CE807-24-SP – Assignment Report

STUDENT ID: 2310618 email: nv23693@essex.ac.uk

Abstract

Toxic comment classification holds significant importance in various online platforms and communities. Identifying and moderating toxic comments is crucial for maintaining a healthy and safe environment for users to engage and interact. This project explores effective methods for toxicity detection in text data, using a dataset annotated with comments, their toxicity status, and designated dataset splits. The applied Machine Learning analytical framework combines statistical text analysis, specifically techniques to analyze and learn from natural language data. The choice of methodology was informed by a thorough review of current best practices in text analytics, focusing on optimizing accuracy and recall in a balanced manner. The experimental setup, tested across train, test, and validation splits, provided good evidence supporting the analytical approach. Results indicated a good degree of efficiency in identifying toxic comments, showcasing the potential of the methods in real-world applications. This work contributes significantly to the ongoing efforts in automating content moderation and enhancing online community management.

Materials

You must provide the clickable link to your Zoom Recorded presentation. For example, this is my link: LaTeX-Tutorial. Make sure that the link is clickable. Now, it has dummy links.

- Video Presentation (Mandatory)
- Google Colab (Mandatory)
- Google Drive (Optional)
- Any other links (Optional)

1 Task 1: Model Selection

1.1 Summary of 2 selected Models

- **Model-1 Summary - Gaussian Naive Bayes**

  Gaussian Naive Bayes (GNB) [6] is a probabilistic-based classifier used for text classification. It assumes that features follow a normal distribution and are independent of each other. GNB calculates the probability of a given sample belonging to each class based on Bayes' theorem and selects the class with the highest probability as the prediction. In text classification, GNB calculates the likelihood of each word occurring in toxic and non-toxic comments and combines these probabilities using Bayes' theorem to determine the overall probability of the comment being toxic or non-toxic. Despite its simplistic assumptions, GNB performs well in practice, particularly with continuous data.

- **Model-2 Summary- Gradient Boosting Classifier**

  Gradient Boosting Classifier (GBC) [4] is a powerful machine learning algorithm for text classification. It progressively builds an ensemble of decision trees, each focusing on correcting the errors of its predecessor. GBC calculates the probability of a given comment belonging to each class and combines these probabilities using a weighted sum. During training, it minimizes the loss function using gradient descent. In text classification, GBC learns to classify comments by identifying the most informative features to distinguish between toxic and non-toxic comments, resulting in a highly accurate and robust model. Despite its complexity, GBC is efficient and provides exceptional performance in handling the difficulty of text data.

1.2 Critical discussion and justification of model selection

Among the models implemented [2], only the Gradient Boosting Classifier (GBC) and Gaussian Naive Bayes (GNB) demonstrated good performance. The limited dataset size might have favored simpler models.

In terms of generative ability, Gaussian Naive Bayes (GNB) is known for its simplicity and assumes that features are independent and follow a normal distribution. Despite its simplistic assumptions, GNB performed well in predicting the out label model in the dataset. This indicates that even with limited data and without complex feature interactions, GNB was able to capture important patterns in the text data and make accurate predictions. Additionally, GNB's computational efficiency makes it suitable for processing large volumes of text data, further justifying its use in toxicity detection tasks.

In terms of discriminative ability, models such as Support Vector Classifier (SVC) and Long Short-Term Memory (LSTM) were also explored. However, they did not perform well in identifying the out label model. In contrast, GBC exhibited the capability to predict the out label model with good accuracy. The strength of GBC lies in its robustness and adaptability to complex datasets. The decision to use GBC was justified based on its superior performance compared to other models and its ability to effectively identify the out label model.

The decision to use both GBC and GNB was justified based on their respective strengths and superior performance compared to other models. While GBC excelled in handling complex datasets and capturing non-linear relationships between features, GNB demonstrated effectiveness even with simplistic assumptions, making it a strong candidate for toxicity detection tasks with limited data..

2 Task 2: Design and Implementation of Classifiers

- **Data Set Details:**

The dataset consists of three CSV files: Train, Test, and Valid. Each row contains the following columns:

Comment ID	Comment	Split	Toxicity
1	"This is a toxic comment."	Train	1
2	"This is a non-toxic comment."	Test	0
3	"Another toxic comment here."	Valid	1

Table 1: Dataset Details

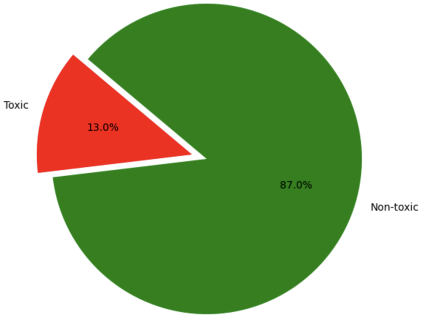

Figure 1: Distribution of Toxic Comments in the Dataset

1. **Comment ID**: A unique identifier for each comment.
2. **Comment**: The text of the comment.
3. **Split**: Indicates whether the comment belongs to the training, testing, or validation set.
4. **Toxicity**: Specifies whether the comment is toxic (1) or not (0).

The primary objective is to classify whether a text is toxic or not. With this goal, the dataset is divided into training, testing, and validation sets. The 'Comment' column contains the text data, while the 'Toxicity' column indicates whether the comment is toxic (1) or not (0) Figure 1 shows the distribution of toxic comment in the dataset and table 1 indicates the dataset details. This dataset can be used to train machine learning models to accurately identify toxic comments. By using this data, text classification models can be developed to automatically identify toxic language, making online platforms safer and more conducive to healthy communication.

Dataset Preprocessing: [5]

1. **Performing Lowercase:** The initial step in preprocessing involved lowercasing the text in the comment column to ensure consistency in the text data.
2. **Removal of Special Characters:** Special characters were removed from the comments to maintain uniformity in the text data. This step helps in standardizing the text, making it ready for further processing.
3. **Removal of Stop Words:** Stop words, such as 'and', 'the', 'is', etc., were removed from the comments. Stop words are commonly used words that do not carry much meaning and can be excluded to focus on the more meaningful words.

4. **Tokenizing:** The comments were tokenized, splitting them into individual words or tokens. Tokenization is a fundamental step in natural language processing, allowing algorithms to work with individual words.

5. **Lemmatization:** Lemmatization was applied to further normalize the text data by reducing each word to its base or dictionary form. Unlike stemming, lemmatization considers the context and meaning of the word, resulting in better-quality data for analysis.

6. **Stemming:** Stemming was performed to reduce words to their root form, which helps in normalization and improving the computational efficiency of the algorithm. It ensures that different forms of the same word are treated as identical.

Implementation:

In the project, two distinct models were employed to handle the task of identifying toxic comments: a Gaussian Naive Bayes model for generative modeling and a Gradient Boosting Classifier for discriminative modeling.

Generative Modeling with Gaussian Naive Bayes: For the generative modeling aspect using Gaussian Naive Bayes, the process began with data preprocessing, ensuring uniformity and readiness for analysis. After cleaning the training and validation datasets, TF-IDF (Term Frequency-Inverse Document Frequency) was applied to transform the text data into numerical features, capturing the importance of each word in the comments relative to the entire dataset. These features were then fed into the Gaussian Naive Bayes classifier for training. The model learned to predict comment toxicity based on these TF-IDF features. Post-training, the model's performance was evaluated on the validation set, and metrics were calculated to show its effectiveness.

Discriminative Modeling with Gradient Boosting Classifier: On the other hand, the discriminative modeling approach utilized a Gradient Boosting Classifier. Similar to the generative model, the training and validation datasets underwent preprocessing to ensure consistency. TF-IDF was again employed to convert the text data into numerical features. These features were then used to train the Gradient Boosting Classifier through a pipeline approach. The classifier learned to predict comment toxicity using these TF-IDF features. After training, the model's performance was evaluated on the validation set.

Dataset	Total	% Class 0	% Class 1
Train	8699	86.987	12.99
Valid	2920	86.78	13.15
Test Generative	2896	67.87	32.09
Test Discriminative	2896	95.95	3.86

Table 2: Dataset Details

Model	F1 Score
Gaussian Naive Bayes Model	0.186
Gradient Boosting Machine Model	0.085
SoTA [1]	0.94

Table 3: Model Performance.

3 Task 3: Analysis and Discussion

Critical Discussion of Performance Comparing Both Models [3]

SoTA Model Comparison: The State-of-the-Art (SoTA) Bidirectional LSTM model outperforms both models in terms of an F1-score of 0.94. Compared to the GNB and GBM models, Table 3 shows that the SoTA model performs better across all metrics. This is due to the SoTA[1] model undergoing extensive hyperparameter tuning to optimize its performance. Parameters such as the number of layers, choice of optimizer, activation function, loss function, learning rate, batch size, and number of epochs were carefully selected to enhance model effectiveness. Specifically, the Adam optimizer was utilized for weight updates due to its efficiency and reduced memory usage. Additionally, a batch size of 32, binary cross-entropy loss function, and sigmoid activation function were chosen to further refine model performance. The superiority of the SoTA model can also be attributed to the characteristics of the dataset used for training. The dataset likely provided a more detailed representation of toxic comments, enabling the model to learn patterns effectively.

The following evaluation metrics carried out to analyse the models performance of GNB and GBC

1. **Accuracy Metric:** The discriminative model significantly outperforms the generative model in terms of accuracy. The discriminative model has an accuracy of 0.844, while the generative model has an accuracy of 0.595.

   Although the discriminative model achieves a higher overall accuracy compared to the generative model, it's important to note that the Gaussian Naive Bayes model (generative) resulted a relatively high ability to predict toxic comments. Table 2 shows that accounting for 32.09 percentage of the test dataset. This indicates that despite its lower accuracy, the generative model captures a considerable portion of the toxic comments in the dataset, showcasing its effectiveness in identifying such instances.

2. **Recall Metric:** The generative model exhibits a higher recall (0.351) compared to the discriminative model (0.055), higher recall for the generative model comes at the cost of potential false positives. By prioritizing the identification of toxic comments, the generative model may also incorrectly classify some non-toxic comments as toxic, leading to a higher false positive rate. The discriminative model, on the other hand, has a lower recall, indicating that it misses a larger proportion of the actual toxic comments. However, its strength lies in its lower false positive rate, as evidenced by its slightly higher precision (0.186) compared to the generative model's precision (0.126).

3. **Precision Metric:** The precision of both models is low, but the discriminative model (0.186) performs slightly better compared to the generative model (0.126). This indicates that the discriminative model produces fewer false positives.

4. **F1 Score Metric:** The F1 score of the discriminative model (0.084) is slightly lower than that of the generative model (0.185), suggesting that the generative model has a slightly better balance between precision and recall.

5. **Confusion Matrix Metric:** In the confusion matrix, the discriminative model has significantly fewer false positives compared to the generative model, which is reflected in its higher precision. The generative model may capture more instances of toxic comments, with a higher rate of false positives.

6. **Summary of Analysis:** While the discriminative model results a good accuracy and precision, it's important to recognize that model selection should be adopted to the specific requirements of the task. In scenarios where the priority is to minimize false negatives (i.e., accurately identifying toxic comments), the generative model's higher recall may offer advantages, despite its lower overall accuracy. On the other hand, if precision and minimizing false positives are considered, the discriminative model may be preferred.

provide interesting examples in a Table 4 for both models and explain the model's output.

Comment ID	GT	Generative	Discriminative
257542852	1	predicted	predicted
78309297	1	predicted	predicted
253911271	1	predicted	predicted
419746843	1	Not predicted	Predicted
635744705	1	predicted	Not predicted

Table 4: Comparing two Models with diverse examples. Don't forget to submit your text via `https://forms.office.com/e/XmPgSVsZia`

3.1 Justification of Model's performance

The dataset used for training and evaluation seems to be imbalanced, with a significantly smaller proportion of toxic comments compared to non-toxic comments. This class imbalance challenges for both models, as they may struggle to effectively learn the patterns associated with the minority of toxic comments. Imbalanced datasets often result in biased models that favor the majority class, leading to lower performance metrics for the minority class.

While the discriminative model demonstrates relatively good accuracy and precision compared to the generative model, both models still struggles with low recall scores, indicating their difficulty in effectively identifying toxic comments. Further enhancements are necessary to improve their performance, particularly in terms of recall, to ensure their suitability for practical use. Additionally, exploring ensemble methods or using advanced natural language processing techniques could potentially address these limitations and enhance the models' overall performance.

3.2 Example and other Analysis

Based on the Table 4 the example and analysis carried below

1. **Comment 257542852**: This comment contains explicit language and is considered toxic. Both models correctly predicted it as toxic.

2. **Comment 78309297**: Similar to the previous comment, this one also contains offensive language and can be categorized as toxic. Both models correctly predicted it as toxic.

3. **Comment 253911271**: This comment seems to be a factual statement about the classification of the Democratic Republic of Congo. It does not contain any explicit lan-

guage or toxic content. Both models correctly predicted it as toxic which is misclassification.

4. **Comment 419746843**: This comment discusses the manipulation of a report and uses offensive language towards the source. The discriminative model correctly predicted it as toxic, while the generative model failed to do so.

5. **Comment 635744705**: This comment appears to be an advertisement or a statement about copyright issues. While it contains offensive language, it may not necessarily be classified as toxic in the context of the dataset. The generative model incorrectly predicted it as toxic, while the discriminative model correctly identified it as non-toxic.

4 Summary

4.1 Discussion of work carried out

The project aimed to identify toxic comments using two models, a Generative Model based on Gaussian Naive Bayes and a Discriminative Model utilizing Gradient Boosting Classifier.

Results: Generative Model achieved an accuracy of 0.595, with a recall of 0.352 and precision of 0.126 and Discriminative Model performed significantly better with an accuracy of 0.844, a recall of 0.054, and precision of 0.185.

Challenges: Despite the generative model exhibiting a higher recall, both models struggled with low precision and high false positive rates **Future Work**:

- **Dataset Improvement**: Consider utilizing a more extensive dataset to ensure a broader range of toxic comments is covered, addressing potential biases and improving model generalization.

- **Model Architecture**: Explore advanced deep learning architectures such as Convolutional Neural Networks (CNN) or Transformer-based models to capture more complex patterns within the data.

- **Text Preprocessing**: Apply more advanced text preprocessing techniques, such as character-level tokenization, to enhance feature extraction and model performance.

- **Ensemble Methods**: Investigate ensemble methods, such as model blending or stacking, to combine the strengths of different models and improve overall predictive performance.

- **Hyperparameter Optimization**: Optimize the hyperparameters of existing models to fine-tune their performance and achieve better results.

- **Addressing Class Imbalance**: Handle class imbalance issues through techniques like oversampling or using appropriate performance metrics that prioritize recall, focusing on effectively identifying toxic comments.

4.2 Lessons Learned

Through experimenting with this project, I've not only gained proficiency in implementing models for text data analysis and understanding the fundamentals of Natural Language

Processing (NLP) but also identified the significance of hyperparameters and their role in improving model performance. Additionally, I've learned the importance of evaluation metrics in assessing model effectiveness. By comparing and contrasting various models, such as the Generative Model (Gaussian Naive Bayes) and the Discriminative Model (Gradient Boosting Classifier), for identifying toxic comments, I've expanded my understanding of model selection and optimization.

Furthermore, this project has highlighted the necessity of continuous experimentation and exploration of different techniques. As I move forward, I am better equipped to handle challenges such as dataset enhancements, exploring advanced model architectures, and addressing class imbalance. Completing this project has provided me with valuable information into the steps required to enhance the performance of text classification models and has paved the way for future improvements and advancements in this domain.

References

[1] Md. Nazmul Abdal, Md. Azizul Haque, Most. Humayera Kabir Oshie, and Sumaya Rahman. Multilingual toxic comment classification using bidirectional lstm. In R. N. Shaw, P. Siano, S. Makhilef, A. Ghosh, and S. L. Shimi, editors, *Innovations in Electrical and Electronic Engineering. ICEEE 2023*, Lecture Notes in Electrical Engineering, pages 273–284, 2024. doi: 10.1007/978-981-99-8661-3_23.

[2] A. Bhavani and B. Santhosh Kumar. A review of state art of text classification algorithms. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1484–1490, 2021. doi: 10.1109/ICCMC51019.2021.9418262.

[3] Eunjung Kwon, Hyunho Park, Sungwon Byon, and Kyu-Chul Lee. Improving text classification performance through data labeling adjustment. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 2277–2279, 2022. doi: 10.1109/ICTC55196.2022.9953026.

[4] M. Janaki Meena and K. R. Chandran. Naïve bayes text classification with positive features selected by statistical method. In *2009 First International Conference on Advanced Computing*, pages 28–33, 2009. doi: 10.1109/ICADVC.2009.5378273.

[5] Saurav Pradha, Malka N. Halgamuge, and Nguyen Tran Quoc Vinh. Effective text data preprocessing technique for sentiment analysis in social media data. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–8, 2019. doi: 10.1109/KSE.2019.8919368.

[6] Swapnil Singh, Deepa Krishnan, Pranit Sehgal, Harshit Sharma, Tarun Surani, and Jayant Singh. Gradient boosting approach for sentiment analysis for job recommendation and candidate profiling. In *2022 IEEE Bombay Section Signature Conference (IBSSC)*, pages 1–6, 2022. doi: 10.1109/IBSSC56953.2022.10037443.