Automatic Classification of Hate Speech

Anonymous ACL submission

1 Introduction

Offensive language is a pervasive issue on social media that can have negative consequences for society. One way to tackle this issue is through automatic identification of offensive language, which can be a useful addition to human moderation on social media (Zampieri et al., 2019). By using automatic methods, we would be able to better detect, categorize, and ultimately discourage and penalize harmful behavior on social media. In addition, it is an important means to protect social media users, especially those who might be in marginalized positions in society and suffer greatly from targeted hate speech.

Detecting offensive language is, however, a complex task due to the variation in offensive language and the subtle nuances in distinguishing offensive from non-offensive language. Offensive language ranges from very direct insults, such as harsh language, swear words, and intimidation, to subtle language, such as sarcasm, metaphors, and irony (Herath et al., 2020; van Aken et al., 2018). On the other hand, non-offensive language might also contain swear words or irony without harmful intent. These subtle differences between offensive and non-offensive language with limited world knowledge and knowledge of the discourse context make offensive language detection a challenging task.

To tackle this challenging task, computational methods must be explored that can capture these fine-grained distinctions. Recently, ensembling methods have become more popular to deal with this, as it is a way of combining the strengths of multiple different models and minimizing the weaknesses of individual models.

What makes it even more challenging is the general issue in NLP of out-of-domain generalizability. Models tend to perform worse on cross-domain experiments, even though it is especially important for offensive language detection models: they must showcase the ability to generalize to out-of-distribution sets to be valuable for later real-world use, for example in addition to human moderation on various social media platforms.

There are few papers to our knowledge that use ensembling methods in combination with in-domain and cross-domain experiments. In this paper, we want to take a step in that direction. We perform experiments for a binary classification task for offensive language detection, with '1' corresponding to offensive texts and '0' to non-offensive texts. We combine the strengths of three state-of-the-art transformer-based models, BERT, HateBERT, and fBERT, by employing three ensembling strategies. These strategies are Hard Majority Voting, Soft Majority Voting, and Stacking. In addition, we train and evaluate these models in an in-domain and cross-domain setting to compare the results across domains and evaluate the possible drop-off in model performance in a cross-domain setting. We then perform a qualitative error analysis on the false positives and false negatives in the predictions of our best-performing ensemble model. Our results indicate that ensemble models perform better than individual models in both domain settings and that there are minor differences in performance amongst ensemble models, with HMV performing the highest of all three models in cross-domain experiments.

We are interested in the following three questions:

- **RQ 1:** How do ensembling methods of multiple transformers perform compared to singular transformers on a binary offensive language detection task?
- **RQ 2:** How do these ensembling methods perform in in-domain vs. cross-domain settings?
- **RQ 3:** Which of the three ensembling methods performs the best and what weaknesses does it still have?

This paper is structured as follows: Section 2 provides an overview of related studies using ensembling methods on this task. Section 3 describes our used datasets, the preprocessing steps, and their distribution. Section 5 reports our methodology and models. Section 6 contains the quantitative analysis of our ensemble models' performance in in-domain and cross-domain experiments. In Section 7, We perform an in-depth error analysis of cross-domain predictions of our best-performing ensemble method. Finally, we discuss our results in Section 8, and conclude our findings in Section 9.

2 Related work

2.1 BERT-based models

BERT-based models have been very popular in the offensive language detection community. In a systematic review of automatic hate speech detection, Jahan and Oussalah (2023) found that 38% of deep-learning models in recent papers on hate speech detection are BERT-based models. This relatively high percentage highlights the extent to which BERT plays an important role in this task. Several studies have found that BERT performs significantly better than other deep learning models such as LSTM and BiLSTM. For example, Alatawi et al. (2021) performed an experiment for white supremacist hate speech detection using a BiLSTM and BERT model, and found that the BiLSTM model got an F1-score of 0.75, whereas BERT achieved a score of 0.81.

2.2 Recent developments: ensemble methods

As previously mentioned in Section 1, ensembling methods have also become more popular in the offensive language detection community. Markov et al. (2022) performed in-domain and cross-domain experiments for Dutch hate speech detection using SVM, BERTje, and RobBERT, and an ensemble model using these three. Their results show a significant drop-off in performance for all models when the models are tested on out-of-domain data. The results also indicate that the ensemble approach greatly outperforms the individual models. Due to the capability of ensemble models to make use of the best-performing model on a specific text, it is often an improvement over individual models (Markov et al., 2022).

The novelty of the ensembling method combined with the little research done using ensemble methods and cross-domain experiments, suggests that similar experiments must be done to further solidify these results. Our paper strives to perform similar cross-domain experiments using various ensembling methods.

3 Datasets and Distribution

Datasets: This section contains general information about our chosen datasets, highlighting the similarities and differences between them. We make use of two different offensive language datasets in our experiments, namely the Offensive Language Identification Dataset (OLID) 1.0 (Zampieri et al., 2019) and Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) dataset 1.0 (Mandl et al., 2019). While the latter also has Hindi and German datasets, in our experiments we only consider the English dataset, maintaining a monolingual approach. In terms of similarities, both datasets have similarly general definitions for what is considered hate speech for sub-task A, as both define hate speech as "any form of non-acceptable language", such as aggression, insults, or threats (Mandl et al., 2019; Zampieri et al., 2019). In terms of differences, the HASOC dataset contains texts from Facebook and Twitter, while OLID contains Tweets (Twitter posts) only.

Preprocessing: The OLID training set has been preprocessed to a smaller subset with the same size and label distribution as the HASOC training set. This ensures that the size of the datasets has no impact on the results and any differences can be written up to the source and nature of the data itself. In addition, all datasets have been preprocessed to either have the label '0' for non-offensive language or non-hate speech, and '1' for offensive language/hate speech. This resulted in three sets for our experiments: the small OLID training subset, the OLID test set, and the HASOC training set.

Distribution: The distribution of the datasets can be found in Table 1. The HASOC dataset, which is the cross-domain, and the OLID training dataset, which is the in-domain, have been adjusted to be the same size. There is a class imbalance leaning towards the non-offensive class for all datasets. It can be assumed this is similar to real-world situations, though the exact distribution may not be the same.

Dataset	Purpose	Non-offensive	Offensive	Total
HASOC	training	3591	2261	5852
OLID	training	3591	2261	5852
OLID	testing	620	240	860

Table 1: Distribution of datasets

4 Experimental setup

Experiments: For the in-domain setting, we trained the three base models on the OLID training set and evaluated them on the OLID test set. Next we separately performed the three ensembling methods and compared their performance. The same process was done for the cross-domain, except the base models were trained on the HASOC training dataset and evaluated on the OLID test set.

For all models, we used the default hyperparameter settings in training. While performing hyperparameter tuning would likely improve the performance of the models, we considered this not the focus of the current paper.

Evaluation metrics: For all experiments, we report macro-averaged precision, recall, and F1-score, overall and per class. We also create confusion matrices for each model. These standard evaluation metrics will allow us to evaluate the type of prediction errors made and detect patterns in which class is more/less difficult to predict, i.e., perform a quantitative error analysis.

5 Methods

This section describes the base models used in our experiments, and chosen ensembling methods.

5.1 Base models

BERT: BERT (Devlin et al., 2019) is a transformer model pretrained on a large amount of English data from Wikipedia and Google's Books Corpus. BERT is pretrained in a self-supervised fashion with two objectives: masked language modeling and next-sentence prediction. We built two fine-tuned BERT models for in-domain and cross-domain experiments with the same default model configuration and one training epoch.

HateBERT: HateBERT (Caselli et al., 2021) is a version of BERT specifically retrained for abusive language detection. It was trained on the dataset RAL-E, which consists of English Reddit comments from banned subreddits. It was evaluated on three different datasets, targeting hate speech, abusive language, and offensive language. On all three, HateBERT outperformed BERT in the original paper (Caselli et al., 2021). This suggests that HateBERT should perform better in our experiment than a general BERT model. To adequately compare the two models, the default setup was kept and both were trained with only one epoch. We experimented with training for 5 epochs but found that this resulted in worse performance for the HateBERT model on the binary offensive language detection task.

fBERT: fBERT (Sarkar et al., 2021) is a version of BERT specifically trained to detect offensive language, which was trained on the SOLID dataset. This dataset contains over 1.4 million instances of offensive language, making it the largest English dataset of this type. SOLID uses the same annotation model as OLID but makes use of semi-supervised learning for its annotations, rather than manual annotation. fBERT is meant to be more general than HateBERT, as it encompasses multiple types of offensive language, such as hate speech, cyberbullying, and profanities, whereas HateBERT focuses mostly on aggression. We expect fBERT to outperform the other models for this task, as the dataset it was trained on is very similar to the one used for testing and in-domain training in our task. fBERT was also trained with 1 epoch. This model replaces the previous underperforming BiLSTM model.

To keep the models consistent across runs, we used set random seeds. For BERT this was 95489322, for HateBERT it was 42, and for fBERT 31415926. These were used both for the in-domain and cross-domain models. We also tried the seed 1021 for BERT, but this predicted non-offensive on all instances. Since we wanted to be able to compare the ensembling methods to the results of the base models, we changed this seed.

5.2 Ensemble strategies

Hard Majority Voting: Hard Majority Voting is similar to taking a democratic vote, with each model having an equal say in the final prediction. With this method, the most common prediction amongst the models will simply be taken. For example, in our case, if BERT and HateBERT predict 0 or not offensive, and fBERT predicts 1, or offensive, then the HMV prediction will be 0, as two out of three models predicted this. To calculate this, the predictions of the three models are summed. If this

number is two or more, the prediction is 1, if the number is less than two, the prediction is 0.

Soft Majority Voting: Soft Majority Voting works similarly to Hard Majority Voting, but with SMV the probability scores of each class per model are taken, rather than only the final predictions. These are put together to decide the final SMV prediction. This is done by summing the probabilities per class, meaning offensive (1) and not offensive (0), and determining the higher probability.

Stacking ensemble: Stacking ensemble involves using a model to learn how to best combine the predictions from basic models. In our experiment, we first train three models under 5-fold cross-validation to produce predictions for each instance in the training data. Then we apply feature engineering to encode input text information generating the following additional features: (1) the Vader compound score generated by utilizing NLTK sentiment package; (2) the number of words (normalized); (3) the number of characters (normalized).

Then we use the above-mentioned six features, which include label predictions (0 or 1) of three models and three additional features, to train a meta-model, which is a gradient boosting classifier (scikit-learn implementation).

To get the six features for the test dataset, we first train three models on the whole training dataset to get label predictions. Then we apply feature engineering to the test dataset to get 3 additional features. Finally, we run these features on the meta-model and get the final predicted labels for evaluation.

6 Quantitative analysis

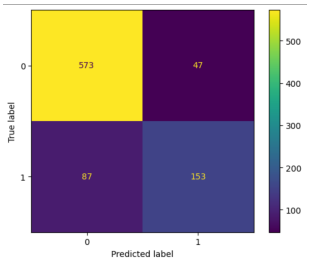

Figure 1: Confusion Matrix of in-domain Hard Majority Voting

There is not much difference between the Hard and Soft Majority Voting methods in both the cross-domain and in-domain settings. This suggests that the three base models are already quite confident in their predictions, such that combining the prob-

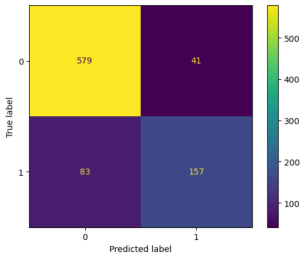

Figure 2: Confusion Matrix of in-domain Soft Majority Voting

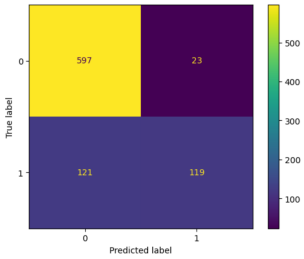

Figure 3: Confusion Matrix of cross-domain Hard Majority Voting

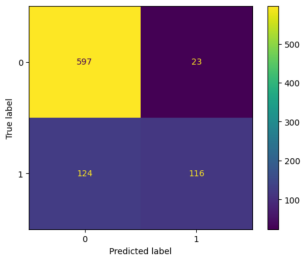

Figure 4: Confusion Matrix of cross-domain Soft Majority Voting

Method	Domain	Class	Prec.	Recall	F1-score
HMV	IN	OFF	0.77	0.64	0.70
HMV	IN	NON	0.87	0.92	0.90
HMV	IN	Macro	0.82	0.78	0.80
SMV	IN	OFF	0.79	0.65	0.72
SMV	IN	NON	0.87	0.93	0.90
SMV	IN	Macro	0.83	0.79	**0.81**
SE	IN	OFF	0.72	0.67	0.69
SE	IN	NON	0.87	0.90	0.89
SE	IN	Macro	0.80	0.78	0.79
HMV	CR	OFF	0.84	0.50	0.62
HMV	CR	NON	0.83	0.96	0.89
HMV	CR	Macro	0.83	0.73	**0.76**
SMV	CR	OFF	0.83	0.48	0.61
SMV	CR	NON	0.83	0.96	0.89
SMV	CR	Macro	0.83	0.72	0.75
SE	CR	OFF	0.71	0.50	0.59
SE	CR	NON	0.83	0.92	0.87
SE	CR	Macro	0.77	0.71	0.73

Table 2: Ensembling results: Precision, recall, and F1-score macro-averaged and per class for the three ensembling methods, using an in-domain and cross-domain setup. 'HMV': Hard Majority Voting, 'SMV': Soft Majority Voting, 'SE' for Stacking Ensemble. The highest macro f1 result per domain is in bold.

abilities of the three models does not lead to major changes compared to simply combining their output predictions. This is the case for incorrect predictions as well.

The drop-off in results between the in-domain and cross-domain is small for both Hard and Soft Majority Voting, with a 0.02-point decrease in accuracy and a 0.05-point decrease in macro F1 score. The biggest decrease for both methods occurs in the recall of the offensive category, with around 0.10 point decrease. This can also be seen in the confusion matrices in Figures 3 and 4. The False Negatives vastly outweigh the False Negatives and even outweigh the True Positives. This suggests that these methods over-predict the non-offensive category. This pattern is not as strong in the in-domain experiments, but the False Negatives still occur around twice as much as the False Positives. These comparative results are similar to those of the individual models. However, unlike the individual models, the precision scores stay relatively consistent and actually see a slight improvement on the cross-domain. It is unclear why this is the case, as we would expect better results on the in-domain, especially since the fBERT model is also trained on a dataset similar to the in-domain dataset. It is possible that the cross-domain contains more offensive examples that the various models were less confident about, but were correctly classified as offensive when the various outcomes were combined.

Overall, Table 2 displays that the Hard Majority Voting method resulted in slightly higher performance for the cross-domain setup, with a macro F1-score of 0.76 compared to the SMV score of 0.75. For the in-domain setup, the difference is also a matter of one point, with Soft Majority Voting achieving a macro F1 score of 0.81 compared to the HMV score of 0.80.

7 Qualitative analysis

We analyzed a sub-sample of 40 cases of false negative and 40 of false positive labeling from the HMV cross-domain experiment results. Our analysis focused on identifying patterns in error class, determining whether the errors were due to doubtful labeling, toxicity without swear words, rhetorical questions, metaphors, use of rare words, or sarcasm and irony, error classes which were inspired by van Aken et al. (2018). This section will first discuss common patterns for false negatives in Subsection 7.1, followed by an analysis of common patterns for false positives in Subsection 7.2.

7.1 Error Classes of False Negatives

Doubtful Labels Of this sub-sample the large majority, 72.5 percent, of cases were deemed to have questionable labels. In most cases, the labels were blatantly wrong, however, there were a handful of tweets that were difficult to classify due to lack of context or clarity.

> **Example A:** "#BiggBossTelugu2 how many of you #KaushalArmy cum @USER fans... URL."

These cases of lack of context are in part related to how the data is cleaned, where specific URLs and user names are removed. Although this practice is justifiable in the data collection process, it may at times remove necessary context that makes a text offensive. However, it would perhaps be difficult for models to be able to process this type of context and understand the implication it has in relation to the tweet it is attached to. Ultimately, the fact that such a large percentage of cases we observed have been mislabeled calls into question

the quality of the dataset, and by extension the true quality of any model trained and tested on this data.

> **Example B:** "@USER #Holder needed to be impeached."

Rare words Of the remaining 27.5 percent of cases, error classes of toxicity without swear words and the use of rare words were the most prevalent. Notably, we included the prevalent use of hashtags in a given tweet as a case of rare words, as it seems likely that in some cases long strings of hashtags could be difficult for a model to accurately parse and understand.

> **Example C:** "#Cuckservative Traitors Are Worse Than Fortnite Players URL #Conservatives #TriggerWarningRadio"

Toxicity without swear words Additionally, cases of the model not identifying toxicity without swear words seem to again be related to the issue of lack of context, but in a slightly different way. In examples of toxic tweets without swear words or intense language, the messages tend to reflect a more personal attack for which one would need some situational context or knowledge of social norms to be able to judge the message as offensive, which models generally lack.

> **Example D:** "#Barbara Boxer If LIBERALS Want to Use RAPE as a Political Tactic! THEY Had Better Have Some HARD EVIDENCE! Her Word Isn't Evidence!!! Its Called HEARSAY . On that note, I Could Say ' YOU ' Cornered Me In A Bathroom, And put YOUR Hands All Over Me' 35 Years Ago. PROVE ME WRONG URL"

7.2 Error Classes of False Positives

Usage of extreme/harsh language in false positives. Similarly to van Aken et al. (2018), we observe a large number of texts that contain harsh language or swear words that convey a non-offensive meaning. 60% of our sample contains harsh language while being non-offensive. As van Aken et al. (2018) argue, the model might be finding the strong correlation between coarse language and offensiveness, which makes it wrongfully label non-offensive texts as offensive. Example E showcases a sample in which the non-offensive text contains such harsh language. The word "screw up" with a negative informal connotation, in combination with the accusatory tone, is what we consider harsh language. However, the text in its entirety is not offensive and does not harmfully target a particular person or group.

> **Example E:** "#SEO #Tips: You are the master of your own fate online, so be wise and don't expect pity. If you screw up, nobody else is to blame."

Out of these samples with harsh language, 22.5% consists of texts conveying the user's strong opinions about certain situations, and are not offensive. Example F exemplifies this pattern, as the texts refers to the Brexit as "an absolute farce", a type of slapstick comedy, which the HMV model predicted as offensive. Similarly to Example E, the harsh language 'absolute farce' might have triggered the model to label it as offensive, without taking into account the context of the Brexit.

> **Example F:** "#ChequersPlan What an absolute farce Brexit is. The conservatives have no idea at all and May is now totally discredited. Time for an election #ForTheMany"

Doubtful labels. Similarly to van Aken et al. (2018), a number of our false positives have been wrongfully annotated as non-offensive, whereas we consider them to be offensive after reviewing. 7.5% of our prediction sample consists of texts labelled wrongfully as non-offensive. These contain strong accusatory offensive language, therefore we consider them to be wrongfully labelled. One example is Example G, in which the harasser directly targets "Mike McCoy", while referring to the target as "hot trash". We consider the reference of a person as "hot trash" in this context to be offensive and a direct attack to the target, and thus wrongfully annotated as non-offensive.

> **Example G:** "@USER My only point of contention is this. He decided to keep this group which has top end talent and also decided to hire hot trash Mike McCoy. Wilks hasn't even been able to get his defense to line up correctly. He is in over his head."

Like van Aken et al. (2018), we suggest more care being put into analyzing annotator agreement when building the dataset, and clearly defining the phenomenon of offensive language to ensure good-quality annotations.

Lack of context. We found that the SMV method wrongfully predicted several texts to be negative due to a lack of context or world knowledge. 12.5% of our sample consists of texts referring to an object or person without the necessary

context to interpret it as offensive language. Example H, for example, contains the word "dicks", which is a vulgar word that might be offensive in a given context, but the lack of context makes it difficult to definitively label it as such.

> **Example H:** And dicks. URL

These are difficult cases where we believe the model should not label it as offensive, as we as humans also would typically not find this offensive outside of the given context.

Unclear texts. Some instances in our sample were texts that we found too unclear to interpret ourselves, let alone the model. We found 7.5% of our sample to be unclear. These texts often contain incoherent sentences due to the sentence structure or lack of punctuation. The example below exemplifies this pattern, as it contains the verb 'be' twice, making the text confusing to read.

> **Example I:** Alex Jones be smokin be dicks out here but mans really got supporters out here

Quotations or references. Finally, similarly to van Aken et al. (2018), we found several cases in which non-offensive comments that were references to other people's offensive language or harmful actions. 10% of our sample were references to harmful behavior. Example J showcases this pattern, in which the user refers to the person 'Mark Knight' and his presumably racist actions. In this case, it appears the classifier was unable to interpret this action of 'drawing a racist picture' in the larger sentence structure of X says Y.

> **Example J:** #Repost thesavoyshow with get_repost... #MarkKnight Says he is not a racist after drawing a racist picture of #SerenaWilliams .. Do y'all believe him or nah? URL

8 Discussion

One of the main takeaways of these results, in particular of our error analysis, is that the quality of the dataset leaves much to be desired. Comparing our findings to assignment 3 on hate speech lexicons, we are not able to directly compare the two error analyses because, for the current paper, we applied a different, more systematic approach to this process. However, a common theme that appears in both qualitative analyses is the discussion of the poor quality labeling of the data.

The quality of the dataset is a crucial aspect of training and testing a model, regardless of the type of task or specific model. At the same time, building and annotating a dataset is a notoriously difficult task, and once we are dealing with datasets of a certain size they will inevitably have issues, such as cases of mislabeling or questionable entries. With this in mind, the sub-samples of the dataset we have surveyed throughout our analysis of the results present a truly concerning image regarding the quality, and by extension utility, of the dataset. From our observations, we have found the data to be poorly labeled to an extent that places the performance of the models trained on this data into question. It is difficult to draw any well-founded conclusions based on the results, particularly in terms of comparing performance across models, when they are trained and tested on such a questionable dataset. We cannot expect a higher performance of models when they are trained on data that is often either mislabeled or difficult for even human annotators to categorize due to lack of context. By extension, it may be unwise to assume that performance on such a dataset can present an accurate reflection of the true performance of these models for the current task.

Stacking Ensemble achieves similar F1 scores to individual models, yet it is not the best-performing approach compared to majority voting methods. This could be due to the quality of the feature engineering. In our experiment, we implemented 3 additional features: 1) the Vader compound score; 2) the number of words (normalized); and 3) the number of characters (normalized). Due to the time limit, we were not able to implement other features and conduct a thorough correlation analysis. For future experiments, we could improve our method by investigating more features based on the literature, namely: 1) the number of first-person personal pronouns, which are often used as features for hate speech detection to distinguish between 'us' and 'them'; 2) the relative number of upper case characters; 3) the relative number of exclamation marks, etc (Markov et al., 2022; van Aken et al., 2018). In addition, we could also use grid search to tune the parameters of the meta model.

In addition, we also ran a small experiment using raw output values as features instead of label values for the cross-domain. For each instance, we added two raw output values generated by the model.predict() method. However, this results in

a lower macro F1-score of 0.61. Thus, we suspect that having many features (nine in total) and lack of value transformation & scaling of raw output hinder the performance. Accordingly, future research can focus more on exploring the effectiveness of using labels as features vs. raw output.

9 Conclusion

This paper discussed three ensembling methods combining three base models across an in-domain and cross-domain setting. The ensembling methods were Hard Majority Voting, Soft Majority Voting and Stacking Ensembling. The base models used were BERT, HateBERT and fBERT. All ensembling methods outperformed the base models in both domain settings. There was only a slight difference between the ensembling method results, with SMV scoring the highest in the in-domain, with a macro f1 score of 0.81, and HMV scoring the highest in the cross-domain, with a macro f1 of 0.76. Our error analysis showed that most False Negative errors were due to questionable annotations, whereas the False Positive errors showed a larger variety, including the use of extreme language in a non-offensive context, and the use of offensive or extreme language to express an opinion without targeting a group or person.

References

Hind S. Alatawi, Areej M. Alhothali, and Kawthar M. Moria. 2021. Detecting white supremacist hate speech using domain specific word embedding with deep learning and bert. *IEEE Access*, 9:106363–106374.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Mahen Herath, Thushari Atapattu, Hoang Anh Dung, Christoph Treude, and Katrina Falkner. 2020. Adelaidecyc at semeval-2020 task 12: Ensemble of classifiers for offensive language detection in social media. *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1516–1523.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Ilia Markov, Ine Gevers, and Walter Daelemans. 2022. An ensemble approach for dutch cross-domain hate speech detection. In *Natural Language Processing and Information Systems*, Lecture Notes in Computer Science, pages 3–15. Springer International Publishing.

Diptanu Sarkar, Marcos Zampieri, Tharindu Ranasinghe, and Alexander G. Ororbia II. 2021. FBERT: A neural transformer for identifying offensive content. *CoRR*, abs/2109.05074.

Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 33–42.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

A Task division