Integrated Capstone Project: Analyzing Hate Speech Context and Content

Christopher S. Fornesa

Technical Report on Analyzing Hate Speech Context and Content

Technical Report: Executive summary

Hate speech promotes political polarization through its adverse impacts on online discourse, deeming it necessary to quantify and classify hate speech and identify the key determinants (features) of hate speech and toxic online content. The goal of modeling was to predict hate speech, abuse level, and text toxicity using metadata features and text embeddings (which are numerical representations of text data) from several datasets containing social media or conversational data with features related to hate speech and sentiment. Here, classifier models predicted hate speech, abuse level, the number of target groups impacted, and toxicity level. Classification was performed by subjecting the original features (after feature selection) or sentence embeddings to standard scaling and dividing them into training and testing sets. Modeling on the training set was performed to predict toxic levels, hate speech, abuse levels, or target groups. Predictions using the testing set were then evaluated to determine the best model for each dataset. These yielded 70% (hate speech) to 99.8% (toxic class) accuracy and 70% (hate speech) to 99.8% (toxic class) accuracy per class (see Table 6 in the Technical Report: Results section).

These experiments confirmed that sentiment is a crucial determinant in predicting hate speech, enabling platforms to weigh the values of text sentiment analysis for effective content moderation. Metadata signals, such as the number of parent tweets associated with toxic tweets (as in the Online Abusive Attacks dataset), could be used to predict post toxicity. Candidate sentiment (as in the Hate Speech 2020 U.S. Elections dataset) can be used to predict hate speech, while more complex feature sets (such as those involving annotator or post sentiment, as in the Dynamically Generate Hate Speech dataset) can gauge the predictive power of their impact on

the target group. Regarding broader impacts, entities can collect data similar to the Dynamically Generated Hate Speech or the Hate Speech 2020 U.S. Elections datasets to train chatbots or hate speech detection models, and data similar to the Online Abusive Attacks dataset to determine whether content is toxic to gauge a candidate's association with hate speech or a model's ability to detect toxicity. If enacted, these measures could keep social media and chatbot users safer by preventing their exposure to toxic speech and, in turn, political polarization.

Technical Report: Introduction and problem definitions

With the rise of social media and generative artificial intelligence (also known as Gen AI), algorithmic political bias has garnered high interest. Algorithmic political bias refers to instances where AI systems inadvertently filter users based on their political orientation or identity, thereby creating social media bubbles. For example, campaign staff may want to know the qualities attributed to their candidate (e.g., kindness), which would require sentiment analysis of social media data that mentions their candidate (e.g., direct social media mentions). Based on these results, campaign staff can strategize to improve the appearance of less desirable qualities in upcoming campaign events and debates. Predicting toxicity levels can improve content moderation and evaluate the fine-tuning results of large language models. For instance, content moderation may use toxicity to predict whether a post contains extremely toxic content, especially when a threshold is applied. Similar strategies can be applied to fine-tune large language models and evaluate chatbot outputs for toxic content.

For classifying hate speech, assessing abuse levels, and determining the number of target groups impacted, the most relevant measures were accuracy (the success rate of classification), weighted F1 (weighted success rate by category), macro F1 (unweighted), and per-class F1 (specific to each label) scores. Training scores were used to tune models, while testing scores

evaluated predictive capacity on new data. Feature (including permutation) importances, which measured the predictive significance of features, helped determine the most relevant features. These are relevant to the Convabuse, Dynamically Generated Hate Speech (DGHS), Hate Speech in the US 2020 Elections (US Elections), and Multilingual and Multi-Aspect Hate Speech (MLMA) datasets. Together, these provide insights for questions, such as: “What are the top contributing factors to hate speech?” and “Do characteristics, like sentiment, exacerbate hate speech?” Both were answered using feature importances in the best model for the DGHS dataset.

Meanwhile, modeling using sentence embeddings (as SentenceBERT embeddings contain 384 dimensions) provided the best answers to questions at the core of this analysis. These questions included: “How many groups are impacted by the sentiment in this text?”, relevant to the MLMA dataset, and “What level of abuse is associated with this text?”, specific to the Convabuse dataset. Shmulewitz et al. stated that “Greater frequency of hate speech was significantly associated with increased PTSD symptomology” (2025). Feature importances alone could not provide adequate analysis for the sentence embeddings, so the SHAP (Shapley Additive exPlanations) method provided a summary plot to determine the impact of the most influential dimensions on each target label, in addition to feature importance (Yadav, 2024).

Root mean squared error, R-squared score, and feature coefficients were used for feature selection for the Online Abusive Attacks (OAA) dataset. (Ahmed et al., 2022) Then, accuracy, F1 scores, and feature importances were evaluated to assess each model’s ability to determine the toxic class (based on toxicity intervals). Together, these results answered the question, “Does a post’s number of toxic parent tweets or toxic replies indicate higher levels of toxicity?” using feature importance data from the best models for the OAA dataset. This was essential as Chitra and Musco stated that “chronological ‘news feeds’ on social media have… been replaced with

individually filtered and sorted feeds" to increase engagement. (2020) Finally, these metrics ensure that the best models have sufficient power to predict new data into each category.

Technical Report: Data Overview

The data used in this analysis consisted of labeled classification datasets collected from outputs of the Eliza and CarbonBot chatbots (Convabuse), social media posts about the 2020 U.S. Election (US Elections), keywords from social media (MLMA), using synthetic outputs to train content classification algorithms (DGHS), and identifying targets and keywords to harvest data (OAA). Models for the Convabuse dataset predicted one of five abuse levels, while those for the DGHS and US Elections datasets classified hate speech. In contrast, models for the MLMA dataset classified the target group, and those for the OAA dataset classified toxicity.

Cercas Curry et al. collected data to train chatbots (2021). The cleaned Convabuse dataset consisted of 12,768 rows and 10 columns (8 context, four combined into 1 for text, and one target), with no missing data. Table 1 contains the data dictionary. Limitations of this dataset included the subjectivity of annotators' perspectives in identifying sentiment. Because text was derived from chatbots, modeling applications are limited to cases such as training chatbots or other uses of evaluating large language models. Discrepancies, like the presence of integer-encoded columns for each targeted group, were present. Table 1 contains the data dictionary.

Table 1

Data dictionary for the Convabuse dataset.

Column	Type	Representation
annotator_id	Categorical integer (1 to 8)	Which annotator labeled this response?
bot	Categorical integer (0 or 1)	Which chatbot outputs this response?

Column	Type	Representation
system	Categorical integer (0 or 1)	Is systemic targeting involved?
explicit	Categorical integer (0 or 1)	Does the text explicitly target a group?
implicit	Categorical integer (0 or 1)	Does the text implicitly target a group?
abuse_level	Categorical integer (1 to 8)	What was the degree of abusive sentiment?
target_groups	Categorical integer (0 or 1)	Were any groups targeted?
text	Text string	Combined agent and user text.

Note: 5 categorical integer columns were numerical representations of booleans (true or false).

Vidgen et al. (2021) trained annotators to generate and label text data for the DGHS dataset. The cleaned and imbalanced DGHS dataset consisted of 37,938 observations and six columns (4 context, one text, one target). No missing data was found. Limitations of this data included bias by annotators and the dynamic nature of the text. Discrepancies included the presence of an original column with excessive labels. Table 2 contains the data dictionary.

Table 2

Data dictionary for the Dynamically Generated Hate Speech (DGHS) dataset.

Column	Type	Representation
label	Categorical float (0 or 1)	Is this content hate speech?
type	Categorical integer (-1 to 1)	Is the sentiment negative, neutral, or positive?
annotator	Categorical integer (1 to 20)	Which annotator labeled this text?
target_groups	Categorical integer (0 or 1)	Were any groups targeted?
original	Categorical integer (0 or 1)	Is this content original or permuted?
text	Text string	This is text derived from a social media post.

Note: “type” was associated with negative (-1), neutral (0), or positive (1).

For the Online Abusive Attacks (OAA) dataset, Alharthi et al. included columns for counts of user and post metadata for context modeling (2023). The preprocessed OAA dataset consisted of 2,313 observations and three columns, representing the counts of toxic parent posts, toxic replies, and toxic level. This dataset contained post metadata for parent post metrics and child post metrics, condensed to the two shown. This dataset contained post metadata for parent post metrics and child post metrics, condensed to the two shown. There were no text columns and no missing data due to prior preprocessing. Excessive targets were also found. Limitations of this dataset include potential sampling bias in the data collection and the time period (circa 2023). Additionally, most observations were from X (formerly Twitter), which limits their applicability. Table 3 contains the data dictionary after addressing inconsistencies.

Table 3

Data dictionary for the Online Abusive Attacks (OAA) dataset.

Column	Type	Representation
Toxic parent tweets	Float (ranges 0 to 804)	How many toxic parent tweets were associated with this content?
Toxic replies	Float (ranges 0 to 549)	How many toxic replies were associated with this content?
Toxic Level	Categorical integers (0 to 4)	What was the degree of abusive sentiment?

Note. All columns were numerical integers, but the two features were represented as floats.

Grimminger and Klinger captured social media data from the 2020 U.S. Presidential Election for the US Elections dataset (2021). The model used 3,000 observations and three columns for context modeling. This dataset had no missing data, no text data, and the only inconsistency was that all values for the “West” column were 0 (and were omitted from the data

dictionary in Table 4. Political stability and the dubiousness of gauging sentiment for each candidate may also impact generalization on new data.

Table 4

Data dictionary for the Hate Speech 2020 U.S. Election (US Election) dataset.

Column	Type	Represents
Trump	Categorical integer (-1 to 1)	The measured sentiment towards Trump.
Biden	Categorical integer (-1 to 1)	The measured sentiment towards Biden.
HOF	Categorical integer (0 or 1)	Is this content hate speech?

Note. Sentiment was represented as -1 (negative), 0 (neutral), and 1 (positive)

The Multilingual and Multi-Aspect (MLMA) Hate Speech dataset contained eight missing entries, with text in English, Arabic, and French, with columns for sentiment, text content, directness, group, and target (Ousidhoum et al., 2019). This resulted in six columns (four context, one text, and one target) and 18,326 observations. Tweets were in French, Arabic, and English, which was a possible inconsistency, as modeling using sentence embeddings from this dataset excluded French and Arabic text. Otherwise, there were eight entries with empty values for sentiment (addressed later), while limitations may include the trainability of models that depend on specific languages for multilingual data. Inconsistencies were present in the numerous sentiment columns. Table 5 displays the columns after inconsistencies were addressed.

Table 5

Data dictionary for the Multilingual Multi-Aspect Hate Speech (MLMA) dataset.

Column	Type	Represents
implicit	Categorical integer (0 or 1)	Is a group implicitly targeted?
explicit	Categorical integer (0 or 1)	Is a group explicitly targeted?

target_groups	Categorical integer (0 to 3)	What is the target impact?
annotator_sentiment	Categorical integer (0 to 9)	+2 (shock, anger, disgust, or fear) +1 (sadness, indifference, or disgust).
abuse_level	Categorical integer (0 to 4)	What is the degree of abuse?
tweet	Text string	This is text derived from each tweet.

Note. Annotator sentiment was calculated using integer encodings for relevant original columns. The "sentiment", "annotator_sentiment", "target", and "group" columns aided in understanding target group impact and sentiment (the scope of impact on targeted groups).

Other limitations included the limited interpretability of sentence embeddings for the text in the Convabuse, DGHS, and MLMA datasets, which was addressed with SHAP. Ethical considerations included the handling of potentially identifying information for posts and accounts (Online Abusive Attacks dataset), which was addressed by removing the relevant columns. Feature engineering was also a concern since data loss could interfere with a model's ability to determine underlying non-linear relationships. Finally, nearly every target variable was imbalanced to various degrees, which is why rebalanced datasets would perform better than imbalanced datasets or imbalanced data used to train models using simple reweighting.

Technical Report: Data cleaning, preprocessing, and exploratory data analysis

Upon data exploration, the following findings were noted. For the Convabuse dataset, there was a relatively uniform balance in annotator labels, with more text from CarbonBot than Eliza. 14.8% of entries had a systemic target, while 11.87% had explicit targets and 2% had implicit targets. The most common abuse level was 0, and most content had no target. In the DGHS dataset, the label had a mean of 0.5366, while the type had a mean of -0.3629, indicating a skew towards negative sentiment and a balance between hate speech labels. The first seven annotators annotated most of the text; 24.4% of entries revealed a targeted group, and 63.52%

were original. For the OAA dataset, with a mean of 18.08 and a median value of 0, most tweets had no toxic parents, while some had many toxic parent tweets—a mean of 4.954 and a third quartile of 0 points to a similar trend for toxic replies. Toxic levels had more than 0 or 1 values, with a mean of 0.7635. In the US Election dataset, the mean of 0.0572 and the first quartile value of -1 indicated that more negative sentiment was associated with Trump compared to the mean of 0.1922 and the first quartile of -1 for Biden. The mean of 0.1177 and the third quartile of 0 indicate that most text was not hate speech. For the MLMA dataset, the mean of 0.9967 and the third quartile of 1 indicate that most entries targeted a single group, while a mean of 1.2827 and the third quartile value of 1 indicate that 1 was the most common abuse level.

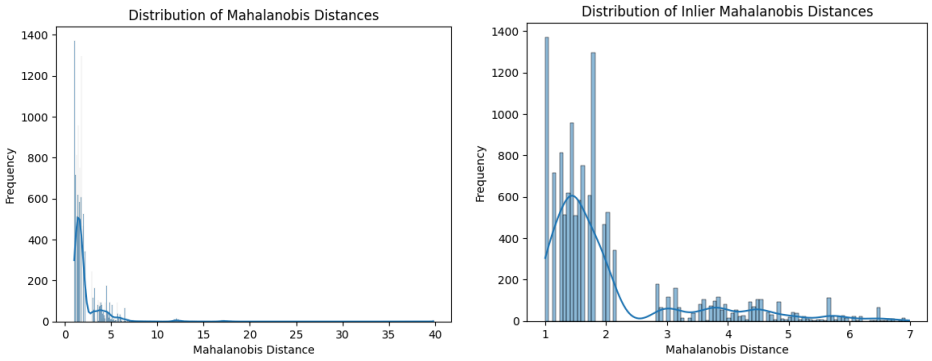
Removing missing or irrelevant values, extreme outliers, and inconsistent entries (such as duplicate entries) also helped increase the integrity of each dataset for hate speech prediction tasks. An example of these actions was the “not given” and “notargetrecorded” string values in the DGHS dataset’s “target” column. These inconsistencies were handled by replacing these values with a “none” string. Only the MLMA Hate Speech dataset was subject to data removal, in which eight observations with NaN values were removed. Outlier removal based on Mahalanobis distance reduced the variance for targets (Figure 1).

Feature engineering on the Convabuse data addressed the existence of integer-encoded columns for each targeted group by combining these columns into a single column that indicated the targets. Feature engineering on the DGHS dataset addressed the presence of an original “target” column with excessive categories by combining these features to indicate the presence of a target and a similar scheme for abuse level. Similar to the Convabuse dataset, target group columns in the OAA data were consolidated via feature engineering into the “target_groups” column. For the US Elections data, the “West” column was removed. Moreover, similar to the

Convabuse dataset, abuse level and annotator sentiment were consolidated in the MLMA data using sentiment columns for each user and each annotator, while also consolidating target groups.

Figure 1

Mahalanobis distribution of the Convabuse dataset based on outliers using standard deviations.



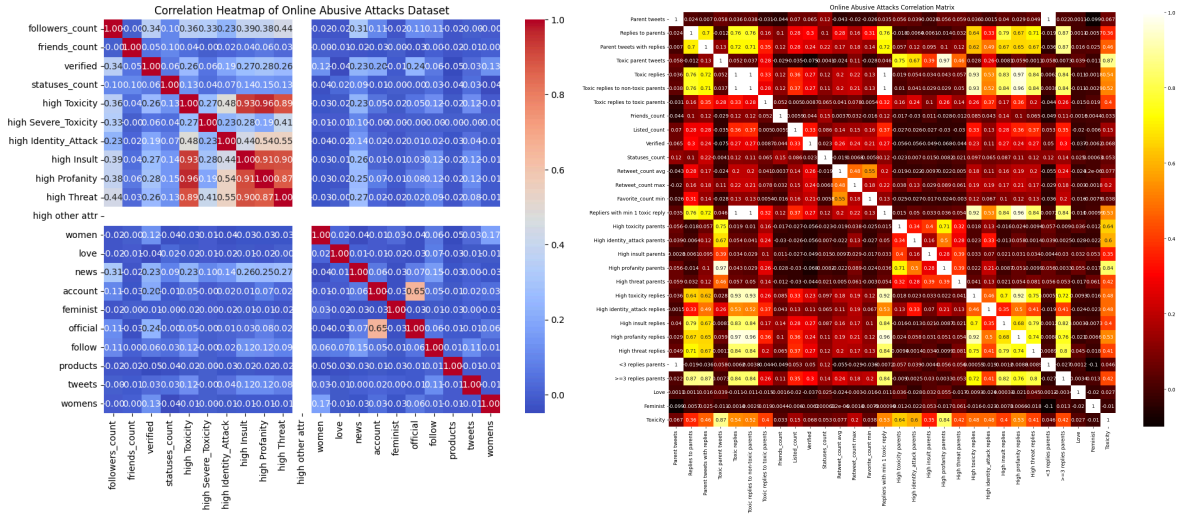
Note. This figure illustrates the Mahalanobis distance of the "abuse_level" variable before and after applying a threshold of 7 as the basis for outlier removal in this dataset.

These crucial steps helped to uncover underlying trends in the data, once ambiguous variables and noisy outliers were removed, resulting in increased correlations, as seen in Figure 2. Otherwise, slight to significant improvements were observed when applying outlier detection and removal using the Mahalanobis distance distribution for each respective target variable, as evaluated by linear regression and decision tree classification, to assess effectiveness. Correlation matrices created during feature selection revealed that certain features did not improve model performance, while the sentiment columns were highly correlated (Figure 3). As a result, the sentiment columns of "high Toxicity", "high Severe_Toxicity", "high Identity Attack", "high Insult", "high Profanity", and "high Threat" were consolidated into a single "toxicity" variable. Ridge regression was further reduced, and a standardized loss function was used to filter out

unimportant columns, until it was found that counts for toxic parent posts and toxic replies were sufficient.

Figure 2

The correlation matrix of the raw vs cleaned and preprocessed Online Abusive Attacks dataset.



Note. Except among the sentiment columns, there were very few correlations that could have indicated a single target variable. The second matrix showed highly multicollinear columns.

For feature engineering, the “type” column of the DGHS dataset (originally containing columns for explicitly targeted groups) was expanded into multiple columns for each observation using one-hot encoding. Then, the resulting figures were summed into a single “target_groups” column to count the total number of target groups for each observation. The same operations were performed for the MLMA Hate Speech dataset to condense the contents of these columns into a single “target_group” calculation; a similar process was used to create abuse levels for the Convabuse dataset where the original “is_abuse” columns (indicated by integers: 1, 0, -1, -2, -3, in order of severity) were consolidated into the “abuse_level” column through one-hot encoding, ranging from 0 to 4. These changes also resulted in improved correlations.

Sentence embedding vectors were created using the SentenceBERT tokenizer on the text content of the Convabuse, DGHS, and MLMA Hate Speech datasets, each with 384 dimensions, to identify similar text content. Most importantly, using actual text content is a more objective measure of analyzing content sentiment. This is in comparison to the bias inherent when paid or volunteer annotators gauge sentiment or the biases that business domain context can impose in feature selection. Sentence data is also more accessible, as data collection would only need to be converted into embeddings before training, eliminating the need for manual labeling. The MLMA dataset also included French and Arabic entries, which were removed to ensure model integrity for embedding models, although they were not removed during context modeling.

Feature and stepwise regression, such as ridge regression, were used in feature selection alongside linear regression, where each feature regression strategy handled loss and dealt with multicollinearity. Meanwhile, forward and backward selection strategies were employed to model progress, implemented using linear regression. Although all these strategies were employed in the feature selection process for the OAA dataset, ridge regression proved to be the most successful initial strategy in removing features while maintaining a high R-squared score. After the initial round of ridge regression, which reduced the number of features from 50 to approximately 30, ridge regression was applied two more times, further lowering the dimensionality to fewer than 10 features. Domain knowledge, evaluating feature coefficients, and assessing correlations for the remaining features were additional techniques employed to eliminate highly multicollinear columns and retain only the “Toxic parent tweets” and “Toxic replies” columns for predicting toxicity. The toxicity scores were then segmented into bins to create the “Toxic Level” target variable. Optimizing for accuracy and F1 scores, along with

analyzing feature importances, led to applying feature reduction for the best model of the Convabuse dataset by removing those with importances below 0.1.

Data cleaning, preprocessing, and EDA were ongoing during model evaluation to ensure optimal performance for each model using these metrics, as the presence of outliers lowered performance, while too many columns obfuscated feature importance results for previous experiments. SHAP also provided dimensionality reduction by interpreting the influence of each dimension in predicting the target label, while importances determined the most important features. Before training, standard scaling was applied to the context features, whereas scaling already occurred during the embedding process. Model choices included linear logistic regression and support vector machine models, as well as non-linear tree-based, ensemble, distance, and neural network models, to gauge underlying linear or non-linear trends. Neural networks and random forests were the best models, as all data exhibited non-linear trends, as evidenced by the findings of imbalanced target variables, as seen in both feature and target skews. Ultimately, from the original five datasets, eight different datasets were created, using either the metadata features (5 datasets) or the sentence embeddings (3 datasets).

Technical Report: Modeling/Analysis

Classification modeling involved linear models (such as linear and logistic regression), decision trees, ensemble models (like random forest), distance-based models (like k-nearest neighbors), and neural network models. Linear models were used to capture linear relationships in the different datasets, particularly for the original features of the OAA, Convabuse, and US Elections datasets. Decision trees analyzed potential outcomes to facilitate classification modeling by assessing the likelihood that an observation would fit into a single target classification. Hence, linear regression and decision tree classification were employed to

evaluate the effectiveness of outlier removal. Ensemble models, built on the logic of decision trees by training and using ensembles of these estimators, proved to be the most useful for the original features of the DGHS dataset. Distance models utilized measures of distance, such as Euclidean distance, for classification tasks. In contrast, simple neural network models used a feed-forward architecture to process data into the different categories of the target variable.

An 80-20 train-test split, along with stratification based on the respective y train set and a random seed of 42, ensured that training and testing models depended on different sets of data without leakage. The original feature data was also subject to standardization (using standard scaling) to reduce variability among the features. K-fold cross-validation with 2 or 5 folds (optimized for computational efficiency where relevant) ensured that the training data was divided into these respective folds, thereby increasing accuracy. It weighted the F1 score by training on different random samples. Additionally, a 30% random sample was used for modeling on the Convabuse and MLMA datasets for models where computational expense was a moderate issue.

In comparison, only a 1% random sample was used for gradient boosting on each dataset, due to its high computational expense. For modeling on embeddings, all embeddings (except for non-English text) were used in training models for sentiment analysis. However, an 80-20 train-test split, random seed of 42, stratification on the respective y train set, and K-fold cross-validation were implemented during the training process. As standardization already occurred in the embedding process, StandardScaler was unnecessary for sentence embeddings.

As mentioned above, evaluation for classification modeling involved accuracy and weighted, macro, and per-class F1 scores. Accuracy was a primary metric for evaluating the overall ability of a model to classify content into specific categories of hate speech, toxic level,

and abuse level, as well as the number of target groups (target group impact) associated with a piece of content. The weighted F1 score evaluates a model's ability to classify each category correctly after rebalancing, while the macro and per-class F1 scores do so without rebalancing. The latter are ideal for evaluating hate speech, as incorrectly classifying hate speech as non-hate speech means that potentially harmful content may continue to be spread on a social media platform, while incorrectly classifying harmful content as having a lower abuse level, toxic level, or target group impact underscores the harm that the spread of such content can cause. With these sets of metrics for classification modeling, training, and testing scores are yielded; however, only the testing scores are recorded. Also, a higher score for one is usually matched with a higher score for the others. For instance, models yielding higher test accuracy also tend to produce the highest weighted F1 score. Therefore, in cases where the test weighted F1 score may be suspiciously low, using the model with the highest test F1 score for a dataset typically yields a high test accuracy score. However, macro and per-class F1 scores were the final mode of evaluation to ensure predictive accuracy per class. In cases such as the MLMA dataset, higher accuracy and weighted F1 scores were achieved. Still, macro F1 scores remained below 0.5, as modeling had completely ignored predictions for one or more target labels. Using macro and per-class F1 scores addressed this issue, even though the "best" model yielded lower accuracy.

In predicting class labels for hate speech, abuse level, and target group impact, hyperparameter tuning and other optimizations were crucial for optimizing the performance of logistic regression, support vector machines, random forests, k-nearest neighbors, and gradient boosting models. Each model was tested using imbalanced data, manual upsampling, automated upsampling with SMOTE (Synthetic Minority Oversampling Technique), and class weight rebalancing on the datasets, where each method was applicable and could be utilized.

Imbalanced data, at times, yielded good accuracy and weighted F1 results, but often had lackluster macro and per-class F1 scores because of overpredicting on majority classes. Thus, only one model, a neural network on the Convabuse dataset, utilized imbalanced data to yield the best results for any dataset. Meanwhile, upsampling rebalanced datasets in different ways. Manual upsampling involves adding new data to existing data in place, where the existing data already exists. SMOTE adds new data to "fill in" the gaps between existing data, using k-nearest neighbors and distance averaging to determine the value and placement of each data point. This addresses the issue of variance by providing more structure to each dataset. Not only did these strategies yield decent to good accuracy and weighted F1 results, but they also led to the highest macro and per-class F1 scores for each dataset for most models: a neural network trained on SMOTE-rebalanced data for the OAA and MLMA datasets, and a random forest trained on manually rebalanced data for the DGHS and US Elections datasets.

For decision trees, optimizations were applied for maximum depth, minimum samples per split, minimum samples per leaf, and the criterion (specified for regression or classification). However, to prevent overfitting in decision trees, only the log2 of the length of each dataset, a min_samples split of 2-5% of each dataset, and a min_samples_leaf of 1-5% of each dataset were implemented. For random forests, these hyperparameters, as well as the number of estimators (up to 100), were optimized using similar thresholds as those for decision trees, since random forests utilize decision tree estimators in their ensembles. In both cases, these optimizations improved generalizability on new data and predictive capacity. Meanwhile, for gradient boosting, the number of estimators, learning rate, and maximum depth and minimum samples per leaf were optimized to avoid overfitting during training. For contextual and sentiment analysis, hyperparameter tuning was conducted for classification on the Convabuse, DGHS, US Elections,

and MLMA datasets, and for regression on the OAA dataset. Tuning mainly involved either manual configuration or automation using grid search, randomized search, and for loops, where applicable, in instances where model limitations prevented the use of explicit hyperparameter searches. Optimizing the regularizer (L1 or L2) and solver (lgfgs, liblinear, newton-cg, newton-cholesky, sag, and saga) for logistic regression was conducted to experiment with penalizing complexity by testing L1, L2, and elastic net penalties for each dataset. Loss penalty techniques reduce overfitting and improve the ability to generalize to new data. For support vector machines, the C value, loss function, maximum number of iterations, and tolerance were optimized for improved training. When implementing the RBFSampler, SGDClassifier, and SGDRegressor (for SVMs), optimizing the gamma, number of components, alpha value, maximum iterations, tolerance, and early stopping hyperparameters ensured the proper kernel coefficient and number of components for the RBF kernel in the SVM models. To prevent overfitting, a maximum C value of 1, a maximum n_components equal to the feature length, and an alpha value of 1e-4 were used. Then, loss regularization techniques were employed to enhance generalizability and yield better results. K-nearest neighbors were optimized for up to 20 clusters, with neighbors ranging from 1 to 10% of the sample size, using either uniform or distance-based weights, and various distances (Euclidean, Manhattan, Chebyshev, and Minkowski) to prevent overfitting.

Neural network modeling on the sentence embeddings necessitated tuning hyperparameters and optimizing layers for classification purposes. To process each of the 384 dimensions for the sentence embeddings, the embeddings were passed through a dense layer using 256 units, kernel regularization (rate of 1e-3), and the ReLU activation function, followed by a dropout layer using a rate of 0.1. Then, that layer's outputs were processed by a hidden

dense layer with 128 units, kernel regularization, and ReLU activation function, followed by another dropout layer with a rate of 0.15. Afterwards, that layer's outputs were processed by a hidden dense layer with 64 units and kernel regularization with the ReLU function before they were processed by another dropout layer with a rate of 0.2. Finally, this output was processed by an output dense layer with a length equal to the number of unique classes for each dataset, using the softmax activation function. This simple neural network architecture possessed sufficient depth and simplicity to process its sentence embedding inputs in a gradual manner. Hence, depending on the dataset and its ideal rebalancing method, this neural network architecture could predict the correct classification of hate speech, abuse level, and target impact with sufficient accuracy. SHAP values provided insights into the predictive ability for each class label by deconstructing relationships between each dimension and the target variable (Lundberg, 2018). Neural networks were tuned for batch sizes ranging from 8 to 64 and learning rates of 1e-2 to 1e-4. It was found that batch sizes of either 8 or 32, combined with a learning rate of 1e-2, yielded the best results due to the fast learning rate and depth of lower batch sizes. Neural networks were trained on the features alone, as well as on the features combined with UMAP (Uniform Manifold Approximation and Projection), a form of dimension reduction, the values of embeddings, and the embeddings themselves.

Technical Report: Results

The results of these experiments yielded the best model results with test accuracy, weighted F1, and macro F1 scores exceeding 70%. For some datasets, models with better per-class and macro F1 scores were favored over those with higher accuracy, as they generalized more effectively to new data and more equitably predicted each target category (Table 6).

Random forest and neural network classifiers yielded the best results, as multiple estimators allowed random forests to reduce overfitting, handle non-linear data, and be robust to outliers.

Table 6

Best model results for each dataset based on accuracy and false vs accurate predictions by label.

Dataset	Model	Balance	Accuracy	Weighted F1	Macro F1	Features
OAA	Neural Network	SMOTE	99.82%	0.9982	0.9982	Original
DGHS	Random Forest	Manual	85.24%	0.8513	0.85	Original
MLMA	Neural Network	SMOTE	82.33%	0.815	0.71	Embeddings
Convabuse	Neural Network	None	80.83%	0.8065	0.8065	Embeddings
US Elections	Random Forest	Manual	70.35%	0.7026	0.7	Original

Note. Test accuracy, weighted F1 score, and macro F1 score were evaluated.

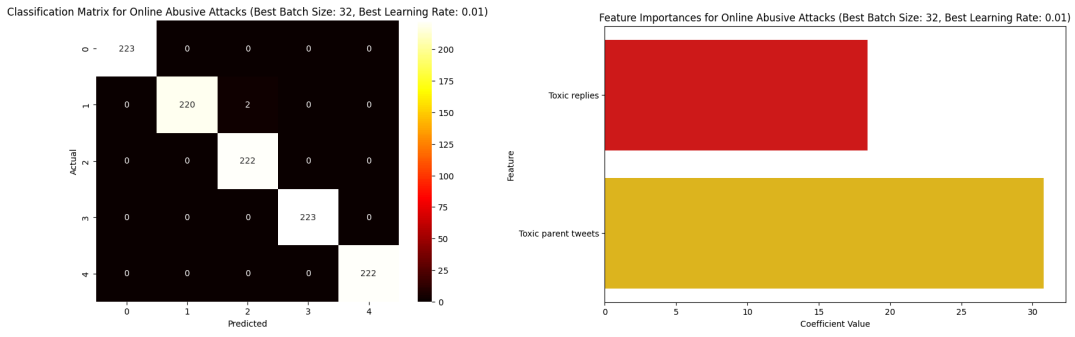
Meanwhile, neural networks identify the most significant features, nonlinear trends in data, and perform well with large datasets. Neural networks also employed deep, feed-forward architectures with basic layers (widths of 256, 128, and 64), utilized L2 regularization, and employed ReduceLRonPlateau to address accuracy plateaus during training. These architectures were ideal for both simple feature and complex embedding inputs due to the universal approximation theorem, which states that “feedforward neural networks can approximate any continuous function under certain conditions.” This is due to the expressiveness and scalability of architectures with deep, dense layers of high width (Sharma, 2025; Li et al., 2023). Effective bagging generated diverse bootstrap samples, thereby increasing the likelihood of obtaining good results with more structured data. The non-linear nature of these datasets, as well as the MLMA and Convabuse datasets, also resulted in the inability of logistic regression or linear SVMs to yield proper decision boundaries or hyperplanes on a per-class basis. Thus, these models, due to their balance of complexity and ability to handle variance and noise in these select non-linear

datasets, are effective. Manual and SMOTE upsampling yielded better per-class results compared to modeling on imbalanced data or using class weights, as they limit variance.

Using context features, neural networks demonstrated the best predictive ability in classifying toxicity levels on the OAA dataset, utilizing SMOTE-rebalancing (Figure 3), with per-class F1 scores of 1.0 for all classes. The count of toxic parent tweets yielded a feature importance of 30.79, while toxic replies had an importance of 18.45, both capturing the variance in predicting toxicity level. This model effectively processed SMOTE-rebalanced OAA data using a neural network classifier (with a batch size of 32, a learning rate of 0.01, and Adam optimization), despite the complexities. Altogether, these scores indicate that counts of toxic parent tweets and toxic replies are effective for predicting the toxicity level of text content, making them useful for gauging toxicity to evaluate tweets associated with a political campaign or for assessing chatbot outputs. However, limitations to using this model involve how toxicity for each post or reply is determined. Addressing this may involve retrofitting the model to predict whether a post is toxic (using a toxicity threshold, such as any post with toxicity above 0), and then feeding the outputs into the original toxicity classifier model mentioned.

Figure 3

Confusion matrix and feature importances for the best model for the OAA dataset.



Note. Only two entries were incorrectly predicted for the SMOTE-balanced set.

For the question of "Does a post's number of toxic parent tweets or toxic replies indicate higher levels of toxicity?", the answer, informed by these results, is that more toxic parent tweets indicate higher levels of toxicity for a post. Automating toxicity category assignments can indicate posts that are likely to be toxic on X or similar platforms.

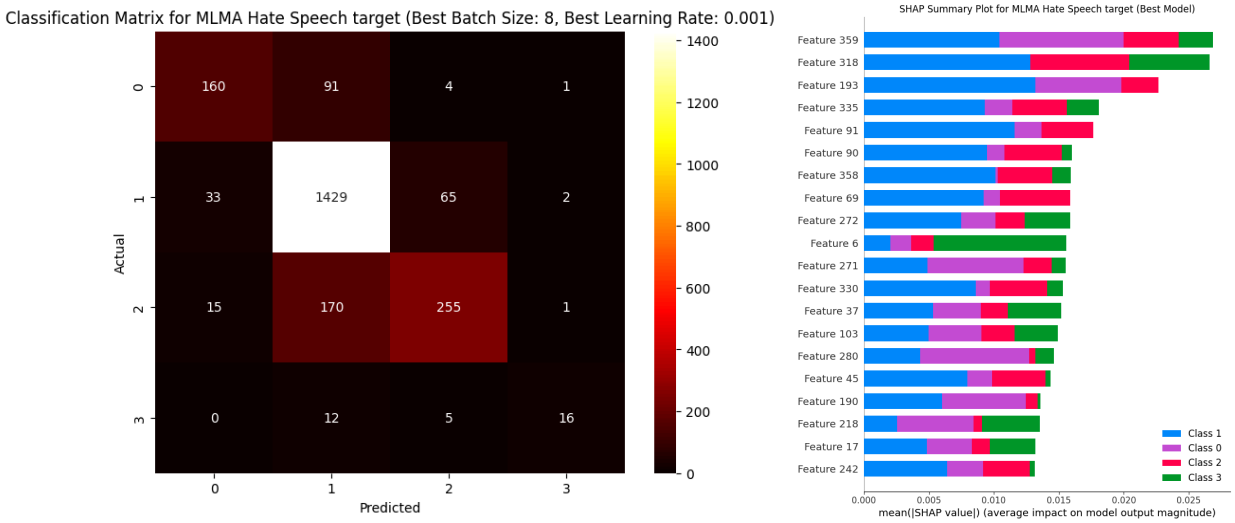
For the MLMA hate speech dataset, the neural network using SMOTE-rebalanced embeddings achieved the highest accuracy (82.33%), weighted F1 score (0.815), and macro F1 score (0.71). Figure 4 shows that top sentence embeddings had the most significant impact in predicting target group impact values of 1, which was the most common value in this dataset. Labels of 0, 2, and 3 were less common but well-represented across some of the top features, indicating a high predictive ability for the feature to predict each label. 243 out of 2,259 labels were misclassified with a lower target group impact, while 164 labels were misclassified with a higher impact. Per-class F1 scores for labels of 0, 1, 2, and 3 were 0.69, 0.88, 0.66, and 0.6, respectively. This indicates issues with the model's predictive ability for labels 0, 2, and 3. Notably, over 10% of outputs have underclassified target group impacts, as such cases underscore the real impact of content and may expose social media audiences to toxic content. Further refinement is necessary to address this limitation for this model to adequately answer the question of: "How many groups are impacted by the sentiment in this text?" for new entries.

Neural network modeling on the imbalanced Convabuse dataset embeddings yielded an accuracy of 80.83%, along with weighted and macro F1 scores of 0.8065, and per-class F1 scores of 0.89, 0.82, 0.79, 0.68, and 0.85 for abuse levels 0, 1, 2, 3, and 4, respectively. The label of 3 yielded high SHAP values, but the lowest class F1 score (0.68), indicating that the feature's predictive ability did not yield better results for this class and that less influential features yielded lesser predictive ability. The medium contributions of the top embeddings for the labels 1, 2, and

4 yielded good to high per-class F1 scores, while the label 0 had the highest F1 score of 0.89. 308 out of 5031 labels were misclassified with a lower abuse level, while 660 were misclassified with a higher abuse level. This is a concerning limitation, as it could lead to users being exposed to harmful chatbot outputs misclassified as non-abusive or less abusive. Therefore, refinement towards higher per-class F1 scores and accuracy is necessary to accurately answer the question of: "What level of abuse is associated with this text?" for new entries.

Figure 4

Confusion matrix and SHAP summary plots for random forest modeling on the MLMA dataset.



Note. Most actual labels of 3 were incorrectly predicted, while 0, 1, and 2 were more accurate.

Random forest classification was the most accurate model used for predicting hate speech, utilizing the original features of the DGHS dataset (Figure 5), with a macro F1 score of 0.85 and per-class F1 scores of 0.86 for non-hate speech and 0.84 for hate speech. These feature importances indicate that the "type" column was the strongest determinant for classifying hate speech, while target group impacts were the second-best determinant for hate speech. These are expected as sentiment strongly determines hate speech, while content involving multiple target groups may be expected to contain more negative sentiment. However, it is a concerning

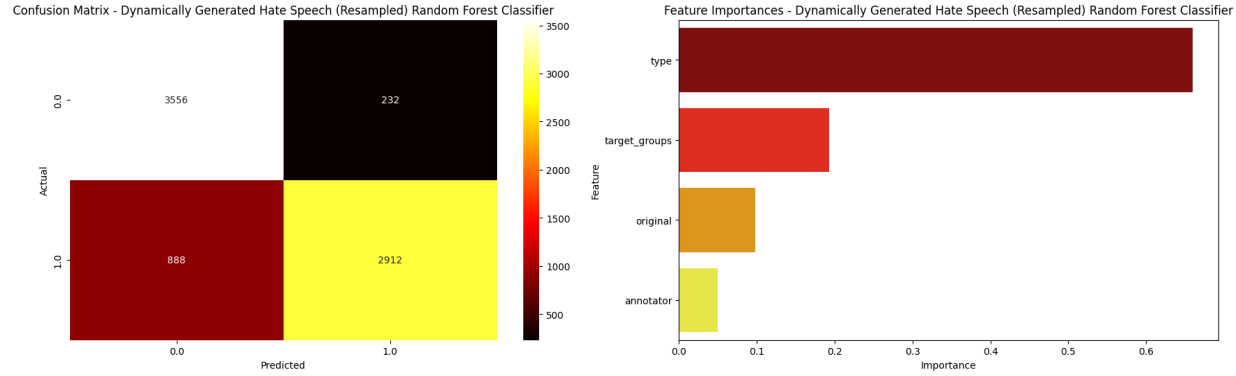
limitation that this model misclassifies 888 observations (11.67%) of hate speech as non-hate speech, as this can expose misclassified hate speech content to social media or chatbot users. The success of the random forest to predict hate speech using manually upsampled DGHS data indicates that this model reduces overfitting. To answer the questions: “What are the top contributing factors to hate speech?” and “Do characteristics, like sentiment, exacerbate hate speech?”, these results suggest that sentiment and the mention of target groups are the main contributors to hate speech. Still, further testing may be needed to address model limitations.

Another dataset with hate speech as the target variable, US Elections, yielded balanced predictions using a random forest on manually upsampled data, with an overall macro F1 score of 0.7. Most macro F1 scores for other models had values under 0.5, despite higher accuracy. This model used 795 estimators, split at a minimum of 215 samples, at least 172 samples per leaf, a maximum depth of 13, and employed the Gini index to measure impurity. Feature importances of 0.6109 associated with Biden and 0.3891 associated with Trump indicated that sentiment about Biden yielded greater predictive ability in predicting hate speech. Manually upsampled data consistently yielded the highest macro F1 scores for this dataset, due to the built-in randomization in bootstrap sampling for each estimator and the fact that manual upsampling results in the same data being used for training by multiple estimators. This combination yielded better per-class results (0.68 for non-hate speech and 0.72 for hate speech), even if accuracy was lower compared to SMOTE or imbalanced data. 131 out of 1,056 instances of hate speech were misclassified as non-hate speech, which can dilute the impact of actual hate speech and potentially harm political campaigns that use this data to inform their decisions about their own and the opposing candidate. Overall, among the best models, the random forest on the

US Election dataset is also the worst-performing dataset in terms of accuracy, weighted F1, and macro F1 scores, which warrants further model optimization to address relevant limitations.

Figure 5

Confusion matrix and feature importances for the best model for the DGHS dataset.



Note. The DGHS dataset features have the following feature importances: 0.66 for "type", 0.1927 for "target_groups", 0.978 for "original", and 0.0494 for "annotator".

Technical Report: Recommendations

Hate speech modeling and prediction have impactful applications for prominent social media companies, such as Facebook and X (formerly Twitter), political campaigns, and regulatory agencies. For these entities, context features can classify content into hate speech and non-hate speech by training a random forest classifier using features such as sentiment from the DGHS dataset or candidates like those in the US Elections dataset. Campaign staff for a candidate can replicate (and improve) the methodology for random forest on the US Elections dataset by collecting relevant sentiment data from X posts, then labeling data as hate speech or not. Fine-tuning and model optimization will be necessary due to the issues with grasping the nonlinear trends underlying the US Elections and DGHS datasets that may exist in new data.

For other hate speech predictions, counts of toxic parent posts and toxic replies (based on the OAA dataset) may be enough to determine the toxicity of content for unseen entries using

neural networks (or simpler models). Neural networks on the MLMA and Convabuse datasets also show that predictive ability must be measured in a multitude of ways to predict target group impact and abuse level, respectively. Features, such as the target group impact (Convabuse), annotator sentiment (MLMA dataset), and counts of toxic parent posts (OAA dataset), may influence data collection and exploration, enabling improvements in predictions for abuse level, target group impact, and toxicity level, respectively. However, improvements towards higher accuracy and macro F1 scores are necessary. This may involve retraining using more balanced sampling and adjusting the neural network architecture to fit the new data better. Other limitations center on the bias behind determining qualities, such as sentiment (which can be addressed by standardizing labeling procedures) and using different sentiment columns (instead of a total sentiment value) to determine sentiment for a given observation as inputs for neural networks. Additionally, for data similar to the OAA dataset, a pipeline using linear regression and feature regression is necessary to calculate and segment toxicity by extremity.

Still, as it stands, the most deployment-ready model from this analysis is the neural network for predicting toxicity on the OAA dataset. More refinement is needed before incorporating the neural networks for the Convabuse and MLMA data, as well as the random forest models for the DGHS and US Elections data. Additionally, context features should be utilized when available or obtainable. Existing sentiment analysis models can also be leveraged to automate labeling content by toxicity in context or via embeddings. If implemented, these recommendations will enhance content moderation algorithms to safeguard users of social media and AI platforms from harm, while improving detection accuracy. This concludes the recommendations section, based on insights gained from exploration, modeling, and results.

Non-Technical Report on Analyzing Hate Speech Context and Content

Non-Technical Report: Executive summary

Hate speech promotes political polarization through its adverse impacts on online discourse, deeming it necessary to quantify and classify hate speech and identify the key determinants (features) of hate speech and toxic online content. For this modeling, the goal was to predict hate speech, abuse level, and text toxicity using metadata features and text embeddings (which are numerical representations of text data) from several datasets containing social media or conversational data with features related to hate speech and sentiment. Here, classifier models were used to predict hate speech, the level of abuse, the impact on the target group, and the toxicity level of the content. Classification was performed by subjecting the original features (after feature selection) or sentence embeddings (numerical representations of the data) to scaling and training on the training set, followed by testing on the testing set. The training set was used to train models to predict toxic level, hate speech, abuse level, or target groups.

In contrast, the testing set was used to evaluate models by obtaining final test scores for accuracy, weighted F1 score, and macro F1 score. These scores were then used to determine the best model for each dataset. They yielded results of 70 (hate speech) to 99.8% (toxic class) accuracy, 70 (hate speech) to 99.8% (toxic class) accuracy for each classification (see Table 7 in the “Non-Technical Report: Results” section).

Measuring sentiment can enable platforms to determine whether content constitutes hate speech. Other metadata signals, like the number of parent tweets of toxic tweets (as in the Online Abusive Attacks dataset), can be used to predict post toxicity. Regarding broader impacts, entities can utilize or collect data, like type (sentiment) from the Dynamically Generated Hate Speech or candidate sentiment from the Hate Speech 2020 U.S. Elections dataset, to train chatbots or models for hate speech detection. Collecting data similar to the Convabuse or Online

Abusive Attacks dataset can also help determine whether content is toxic, identifying a candidate's association with hate speech, or train a model's ability to detect toxicity. If enacted, these measures could keep online users safer by preventing their exposure to toxic speech.

Non-Technical Report: Introduction and problem definitions

With the rise of social media and generative artificial intelligence (also known as Gen AI), algorithmic political bias has become a topic of high interest. What is algorithmic political bias, and what does it have to do with hate speech? It occurs when AI systems promote filtering based on political orientation. Hate speech may be encouraged in what are called "social media bubbles", which are generated by algorithmic political bias. Mitigating these bubbles will require a multifaceted effort, including the classification and assessment of hate speech severity. Other use cases highlight the relevance of current events, such as elections, that warrant sentiment analysis of social media data to quantify factors like relatability and sentiment. Predicting toxicity levels in text can also be helpful in social media moderation tasks and evaluating the outputs of large language models for toxic or otherwise hateful content.

For classifying hate speech, abuse levels, and the number of target groups impacted, the most relevant measures are accuracy (the success rate of classification) and F1 score (success rate by category), which could be weighted (rebalanced figures based on frequency), macro (no rebalancing), and per-class (specifically for each label). Feature importances measure the significance of features to determine the most relevant factors contributing to hate speech. SHAP (SHapley Additive exPlanations) explained how each feature contributed to the results, making it useful for training sets with many features. These are relevant for the Convabuse, Dynamically Generated Hate Speech (DGHS), Hate Speech US 2020 Elections (US Elections), and Multilingual and Multi-Aspect (MLMA) Hate Speech datasets to provide insights determining:

"What are the top contributing factors to hate speech?" and "Do characteristics, like sentiment, exacerbate hate speech?" (Yadav, 2024) Accuracy, F1 scores, and feature importances determined the significance of features in predicting toxicity, allowing evaluation of questions such as "Does a post's number of toxic parent tweets or toxic replies indicate higher levels of toxicity?" Finally, these metrics ensure that the best models have sufficient predictive power to generalize new data into each relevant category without significant prediction imbalance.

Non-Technical Report: Data Overview

The data used in this analysis consisted of labeled classification datasets collected from outputs of chatbots (Convabuse), gathering social media posts about the 2020 U.S. Election (US Elections), specific keywords from social media (MLMA), using synthetic outputs to train content classification algorithms (DGHS), and identifying targets and keywords to harvest data (OAA). For the Convabuse dataset, Cercas Curry et al. collected and combined data intended to train chatbots. Modeling on the Convabuse dataset utilized current and previous agent and user text data to predict one of five abuse levels, ranging from the least severe to the most severe (2021), with a total of 12,768 rows and 10 columns (8 context, combined into 1 for text, and one target). Then, the DGHS dataset and the US Elections dataset were used to train the model to classify observations as hate speech. Vidgen et al. (2021) trained annotators to generate and label dynamically generated text data for the DGHS dataset, which included columns for hate speech labels, abuse level, targeted groups, and originality. The dataset comprised 37,938 observations and six columns: four context columns, one text column, and one target column. Grimminger and Klinger gathered social media data from the 2020 U.S. Presidential Election to analyze sentiment toward Biden, Trump, and hate speech (2021), utilizing 2,313 observations (in context). Modeling on the Online Abusive Attacks (OAA) dataset predicted toxicity using posts

on Twitter (now X). Alharthi et al. (2023) included the number of toxic replies and toxic parent posts, while training involved applying toxicity levels to its 2,313 context entries.

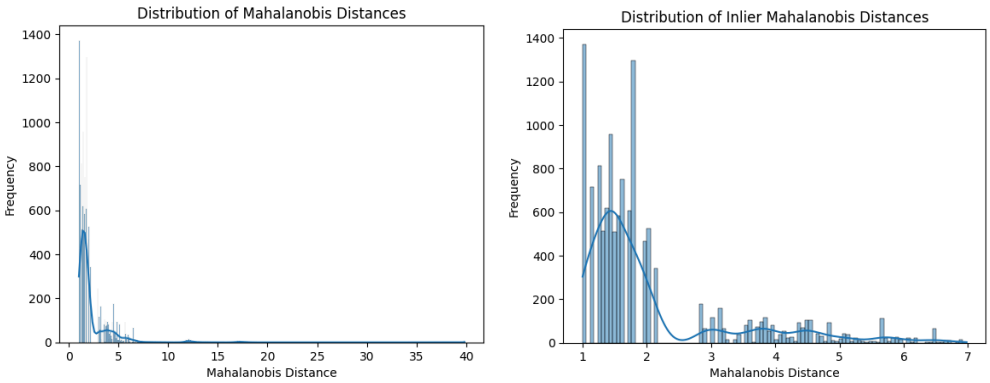
Meanwhile, modeling on the MLMA dataset predicted the number of target groups, ranging from 0 to 3. For the MLMA dataset, Ousidhoum et al. (2019) included columns for sentiment, tweet, target groups, implicit, explicit, and abuse level, with 18,126 observations and six columns (four context, one text, and one target). Most inconsistencies were due to columns with too many labels, which were addressed through feature engineering. Only eight entries had null values (all from the MLMA dataset). Ethical considerations involved in these datasets included user identification information associated with social media accounts, which was most problematic in the OAA dataset. This issue was addressed by removing ID, index, and other potentially sensitive details from each dataset. It was also troubling that the Convabuse dataset was more accurate for lower abuse levels, and that the DGHS and US Elections models performed worse on actual hate speech, as misidentified actual hate speech is more harmful than false positives misidentified as actual non-hate speech. Finally, nearly every target variable was imbalanced to various degrees, which is why rebalanced datasets would perform better than imbalanced datasets or imbalanced data used to train models using simple reweighting.

Non-Technical Report: Data cleaning, preprocessing, and exploratory data analysis

Data cleaning, preprocessing, and exploratory data analysis for each dataset resulted in extreme restructuring. This process began by removing missing values, extreme outliers, and inconsistent duplicates (e.g., duplicate entries) through methods such as calculating the degree to which an entry is considered an outlier in the target variable (as shown in Figure 6).

Figure 6

Mahalanobis distribution of the Convabuse dataset based on a threshold of 7 for outlier removal.



Note. This illustrates how observations of the "abuse_level" column were assigned a Mahalanobis distance (using the distance between each point and the median "abuse_level") and how outliers were identified as points over seven standard deviations from the mean.

For context modeling, ridge regression reduced and standardized the impact of loss, thereby reducing the necessary features to two essential columns for the OAA dataset. Expanding the DGHS dataset's "target" column and the MLMA dataset's "type" column, and summing their results into "target_groups," measured the impact of each entry's target group. Similarly, abuse levels were applied using relevant columns in the Convabuse and MLMA datasets. For sentence embedding models, SentenceBERT was used to calculate embedding vectors (a list of numerical representations for each sentence) for text in the Convabuse, DGHS, and MLMA datasets, with 384 dimensions (numbers) per vector. SHAP also provides a degree of dimensionality reduction in interpreting the influence of each dimension in predicting the target label, while importances determine the most impactful features. A total of five sets of context features and three sets of embeddings were used for modeling. Meanwhile, model choices involved both linear logistic regression and support vector machines, as well as non-linear tree-based, ensemble, distance, and neural networks to gauge the underlying linear or non-linear trends in the data. Given that neural networks and random forests were the best models (see Table 7 in the "Non-Technical Report: Results" section), all datasets exhibited non-linear trends,

as evidenced by the findings of imbalanced target variables, particularly in abuse levels, and skewed distributions for notable features, such as a preference towards specific annotators.

Non-Technical Report: Modeling/Analysis

Classification modeling involved using linear models (such as linear regression, logistic regression, and linear support vector machines), decision trees, ensemble models (e.g., random forest), and neural network models. Linear models captured linear relationships in various datasets, such as the Convabuse and US Elections datasets. Decision trees analyzed potential outcomes to aid decision-making, facilitating classification for multiple datasets. Both were used to evaluate the quality of outlier removal. Ensemble models, which utilize the logic of decision trees, train ensembles of estimators and mitigate overfitting. Simple neural network models classified sentence embeddings to predict a target variable by processing original features and embeddings through different layers into different target classifications.

Evaluation involved accuracy, weighted F1 score, macro F1 score, and per-class F1 score. Accuracy was the primary metric, though weighted and macro F1 scores were used to determine the best model. The 80-20 train-test split (where the data is split into features and targets, then into 80% for training and 20% for testing) and stratifying by the y train set (maintaining balance throughout each subset) were randomly applied to all datasets. Then, each model was trained using data that was imbalanced or rebalanced using different methods to address class imbalances in the target variable through either class weights (lower frequency class labels are weighted more heavily than higher frequency class labels) or oversampling minority class labels (lower frequency class labels are resampled to the number of entries associated with the majority label). Then, cross-validation randomly divided the data into subsets (“folds”), applied to each dataset (five folds were the most successful) before modeling, where training occurs once per

fold. Finally, additional model optimizations were crucial to improve performance through fine-tuning, where optimizations balanced model performance but were limited to specific ranges (e.g., up to 100 estimators) to avoid overfitting, which is when a model overtrains on its training data but cannot generalize well to new data.

Non-Technical Report: Results

These experiments utilized different datasets and balancing techniques, yielding the best model results with test accuracies, weighted F1, and macro F1 scores exceeding 70% (Table 1).

Table 7

Best model results for each dataset based on accuracy and false vs accurate predictions by label.

Dataset	Model	Balance	Accuracy	Weighted F1	Macro F1	Features
OAA	Neural Network	SMOTE	99.82%	0.9982	0.9982	Original
DGHS	Random Forest	Manual	85.24%	0.8513	0.85	Original
MLMA	Neural Network	SMOTE	82.33%	0.815	0.71	Embeddings
Convabuse	Neural Network	None	80.83%	0.8065	0.8065	Embeddings
US Elections	Random Forest	Manual	70.35%	0.7026	0.7	Original

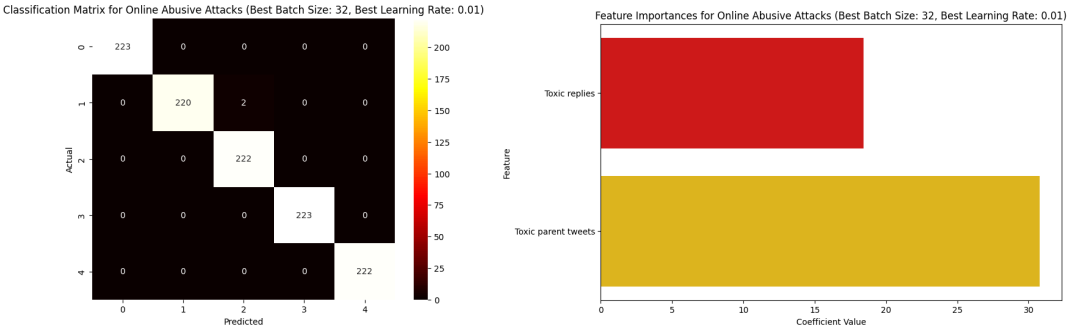
Note. Test accuracy, weighted F1 score, and macro F1 score were used for evaluation.

Random forest and neural network classifiers yielded the best results. Random forests' multiple estimators reduced overfitting, handled non-linear data, and were robust to outliers. Neural networks identified the best features, nonlinear trends in the data, and performed well with large datasets. Neural networks also utilized deep, feed-forward architectures (where data is sent from one layer to the next), which reduced the impact of loss and lowered the learning rate during periods of accuracy plateau. Upsampling yielded the best macro and per-class F1 scores, as SMOTE (Synthetic Minority Oversampling Technique) placed new data points between existing data points. In contrast, manual upsampling included more data in place.

Using context features, neural networks demonstrated the best predictive ability for classifying toxicity levels on the OAA dataset, with SMOTE-rebalancing, predicting the correct toxicity level nearly all the time (see Figure 6) and achieving per-class F1 scores of 1.0 for all classes. This model, trained on SMOTE-rebalanced embeddings for the MLMA data, achieved the best combination of accuracy, weighted F1 score, and macro F1 score. Neural network modeling on the imbalanced Convabuse embeddings yielded high accuracy, weighted F1 and macro F1 scores, and per-class F1 scores of 0.68 to 0.89 (which are good). Toxic parent tweets had a larger importance compared to toxic replies; both were effective in determining toxicity levels using the OAA data. Figure 7 shows that the top embeddings had the most significant impact in predicting an impact of 1, the most common value in the MLMA data. 0, 2, and 3 were less common but had per-class F1 scores at or above 0.66. Out of 2,259 labels, 243 were misclassified, with a lower impact on the MLMA dataset. This is problematic, as speech targeting more groups may be harmful, so social media users may be exposed to content misclassified as less dangerous. 308 out of 5031 labels were misclassified with a lower abuse level, which is also potentially damaging to social media and chatbot users who may be exposed to more abusive content misclassified as being less abusive. Finally, using neural networks on data similar to the OAA dataset involves determining toxicity for each post or reply. Addressing this may involve retrofitting the model to predict whether a post is toxic (using a toxicity threshold, such as any post with toxicity above 0) and then feeding the outputs back into the original model.

Figure 6

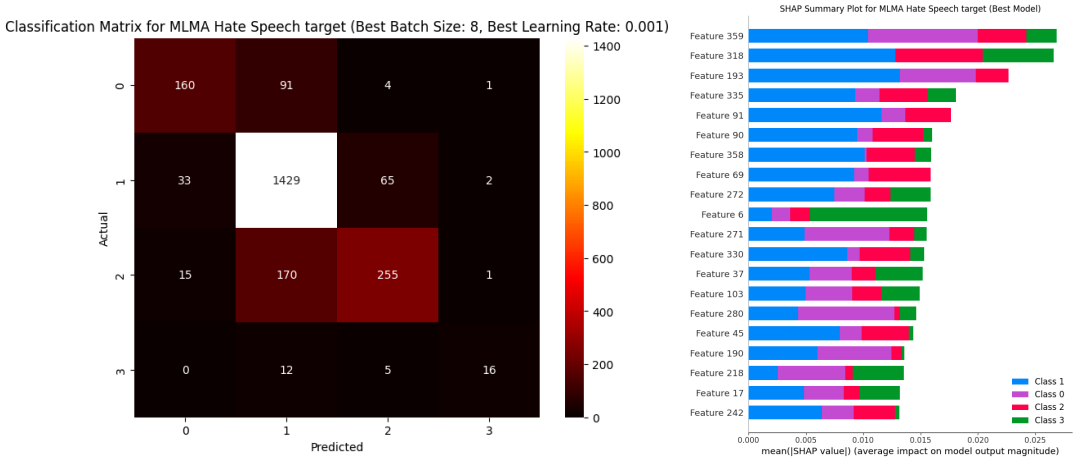
Confusion matrix and feature importances for the best model for the OAA dataset.



Note. Only two entries were incorrectly predicted for the SMOTE-balanced set.

Figure 7

Confusion matrix and SHAP summary plots for random forest modeling on the MLMA dataset.



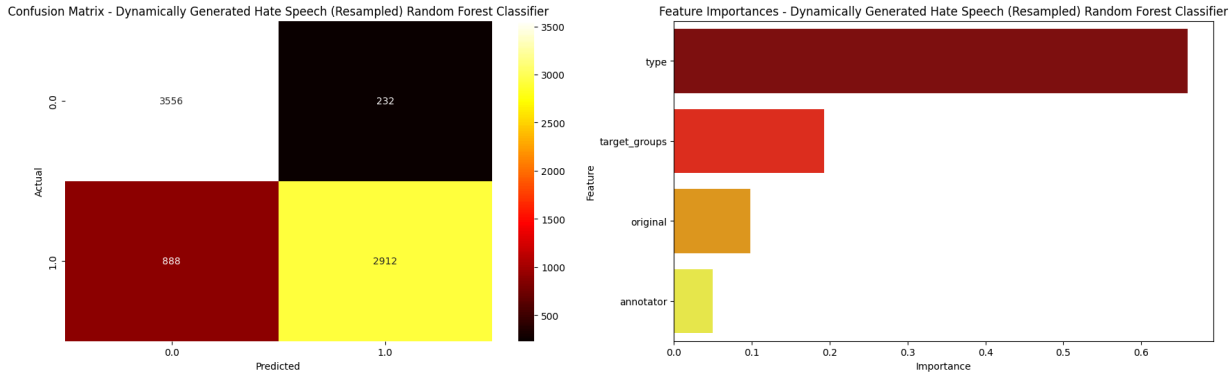
Note. Most actual labels of 3 were incorrectly predicted, while 0, 1, and 2 were more accurate.

Random forest classification was the most accurate model used for predicting hate speech, utilizing the original features of the DGHS dataset (see Figure 8), with high macro F1 and per-class F1 scores exceeding 0.8, and on the US Elections dataset, with balanced predictions and per-class F1 scores around 0.68 for non-hate speech and 0.72 for hate speech (medium to good scores). The feature importances of the “type” and “target_groups” columns indicate them as the strongest determinants for classifying hate speech in the DGHS dataset. Higher feature importance for Biden also showed that sentiment about Biden yielded better

predictive ability for determining hate speech, as confirmed by Grimminger & Klinger (2021). This model misclassifies 888 observations of actual hate speech as non-hate speech in the DGHS dataset, potentially exposing social media and chatbot users to misclassified hate speech. 131 out of 1,056 instances of hate speech were misclassified as non-hate speech in the US Elections dataset, which could underscore the impact of actual hate speech and result in harm to campaigns seeking to use this model to analyze content related to campaigns.

Figure 8

Confusion matrix and feature importances for the best model for the DGHS dataset.



Note. The DGHS dataset's "type" and "target_groups" were the most essential features.

Non-Technical Report: Recommendations

Hate speech modeling and prediction have impactful applications for prominent social media companies, such as Facebook and X (formerly Twitter), political campaigns, and regulatory agencies. For social media and chatbot entities, context features can classify content into hate speech and non-hate speech by training a random forest classifier using features such as sentiment from the DGHS dataset or candidates like those in the US Elections dataset. Campaign staff for a national political position can replicate (and improve) the methodology for random forest on the US Elections dataset by collecting sentiment data from X posts about them and the opposing candidate, then labeling this data as hate speech or not. However, hyperparameter

tuning and model experimentation will be necessary due to the issues with grasping the nonlinear trends that may also exist in new data.

For other hate speech predictions, counts of toxic parent posts and toxic replies (based on the OAA dataset) may be enough to determine the toxicity of content for unseen entries using neural networks (or simpler models). Neural networks on the MLMA and Convabuse datasets also show that predictive ability must be measured in a multitude of ways to predict target group impact and abuse level, respectively. Features, such as the target group impact (Convabuse), annotator sentiment (MLMA dataset), and counts of toxic parent posts (OAA dataset), may influence data collection and exploration, enabling improvements in predictions for abuse level, target group impact, and toxicity level, respectively. Improvements for higher predictive capacity involve more balanced sampling (such as sampling based on different groups) and re-adjusting the neural network architecture. Other limitations for these strategies center on the bias inherent in determining qualities, such as sentiment (which can be addressed by standardizing labeling procedures) and using different sentiment columns (instead of a total sentiment value) to determine sentiment for a given observation as inputs for neural networks.

Insights from context modeling can guide organizations in implementing models and identifying datasets containing essential features. This scheme would involve different feature sets in modeling based on the business context. A large AI firm may want to predict sentiment, using columns for the chatbot and group vs individual targets, then ensuring proper annotation for abuse level and the number of target groups, summing weighted abuse levels to determine sentiment. Sentence embeddings, obtained by applying SentenceBERT to text content, would be fed into a neural network to predict sentiment on new data. Meanwhile, sentiment and other columns would be fed into simpler classification models, such as logistic regression on

imbalanced data (akin to the Convabuse dataset), to classify hate speech. Similarly, a campaign could use a sentiment analyzer on labeled data to determine the sentiment associated with their candidate and an opposing candidate, gauging whether a piece of text constitutes hate speech. From this, campaign staff can use feature importances to determine which candidate is more closely associated with hate speech, using these insights in campaign ads and messaging.

A pipeline, such as the one used to model toxicity levels on the OAA dataset, can extrapolate insights from metadata, including the count of toxic parent posts and replies, and a sentiment analyzer model to predict toxicity within a numerical range. The distribution of toxicity levels can guide categorical encoding of toxicity into different classes (e.g., five categories from 0 to 4) or in labeling hate speech. In the latter case, the resultant dataset could be fed into a hate speech random forest classifier, where the determination would be manually made by annotators or calculated based on toxicity. Other metadata, such as that for opposing candidates, could provide further context for modeling.

Still, as it stands, the most deployment-ready model from this analysis is the neural network to predict toxicity on the OAA dataset. More refinement is needed before incorporating the neural networks for data like Convabuse and MLMA data, as well as the random forest models for the DGHS and US Elections data. Additionally, context features should be utilized when available or obtainable. Existing sentiment analysis models can also be leveraged to automate labeling content by toxicity in context or through embeddings. If implemented, these recommendations will enhance content moderation and sentiment analysis algorithms to safeguard users of social media platforms and AI chatbots from harm, while also improving detection capabilities. This section outlines the recommendations section, based on insights gained from exploration, modeling, and results.

References

Ahmed, S., Kamal, K., Mathavan, S., Hussain, G., Alkahtani, M., & Alkahtani, M. (2022). Aerodynamic Analyses of Airfoils Using Machine Learning as an Alternative to RANS Simulation. Applied Sciences, 12(10), 5194.

Alharthi, R., Alharthi, R., Shekhar, R., & Zubiaga, A. (2023). *Target-oriented investigation of online abusive attacks: A dataset and analysis.* IEEE Access, 11, 64114–64127. https://doi.org/10.1109/access.2023.3289148

Cercas Curry, A. (n.d.). Amandacurry/Convabuse. GitHub. https://github.com/amandacurry/convabuse

Ceras Curry, A., Abercrombie, G., & Rieser, V. (2021). *ConvAbuse: Data, analysis, and benchmarks for nuanced detection in conversational AI. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.* https://doi.org/10.18653/v1/2021.emnlp-main.587

Chitra, U., & Musco, C. (2020). Analyzing the impact of filter bubbles on social network polarization. Proceedings of the 13th International Conference on Web Search and Data Mining, 115–123. https://doi.org/10.1145/3336191.3371825

Capstone project guidance. ChatGPT. (2025, December 1). https://chatgpt.com/share/692cf1c1-eb40-8000-8cc4-555dbb9a882e

Grimminger, L., & Klinger, R. (2021, April). *Hate towards the political opponent: A twitter corpus study of the 2020 US elections on the basis of offensive speech and stance detection.* ACL Anthology. https://aclanthology.org/2021.wassa-1.18/

Klinger, R. (n.d.). Hate speech / offensive speech in the US 2020 elections. Hate Speech / Offensive Speech in the US 2020 Elections | Institut für Maschinelle Sprachverarbeitung |

Universität Stuttgart. https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/stance-hof/

Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W.-C. (2023). Effective entity matching with Transformers. The VLDB Journal, 32(6), 1215–1235. https://doi.org/10.1007/s00778-023-00779-z

Lundberg, S. (2018). Violin summary plot. violin summary plot - SHAP latest documentation. (n.d.). https://shap.readthedocs.io/en/latest/example_notebooks/api_examples/plots/violin.html

Ousidhoum, N. (2024, November 26). HKUST-KnowComp/MLMA_hate_speech: Dataset and code of OUR EMNLP 2019 paper "multilingual and multi-aspect hate speech analysis." GitHub. https://github.com/HKUST-KnowComp/MLMA_hate_speech

Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., & Yeung, D.-Y. (2019). *Multilingual and multi-aspect hate speech analysis.* Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). https://doi.org/10.18653/v1/d19-1474

Sharma, S. (2025, July 23). Universal approximation theorem for Neural Networks. GeeksforGeeks. https://www.geeksforgeeks.org/deep-learning/universal-approximation-theorem-for-neural-networks/

Shmulewitz, D., Levitin, M. D., Skvirsky, V., Vider, M., Lev-Ran, S., & Mikulincer, M. (2025). Exposure to online hate speech is positively associated with post-traumatic stress disorder symptom severity. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-16168-1

Vidgen, B., Thrush, T., Waseem, Z., Kiela, D. (2020). *Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection.* https://doi.org/10.48550/arXiv.2012.15761

Yadav, A. (2024, September). SHAP Values vs Feature Importance. Medium. https://medium.com/