Generative LLM oversampling for the class imbalance problem in hate speech detection

Nicolas Antonio Cloutier and Nathalie Japkowicz[1]

1. American University, Washington, D.C., USA

Abstract. Online hate speech has become increasingly prevalent with the rise of social media. As such, methods for automatically detecting and classifying hate speech have become the subject of much research. A common challenge in this and other domains is the class imbalance problem, where one class in a dataset is far more common than another. We propose the use of a generative large language model (LLM), specifically OpenAI's GPT-2, as a method of oversampling text data in order to account for this imbalance, comparing it to other resampling methods on three tasks: binary classification of tweets, classification of antisemitic tweets into types of antisemitism, and a combination of the two. We find that generative LLM resampling does not produce better results for binary classification than other resampling methods, but does improve performance on the other two tasks.

1 Introduction

In recent years, hate speech has become increasingly mainstream and common within social media sites [22]. This rise has not only caused online spaces to become less hospitable, but has also had several offline effects. On top of having severe psychological effects on the recipient [22], online hate speech also played a role in disseminating extremist anti-Rohingya voices leading to violence in Myanmar [11], and has provided motivation for several perpetrators of offline violent hate crimes [22]. These events have motivated responses from numerous parties, including the social media sites themselves, that are increasingly looking to stop the spread of these messages [24], and governmental bodies, that are seeking to regulate or prevent the spread of hate speech [2]. At the same time as hate speech has been becoming more common, social media sites have been generating more and more content, with popular social media site Twitter generating an average of 500 million tweets per day in 2019 [18].

With these developments and the large amount of content on social media sites, these sites have been increasingly looking to automatic detection methods for hate speech [24]. These methods use Machine Learning (ML) to automatically detect and classify hateful speech, removing some of the work done by moderators, whose primary job is to respond reactively to user reports of hate speech [24]. Chandra et al. [4] used a combination of image and text processing and classification algorithms to classify images and text on social media sites Twitter and Gab [4]. We use their Twitter dataset to further analyze the presence of hate speech on Twitter and investigate new algorithms for classification.

One difficulty with this dataset is that it is imbalanced, meaning one classification is far more common in the dataset than another. Imbalanced data can negatively impact the performance of ML classification algorithms [23], affecting numerous domains that use ML classification. Many ML algorithms are inadequately prepared to handle the class imbalance problem [23], leading many to look to other solutions, including resampling methods, that in some way change the training data by adding or removing data points in order to allay the effects of the class imbalance problem [13].

With this in mind, there are two crucial research questions this paper seeks to answer. First: how can automatic ML methods be improved in the domain of text classification for hate speech detection? Second: how can the class imbalance problem be dealt with for text data? These are the questions we seek to provide answers to, with the hope that they may inform future research and hate speech detection systems.

2 Previous work

Antisemitism detection, as well as hate speech detection generally, has been the subject of much research. Martins et al. [17] used ML models to analyze and classify antisemitism in posts on social media sites Gab and 4chan, and González-Pizarro and Zanettou [10] used large language models (LLMs) to do the same. Chandra et al. [4] introduced a new dataset of labeled antisemitic posts, including text and images. These posts were labeled not only for whether or not they are antisemitic, but also their type of antisemitism, with the researchers grouping antisemitic posts into political, economic, religious, and racial antisemitism, and trained models to classify both antisemitic status and type of antisemitism. This Twitter dataset is imbalanced, with the non-antisemitic class far outnumbering the antisemitic class, and with political and racial antisemitism being more common than religious and economic antisemitism.

Imbalance is a common issue in ML classification problems. The class imbalance problem can severely negatively affect model predictive power, with the less common class (termed the "minority class") often being misclassified due to its low prevalence in the dataset relative to the larger class (the "majority class") [1]. This problem has affected fields as distinct as medicine, fraudulent call detection, and risk management [23]. Due to the severity with which this problem can limit model performance and its widespread nature, being seen in numerous distinct fields, it has become a focus of researchers as an area of improvement [1], with numerous techniques being created to allay the effects of the problem.

One such group of techniques is resampling, altering the training data by adding new data points or taking existing data points away in order to improve model performance. Generally, there are two types of resampling: oversampling, which describes the process of adding synthetic data points to a dataset, and undersampling, which describes the process of removing existing data points from a dataset [21]. Resampling techniques can improve the performance of ML models when trained on imbalanced data [16] [14].

One method for oversampling is the use of generative models, including generative adversarial networks (GANs) and autoencoders to generate synthetic data. These methods seek to match the distribution of the original dataset and create synthetic datapoints in accordance with that distribution [12], with the goal of creating authentic synthetic examples for model training. They have achieved success in their applications to primarily computer vision and tabular data [12] [7] [3] [5]. Applications of these generative models to natural language processing (NLP) do occur [19], but they are generally less common than applications to other areas. NLP tends to use more traditional statistical methods for oversampling such as SMOTE and random oversampling [25] [9].

While these advances in resampling have been occurring, similar advances have been made in generative LLMs, such as the GPT series of models from OpenAI, which have the ability to generate human-like text [8] and have been applied to fields such as patent claim generation [15] and healthcare education [20]. These LLMs, despite their ability of producing authentic examples given prompts, have not been widely investigated as a method for oversampling text data.

3 Methodology

Models were trained on three different tasks. The first was a simple binary classification task, where the models would attempt to classify the text of a tweet as either antisemitic or not antisemitic. The second was the 4-class type classification, where the dataset would be limited to only antisemitic samples, that would then be grouped into four classes: political, economic, religious, and racial antisemitism. Finally, the models were trained on a 5-class type classification task, where the dataset included both antisemitic and non-antisemitic samples, and the model would attempt to classify the samples into one of the four groups of antisemitism or classify them as not antisemitic, creating five classes total. Every model was trained and tested separately for all three tasks.

In order to best represent a variety of ML architectures, different algorithms and methods of text representation were used. For algorithms, we trained classifiers utilizing the Naïve Bayes, Extreme Gradient Boosting, Decision Tree, and Support Vector Machine algorithms. We also used three methods of representing text: term frequency-inverse document frequency (TF-IDF), raw frequency, and Bag of Words (BoW). Each model was trained with each method of text representation, creating a total of twelve models that were trained for each task. In order to reduce the dimensionality of the dataset, representations for training and testing data were limited to words that had a frequency score of at least 0.5%, meaning that the word had to have a frequency of at least 0.5% of the original, full length of the text in order to be considered by the models.

The dataset is imbalanced. The following figures show the distribution of different classes in the dataset. The first is for the entire dataset, and the second, containing classifications on types of antisemitism, only contains datapoints that are antisemitic.

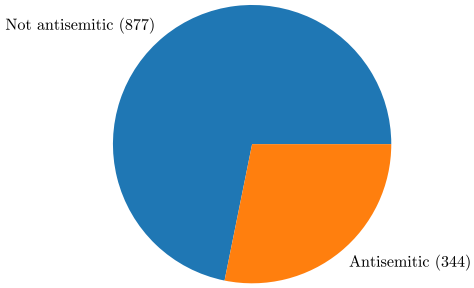

Figure 1. Binary classification distributions

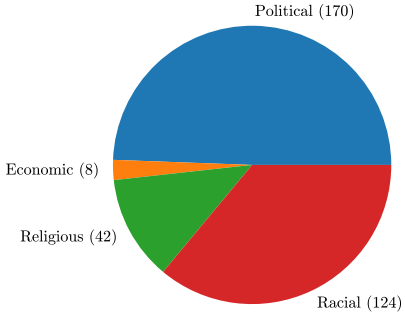

Figure 2. Type classification distributions

Several methods of resampling were used to reduce the impact of the class imbalance. One set of models was trained with no resampling, with additional sets being trained using random undersampling, random oversampling, SMOTE with Tomek Links, ADASYN, and a final set being trained on the augmented dataset generated using the LLM. GPT-2 was used to generate the samples because it is easily available and callable programmatically with the HuggingFace API. The models that were trained using the augmented dataset also used random undersampling on the augmented data.

In regards to testing, each model was tested using 10-fold cross-validation on the original dataset. It was ensured that no samples generated from oversampling methods were used during testing, and additionally that, when testing the augmented model, if a sample that was used to generate more samples appeared in the testing split, the samples generated with it would not appear in the train-

ing split. This ensures that the models were not unfairly advantaged, and that each was tested on a large amount of genuine, unseen data.

For model evaluation, the main metric used was the mean of the recall scores across each of the classes the model had to classify. This was used in lieu of accuracy in order to account for the class imbalance in the data, but accuracy was also tracked for informative purposes. Once the models were evaluated, their answers to the testing samples were converted to a binary matrix with each column representing a model and each row representing a sample. We then used the Cochran's Q-test to test for significant difference in the models, then the Dunn test for post-hoc analysis. When testing the resampling methods against each other, the data were turned into another matrix with each column representing a resampling method and each row representing a model trained with that method, with the value in the cell being the mean recall of that model. A Friedman's χ^2 test was then performed with a post-hoc Nemenyi test to analyze the results.

4 Results

Overall, the models trained on the LLM-augmented data showed no improvement on the binary task, but they did show improvement on the 5-class classification and especially the 4-class classification tasks. The following table contains information for the binary classification task with each resampling method, as well as the best-performing model performance and mean model performance for that resampling method, with "performance" meaning the mean recall score the model achieved. The data used to create the below table received a Friedman χ^2 test statistic of 28.19 and p-value of 3.34×10^{-5}. After the table, a matrix is provided with the results of the post-hoc Nemenyi test on the resampling methods, with p-values displayed in the array. A higher p-value and lighter color means a greater degree of similarity between two models' answers and performance, while a lower p-value and darker color means a lower degree of similarity. In this array, "RU" stands for random undersampling, "ST" stands for SMOTE with Tomek links, "RO" stands for random oversampling, "AD" stands for ADASYN, "NO" stands for no resampling method, and "LA" stands for LLM-augmented.

Table 1. Binary task resampling method performance

Resampling Method	Best Score	Mean Score
Random Undersampling	0.696	0.640
SMOTE-Tomek	0.670	0.612
Random Oversampling	0.686	0.624
ADASYN	0.660	0.606
None	0.667	0.613
LLM-Augmented	0.679	0.617

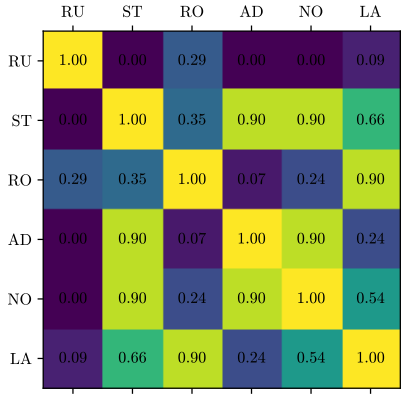

Figure 3. Binary Nemenyi test results

In the binary task, the best performing resampling method both in mean score and best score was random undersampling, which was also found through the Nemenyi analysis to be the least similar to the other models. Overall, it seems as if the augmented data had no positive impact on the binary classification task, and that in this case simple and traditional statistical methods are better than generative LLM oversampling. The generative LLM did beat some oversampling methods (SMOTE and ADASYN), and was better than no resampling at all, but was beaten by simple random oversampling, suggesting an inappropriateness of the method to this task.

The results of the 5-class classification task are shown below. The data used to create the below table received a Friedman χ^2 test statistic of 29.19 and p-value of 2.13×10^{-5}.

Table 2. 5-class task resampling method performance

Resampling Method	Best Score	Mean Score
Random Undersampling	0.300	0.260
SMOTE-Tomek	0.306	0.276
Random Oversampling	0.371	0.298
ADASYN	0.316	0.251
None	0.312	0.264
LLM-Augmented	0.402	0.347

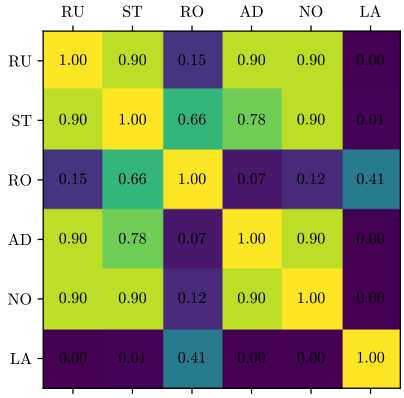

Figure 4. 5-class Nemenyi test results

These 5-type results paint a better picture of the potential of generative LLM oversampling. This time, it is the LLM augmented models that are most different from the other models, with these models also achieving the highest best and mean scores. In the 5-class problem, the random undersampling that was previously the best choice ends up having the second worst mean score and the single worst best score, overall producing worse models than no resampling at all. Oversampling methods such as SMOTE, ADASYN, and especially random oversampling, fared better, but they were still beaten by the generative LLM.

The results of the 4-class classification task are shown below. The data used to create the below table received a Friedman χ^2 test statistic of 23.14 and p-value of 3.16×10^{-4}.

Table 3. 4-class task resampling method performance

Resampling Method	Best Score	Mean Score
Random Undersampling	0.426	0.339
SMOTE-Tomek	0.438	0.366
Random Oversampling	0.405	0.351
ADASYN	0.471	0.319
None	0.348	0.323
LLM-Augmented	0.509	0.433

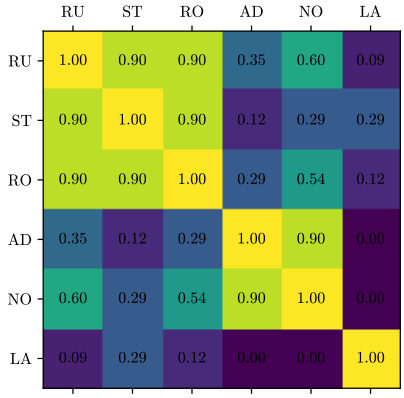

Figure 5. 4-class Nemenyi test results

Again in the 4-class classification, the models trained on LLM-augmented data showed the best performance, and by a wider margin than in the 5-class task. In the 5-class task, there was a difference of 0.049 in the score of the LLM augmentation and the second place model, and that margin increases to 0.067, more than a 33% increase in margin. Again, looking at the Nemenyi analysis results, the models trained with LLM-augmented data appear to be the odd men out, having the least in common with other models. An interesting portion of the results given is the stark contrast between the best and mean scores for the ADASYN models. This is due to a single model, the Naïve Bayes model trained with the Bag-of-Words representation, significantly outperforming the other ADASYN models.

One initial possible explanation for the high performance of the models with LLM-augmented data relative to other models on the 4- and 5-class tasks is that the LLM augmentation provides performance improvements with only one particular class that other resampling methods did poorly on, artificially raising the mean recall even though there was only one class affected significantly. Looking further at the data, however, while some classes were more affected than others, overall, the models trained on LLM-augmented data show improvements in multiple classes. On the 4-type classification task, the following table shows the mean recall of each method on each particular class.

Table 4. 4-class task class-wise resampling method recall

Resampling Method	Political	Economic	Religious	Racial
Random Undersampling	0.355	0.250	0.450	0.196
SMOTE-Tomek	0.511	0.135	0.460	0.246
Random Oversampling	0.482	0.134	0.448	0.232
ADASYN	0.568	0.125	0.219	0.264
None	0.622	0.096	0.193	0.280
Non-LLM mean	0.508	0.148	0.354	0.245
LLM-Augmented	0.492	0.230	0.471	0.405

The performance of the models trained on LLM-augmented data is roughly consistent with the performance of other models on political antisemitism, although it is slightly lower than the average, but it outperforms the other models in every other class. It is expected for a resampling method that the smallest classes be the most affected, as its purpose is to account for the imbalance of classes.

5 Discussion

The most striking result of these trials is the difference in relative performance among the different tasks. Given that a class imbalance exists in all tasks, it may be expected that the generative LLM oversampling would produce similar results for each of these imbalances, but this is not the case. Furthermore, the difference in performance of the generative LLM-trained models to the models trained on data from other resampling methods grows from the 5-class to the 4-class task, suggesting that there is something particular to the binary classification problem that the generative LLM is not as good at accounting for as other resampling methods.

One possible reason for this is the relative performance of undersampling and oversampling methods in general. Generally, oversampling methods tend to, at best, perform as well as undersampling methods [6], explaining why the only pure undersampling method scored best on the binary classification task. That being said, in tasks where undersampling would leave a dataset very small, this pattern may reverse. In the case of the 4-class classification task, one of the classes has only 8 observations, meaning, during undersampling, all other classes would also be reduced to 8 samples, making a very small dataset, that is perhaps inadequate for classification, providing one possible explanation for why oversampling outperformed undersampling on the 4-class and 5-class problems. As for why the generative LLM specifically outperformed other oversampling methods, it may simply be that generating text and then converting it into a vectorized form produces more authentic results than generating new samples from those vectorized forms with methods such as ADASYN and SMOTE. This would explain the augmented LLM's outperformance of other oversampling methods on the 4-class and 5-class tasks, but would leave unexplained why it was outperformed by random oversampling in the classification task.

In order to reach a better understanding of the applications of generative LLM oversampling, more research should be done on the abilities of the method to account for class imbalances in text data. Hopefully, future research that applies the method to datasets with differing sizes, levels of severity of imbalance, and number of classes to classify will help pinpoint the exact situations where generative LLM oversampling can be most helpful, informing future systems that may use the method.

6 Conclusions

This paper proposes the use of generative LLMs to oversample imbalanced text datasets in order to account for the imbalance. Training models with different

resampling methods, we find that results are unequal among tasks. The models trained on generative LLM data showed no improvement over other models in the binary classification of social media posts into an antisemitic/not antisemitic dichotomy, but they showed great improvement on the classification of tweets into types of antisemitism. Further research on imbalanced datasets with different characteristics is required to understand why the difference in relative performance was so stark and what factors most contribute to the utility of this method to accounting for the imbalance problem.

References

1. Abd Elrahman, S. M., Abraham, A: A review of class imbalance problem. Journal of Network and Innovative Computing **1**, 332–340 (2013)
2. Banks, J.: Regulating hate speech online. International Review of Law, Computers and Technology **24**(3), 233–239 (2010)
3. Bellinger, C., Japkowicz, N., Drummond, C.: Synthetic oversampling for advanced radioactive threat detection. IEEE CONFERENCE ON MACHINE LEARNING AND APPLICATIONS 2015.
4. Chandra, M., Pailla, D., Bhatia, H., Sanchawala, A., Gupta, M., Shrivastava, M., Kumaraguru, P.: "Subverting the Jewtocracy": Online Antisemitism Detection Using Multimodal Deep Learning. ACM WEB SCIENCE CONFERENCE 2021, pp. 148-157.
5. Dai, W., Ng, K., Severson, K., Huan, W., Anderson, F., Stultz, C.: Generative oversampling with a contrastive variational autoencoder. IEEE INTERNATIONAL CONFERENCE ON DATA MINING 2019, pp. 101-109.
6. Drummond, C., Holte, R. C.: C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. WORKSHOP ON LEARNING FROM IMBALANCED DATASETS 2003.
7. Engelmann, J., Lessmann, S.: Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications **174**, (2021)
8. Floridi, L., Chiriatti, M.: GPT-3: Its nature, scope, limits, and consequences. Minds and Machines **30**, 681–694 (2020)
9. Glazkova, A.: A comparison of synthetic oversampling methods for multi-class text classification. *arXiv preprint arXiv:2008.04636t*, (2020)
10. González-Pizarro, F., Zannettou, S.: Understanding and detecting hateful content using contrastive learning. *arXiv preprint arXiv:2201.08387*, (2022)
11. Green, P., MacManus, T., De la Cour Venning, A.: Countdown to Annihilation: Genocide in Myanmar. International State of Crime Initiative, London (2015)
12. Hao, J., Wang, C., Zhang, H., Yang, G.: Annealing genetic GAN for minority oversampling. *arXiv preprint*, (2020)
13. Japkowicz, N., Shaju, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis **6**(5), 429–449 (2002)
14. Khushi, M., Shaukat, K., Alam, T. M., Hammed, I. A., Uddin, S., Luo, S., Yang, X., Consuelo Reyes, M.: A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data. IEEE Access **9**, 109960–109975 (2021)
15. Lee, J. S., Hsiang, J.: Patent Claim Generation by Fine-Tuning OpenAI GPT-2. World Patent Information **62**, (2020)

16. Lee, P. H.: Resampling methods improve the predictive power of modeling in class-imbalanced datasets. International Journal of Environmental Research and Public Health **11**(9), 9776–9789.
17. Martins, R., Gomes, M., Almeida, J. J., Novais, P., Henriques, P.: Hate speech classification in social media using emotional analysis. BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS 2018, pp. 61-66
18. Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., Camacho-Collados, M.: Detecting and Monitoring Hate Speech in Twitter. Sensors **19**(21), 4654–4691 (2019)
19. Phung, N. M., Mimura, M.: Evaluation of a cGAN Model and Random Seed Oversampling on Imbalanced JavaScript Datasets. Journal of Information Processing **30**, 591–600 (2022)
20. Sallam, M.: The utility of ChatGPT as an example of large language models in healthcare education, research and practice: Systematic review on the future perspectives and potential limitations. *medRxiv preprint*, (2023)
21. Shelke, M. S., Deshmukh, P. R., Shandilya, V. K.: A review on imbalanced data handling using undersampling and oversampling technique. International Journal of Recent Trends in Engineering Research **3**(4), 444–449 (2017).
22. Siegel, A.: Online Hate Speech. Social media and democracy: The state of the field, prospects for reform, 56–88 (2020).
23. Sun, Y., Wong, A., Kamel, M.: Classification of Imbalanced Data: A Review. International Journal of Pattern Recognition and Artificial Intelligence **23**(4), 687–719 (2009)
24. Ullmann, S., Tomalin, M.: Quarantining online hate speech: technical and ethical perspectives. Ethics and Information Technology **22**, 69–80 (2020)
25. Wijaya, I. D., Putrada, A. G., Oktaria, D.: Overcoming Data Imbalance Problems in Sexual Harassment Classification with SMOTE. International Journal on Information and Communication Technology **8**(1), 20–29 (2022).