Implicit and Explicit Hate Speech Detection with HateBERT

Jeremy Serillon **Océane Voland** **Thomas Cirillo**

Group 21

Abstract

This project addresses hate speech detection and binary/multi-class classification on social networks. We used three categories: not hate, implicit hate and explicit hate. We will present our implementation of a pre-trained *HateBERT* model that we trained on a 20'000 tweets dataset from the Implicit Hate Corpus and with additional synthetic data.

Keywords: Implicit Hate, Hate Speech Detection, *HateBERT*, BERT, Social Media, Binary Classification, Multi-class Classification, Fine-Tuning, Synthetically Augmented Dataset.

1. Introduction

Although explicit hate speech can be relatively easy to identify via direct slurs and insults, implicit hate is a more intricate and elusive form of hate to identify correctly. Detecting such forms of online hate is a non-negligible factor for discrimination moderation and healthier online interaction that our project can be used for.

For these reasons, this paper aims at expanding a pre-trained deep learning model (specifically *HateBERT*) to be able to detect and discriminate implicit forms of hate speech from explicit hate and not hateful comments of different social platforms such as X (formerly Twitter) or Reddit. Additionally, we also want to study the performance of our model compared to existing work on hate speech detection.

2. Related Work

For our project, we aimed at reproducing a similar approach as the study "*Latent Hatred: A Benchmark for Understanding Implicit Hate Speech* " [1], but bringing possible improvement and generalization

The authors of the paper built multiple classifiers using SVMs and BERT model. We believe that the hate-specific pre-trained *HateBERT* model could be a valuable candidate to improve the model's predictions. The corresponding study "*HateBERT: Retraining BERT for Abusive Language Detection in English*" [1] has been released only few months before the *Latent Hatred* paper, preventing the authors from using this model.

The *HateBERT* model has been trained on banned Reddit communities which used varying forms of hate speech on diverse topics. It is the exact reason why we used a dataset coming from another social network (X): the goal is not to be biased on a specific social network's way of user-to-users communication standards.

Indeed, the two platforms are different in the way an individual or a group of people can share informations. Reddit is a community based platform where users can subscribe to certain "subreddits" (i.e. communities) and specific users, while X is based on quick interaction and broad exposure between users only without the notion of "proper" communities.[1] This factor could make our study biased towards a specific form of hate speech, which we would like to avoid by changing the social network of the original model.

To reproduce the results of the study "*Latent Hatred: A Benchmark for Understanding Implicit Hate Speech*" paper [2], we decided to use their own made dataset of 21'482 tweets originates from which is an open source MIT-licensed dataset. The *Implicit hate Corpus* contains example of explicit, implicit and not hateful texts with their labels. The authors of the paper only fine-tuned BERT for binary classification of implicit and non hateful texts. In our project, we replicated this approach to be able to compare our results but we also fine-tuned *HateBERT* on the three classes, as this approach better reflects real-life classification scenarios.

Upon observation of the dataset, we can notice a clear unbalance of the classes distribution as visible in *Figure 1*.

Unbalanced dataset often lead to *"over-classify the majority group due to its increased prior probability."*[3]. Addition-

[1] As of February 3rd 2025, the social network X fully integrated its new "X Communities" feature across its platform. This release could make X's information sharing similar to how Reddit operates but because this is extremely new and because our used research papers where both made in the year 2021, we will not address these changes. This project was made from April to May 2025.

EE-559: *Deep Learning*, 2025.

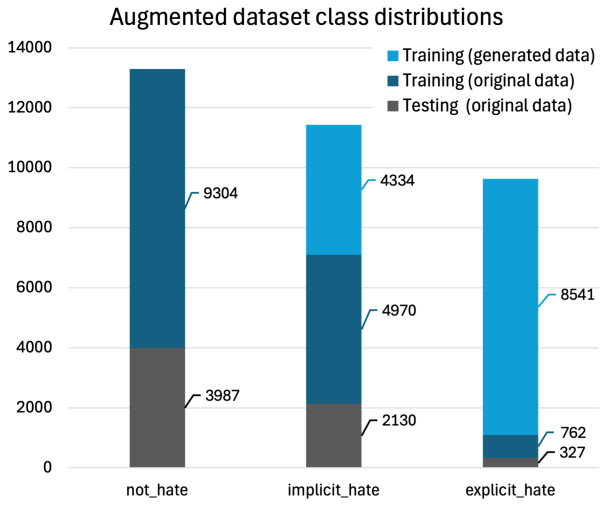

Figure 1. Class distribution of the "Implicit hate corpus"[2] (labelled as Original dataset) and our own augmented datasets.

ally the dataset contains a limited number of data for the implicite (7000) classe and particularly for the explicit (only 1000) one. Small datasets often *"negatively affect the performance of a DL model due to overfitting"* [4].

3. Method

The code used for this project, can freely accessed and consulted in a public Git repository [5].

As discussed previously, we aim at reproducing the same fine-tuning process as in the original paper [2], but using *HateBERT* model and for a multiclass classification. Thus we defined the two classifications tasks :

1. **Binary classification:** distinguishing between the '*not_hate*' and the '*implicit_hate*' labels (similar to the reference paper).
2. **Multi-class (Tertiary) classification:** distinguishing between the '*not_hate*', '*implicit_hate*', and '*explicit_hate*' labels.

Following the indication of the paper, we first implemented the training process using the same configuration and learning parameters, and tried to replicate as close as possible the training methods mentioned. Thus, we split the data as followed: 60% training, 20% validation and 20% testing. We fine-tuned *HateBERT* with learning rate in {0.5e-5, 1e-5, 2e-5, 3e-5, 5e-5}, epochs in {3,5,10,20} and batch size in {2,8,16,32}.

However due to over-fitted training, we decided to add additional regularization techniques, unmentioned in the reference paper, to prevent over-fitting and hopefully improved our results. Following some guides [6], we introduced dropout, weight decay, gradient clipping and switched the learning-rate scheduler from *linear* to *cosine* decay (often preferred for transformer fine-tuning). We also tried to use smaller learning rate in range of {3e-6, 5e-6}. *Table 1* summarized configuration we used compared to the one of the reference paper.

As presented in the previous section, the small and unbalanced dataset could lead to over-fitting. Thus we also decided to use the LLM model *ChatGPT-o3* for data augmentation. We used it to generate 4'334 data samples for implicit hate and 8'541 data samples for explicit hate, as summarized in Figure 1. No data was generated for the 'not_hate' class because of the already large sample quantity of the original dataset.

It is important to note that the artificially generated data were used exclusively in the training and validation datasets, and not in the testing dataset. Synthetic data could be incorrect or biased and it is important to evaluate the model performance only on accurate and verified annotated data. Thus the testing set, used to determine the model accuracy, only contains data from the *"Implicit hate Corpus"*.

4. Validation

Table 1 summarize the model and training configuration used in the reference paper [2] and in our own pipeline. We used the same configuration for both binary and tertiary classification. These parameters were given overall satisfying results but could be further fine-tuned for each specific tasks.

Set up	Article's baseline[2]	Ours
Batch size	8	16
Epoch	{1,2,3,4}	10
Learning rate	{2e-5,3e-5,5e-5}	5e-6
Weight decay	-	0.05
Dropout	-	0.3

Table 1. Models' configuration

Binary	Baseline	HateBERT	HateBERT with Augmentation
Precision	72.1	**73.3**	72.2
Recall	66	**74.3**	74.3
Accuracy	78.3	75.5	73.3
F1-score	68.9	**73.6**	72.3

Table 2. Models' performance for binary classification

Tertiary	HateBERT	HateBERT with Augmentation
Precision	63.3	59.1
Recall	58.9	59.3
Accuracy	72.3	70
F1-score	60.1	58.7

Table 3. Models' performance for tertiary classification

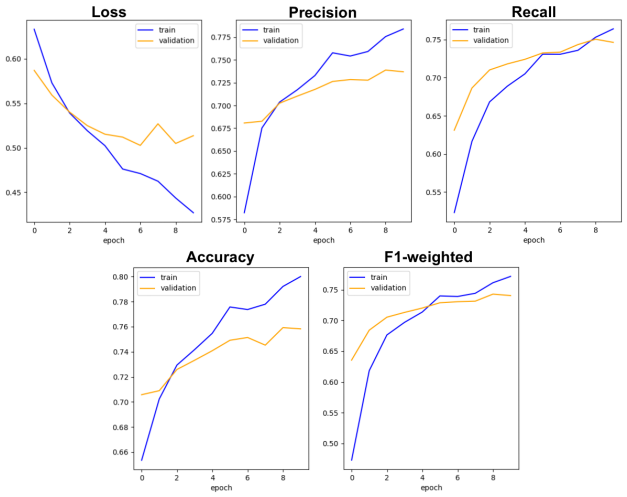

Figure 2. Binary classification: training and validation loss without data augmentation.

Using our pipeline without data augmentation, HateBERT didn't outperform the accuracy of the reference paper baseline [2], however it did achieved better results for the other metrics, as seen in the Table 2. These results actually revealed to be quite promising. Indeed, as explained in Google ML crash course [7], *"when dataset is imbalanced [...] it's better to optimize for one of the other metrics instead"*. F1-score is usually the preferred metric for unbalanced dataset as it *"balances the importance of precision and recall"*. Therefore, *HateBERT* seems to provided more stable predictions than BERT. Additionally, the training plot indicates us that the model did not over-fitted significantly 2, thanks to the regularization techniques added.

Concerning the multi-class classification, we achieved a max accuracy of 72.3% but a F1-score of only 60.1%. As the dataset for tertiary classification is even more unbalanced, F1-score is clearly more relevant than the accuracy. Such low F1-score is due to the consequent lack of explicit hate speech examples, making this class prediction really challenging for the model, as visible in the Table 4.

Metrics	Not hate	Implicit	Explicit
Precision	77.3	69.5	30
Recall	83.0	58.3	48.5
F1-score	80	63.4	37

Table 4. Tertiary classification metrics depending on the class (no augmentation)

For both binary and tertiary classification, the implemented data augmentation didn't help improving the performance and rather lower them. Indeed, the synthetic data represented almost 50% of the entire training-validation dataset. Therefore the model was mostly training on synthetic data rather than real, correctly proven data. And when analysing the nature of the augmented data, we notice a lot of repetitive template phrases and patterns. Using a lower percentage of synthetic data in our training, or using better quality data generation could hopefully enhance the model performance. Making the LLM generate samples of smaller batches could be helpful. We could also use multiple different LLM models to counter act the bias of one's model.

Over-fitting remains a challenge, especially for the binary classification but could be solvable through further parameter tuning, regularization method (k-fold) implementation or adjustments to the model's size and layer structure

5. Conclusion

The objectives of the project were mostly successfully met. Despite a little lower accuracy compared to the paper's model [2], our other metrics for the binary classification without augmentation outperform their results. This is particularly encouraging given the challenges posed by the imbalanced datasets and the sparse data for the implicit class. On the other hand, our data augmentation revealed to be particularly inefficient, biased and leading to lower performances.

Overall, the project provided valuable insights into the trade-offs between model complexity, data augmentation using an existing LLM (Large Language Model), and generalization performance.

References

[1] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)* (A. Mostafazadeh Davani, D. Kiela, M. Lambert, B. Vidgen, V. Prabhakaran, and Z. Waseem, eds.), (Online), pp. 17–25, Association for Computational Linguistics, Aug. 2021.

[2] M. ElSherief, C. Ziems, D. Muchlinski, V. Anupindi, J. Seybolt, M. De Choudhury, and D. Yang, "Latent hatred: A benchmark for understanding implicit hate speech," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 345–363, Association for Computational Linguistics, Nov. 2021.

[3] J. Johnson and T. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[4] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, and M. Ryo, "Ten deep learning techniques to address small data problems with remote sensing," *International Journal of Applied Earth Observation and Geoinformation*, vol. 125, p. 103569, 2023.

[5] J. Serillon, O. Voland, and T. Cirillo, "Ee-559 group 21 project - hate speech detection." https://github.com/Ovoland/Deep_Learning_grp21, 2025. Accessed: 2025-06-09.

[6] A. Sharma, "8 simple techniques to prevent overfitting." https://medium.com/data-science/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d, August 2019. Accessed: 2025-05-25.

[7] Google, "Classification: Accuracy, recall, precision, and related metrics." https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall, 2025. Machine Learning Crash Course, Google for Developers.