the relative importance of selected features and feature names on the right.

False Negatives

- **'Hate speech' comments without toxic language** All the False Negatives have the `toxicity` feature contributing to class 0. Since `toxicity` is the most contributing feature, the predicted label for these

comments is 0. This again points to an already stated issue of detecting comments containing hate speech but not swear words or toxic language.

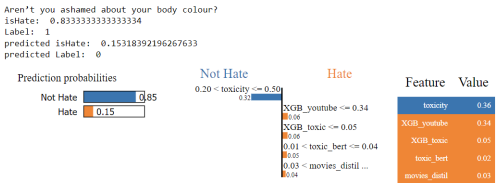


Fig. 7: An example of a 'hate speech' comment without toxic language

False Positives

- **'Not hate speech' comments with toxic language** Upon detailed analysis, it can be concluded that `toxicity` is the most contributing feature, which is the cause of the majority of False Positives. Each of the following comments contain words such as "wtf", "bastard", "disgusting" and other words likely to be overrepresented in the training data of classifiers for toxic language. Detecting toxic language is a similar task, but the differences become more apparent once a dominant toxicity feature is used in training models for a hate speech detection task.

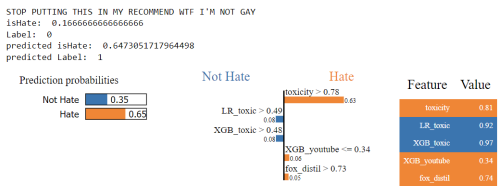


Fig. 8: An example of a 'not hate speech' comment with toxic language

B. Error analysis summary

Analysis showed that the analysed models had 4 False Negatives and 8 False Positives in common. After inspecting their content and getting the same prediction using both models, it is reasonable to question the quality of these labels. Through the error analysis, some of the possible biases were detected. Definitions of these biases and their consequences are given in Table VII.

Bias	Meaning	Result
Sampling bias	The way that dataset examples were sampled doesn't reflect real-world distribution	Misclassification due to terms over-represented in one class ("white", "muslim"...)
Annotator bias	A bias caused by annotators because of subjectivity and differences in knowledge [16]	Lack of quality and consistency in labels

TABLE VII: Definitions and results of detected biases

IX. CONCLUSION

Due to the ambiguous nature of language and its structural complexity, differentiation of hate speech and other forms of similar language presents a challenging task. In this work, the relationship between hate speech and other forms of similar language phenomena is explored through application of transfer learning techniques. Models are trained for hate speech detection using additional features, obtained through predictions of models created for similar tasks. Performed error analysis using LIME highlighted a problem in this approach. Even though related language phenomena helps in improving classifier performance due to its semantic similarity with hate speech, it also causes inaccurate predictions because it does not necessarily imply hate speech. Despite this problem, the presented approach still proved to be an effective and affordable strategy to increase model performance on smaller datasets. Error analysis brought to attention other underlying problems including sampling bias and lack of quality and consistency of ETHOS labels. This is arguably one of the most problematic issues, having in mind that only the test set was inspected closely, thus, incorrect labels are possibly present in the training set. Finally, the future research steps will be focused on ensuring better label quality. Specifically, we argue that better distinction between hate-speech and similar phenomena, might be a way of improving classifier performance. This stems from the fact that poor distinction between the the mentioned phenomena presents performance issues even for the transformer-based models.

REFERENCES

[1] B. Mathew, R. Dutt, P. Goyal, and A. Mukherjee, "Spread of hate speech in online social media. in: Proceedings of the 10th acm conference on web science," pp. 173–182, 2019.

[2] European Comission, "The code of conduct on countering illegal hate speech online," https://ec.europa.eu/commission/presscorner/detail/en/qanda_20_113, accessed: 2022-04-20.

[3] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," 2016.

[4] P. Burnap and M. L. Williams, "Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making," 2015.

[5] B. van Aken, J. Risch, R. Krestel, and A. Löser, "Challenges for toxic comment classification: An in-depth error analysis," 2018.

[6] G. Rajput, N. Singh, S. Sonbhadra, and S. Agarwal, "Hate speech detection using static bert embeddings," 2021.

[7] T. Caselli, V. Basile, J. Jelena Mitrović, and M. Granitzer, "Hatebert: Retraining bert for abusive language detection in english," 2021.

[8] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, "Deep learning models for multilingual hate speech detection," 2020.

[9] K.-L. Chiu, A. Collins, and R. Alexander, "Detecting hate speech with gpt-3," 2022.

[10] I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas, "Ethos: an online hate speech detection dataset," 2021.

[11] "Perspective api," https://perspectiveapi.com/how-it-works/, accessed: 2022-04-20.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[13] MathWorks, "What is interpretability?" https://www.mathworks.com/discovery/interpretability.html, accessed: 2022-05-15.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?": Explaining the predictions of any classifier," 2016.

[15] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," 2017.

[16] M. Wich, H. Al Kuwatly, and G. Groh, "Investigating annotator bias with a graph-based approach," 2020.