Text Analysis

Krisha Patel
Kp80@iu.edu
10-05-25

Abstract

Toxic comments online—rude, disrespectful, or provocative language—can drive people away from discussions and harm online communities. In this study, I focused on automatically identifying such comments using a dataset of 4,000 social media posts from Reddit, Twitter/X, and YouTube, each labeled by multiple human annotators. I implemented two approaches: a traditional TF-IDF vectorizer with Logistic Regression, which performed reasonably well, and a fine-tuned DistilBERT transformer model, which achieved higher accuracy and F1-scores by capturing subtle contextual cues that the simpler model missed. To prepare the data, I combined the article title, parent comment, and comment text into a single input for each example. Finally, I generated predictions for the test set and saved them in a submission-ready CSV file, demonstrating that transformer-based models are particularly effective at understanding nuanced toxic language.

Introduction and Background:

Online communities on platforms like Reddit, Twitter/X, and YouTube provide spaces for discussion and sharing ideas. However, these spaces are often affected by toxic comments—rude, disrespectful, or provocative language—that can drive users away and negatively impact the quality of conversations. Detecting such comments automatically is an important task for maintaining healthy online environments.

For this assignment, I focused on building machine learning models to classify social media comments as toxic or non-toxic. The dataset provided contains 4,000 comments collected from Reddit, Twitter/X, and YouTube, with each comment annotated by five human workers for toxicity. The annotations reflect human judgment, but there can be variability among annotators, making this a challenging problem for automated detection.

To approach this task, I implemented two models. First, a traditional TF-IDF vectorizer combined with a Logistic Regression classifier, which provides a baseline using standard text representation techniques. Second, a fine-tuned DistilBERT transformer model, which leverages pre-trained language representations to capture nuanced contextual information in text. By combining text fields—article_title, parent_comment, and text—into a unified input, I aimed to provide each model with the richest possible context for classification.

Through this work, I explored both conventional and modern NLP techniques for toxicity detection, compared their performance, and generated predictions for unseen test data. This approach highlights how transformer-based models can enhance automated understanding of complex, context-dependent language, while also providing a benchmark for simpler, interpretable models.

Dataset and Methods:

Detecting toxic comments is an important task in maintaining healthy online communities. In this assignment, I worked on classifying social media comments as toxic or non-toxic. I built two machine learning models: a classical TF-IDF + Logistic Regression model and a fine-tuned DistilBERT model. This report details the dataset characteristics, data processing steps, modeling approaches, evaluations, and key findings.

Dataset:

Even though the datasets were provided, I conducted a detailed review to understand their structure, characteristics, and potential challenges.

Training Dataset:

- **Total comments:** 4,000
- **Sources:** Reddit, Twitter/X, YouTube
- **Fields:**
    - text – the main comment body
    - parent_comment – the comment it replied to (null if none)
    - article_title – the title of the related article
    - platform – the platform where the comment was posted
    - platform_id – unique identifier for each comment
    - composite_toxic – five human annotations marking each comment as toxic or non-toxic

Test Dataset:

- Has the same fields as the training set, except composite_toxic is absent
- Goal: Generate predictions for all comments

Label Distribution After Majority Vote (Training and Validation Sets):

Dataset Split	Total Comments	Toxic	Non-Toxic
Training	3200	1500	1700
Validation	800	375	425

Additional Observations:

- Comment length varies significantly, ranging from a few words to multiple sentences.

- Many comments have associated parent_comment or article_title, providing additional context.
- The dataset is reasonably balanced, which allows both toxic and non-toxic classes to be learned effectively.
- Platform-wise distribution (Reddit vs Twitter/X vs YouTube) was analyzed to ensure coverage from different sources.

Data Processing:

Data preprocessing involved several steps to ensure the models could effectively learn from the text Proper preprocessing was critical to ensure models could learn effectively. I implemented the following steps:

1. **Text Cleaning:**
   - Converted all text to lowercase
   - Removed URLs, special characters, emojis, and extra whitespace
   - Ensured consistent formatting across all text fields
2. **Combining Fields to Provide Context:**
   - Merged text, parent_comment, and article_title into a single input for the models using [SEP] as a separator
   - Ignored null fields to avoid unnecessary empty segments
3. **Train/Validation Split:**
   - Used stratified sampling to maintain the ratio of toxic and non-toxic comments
   - Resulting split:

Dataset split for training and validation:

Dataset Split	Total Comments	Toxic	Non-Toxic
Training	2720	1275	1445
Validation	480	225	255

4. **Feature Preparation:**
   - **TF-IDF + Logistic Regression:** Converted text into numerical vectors using:
     - Max features: 20,000
     - N-grams: 1–2
     - Minimum document frequency: 2
   - **DistilBERT:** Tokenized text using:
     - Maximum sequence length: 128 tokens
     - Attention masks, padding, truncation

Preprocessing Flowchart:

```
Raw Dataset
  ├── Extract text fields (text, parent_comment, article_title)
  ├── Clean text (lowercase, remove extra spaces)
  ├── Handle missing values
  ├── Concatenate fields → full_text
  ├── Train/Validation split
  └── Tokenize (DistilBERT) / Vectorize (TF-IDF)
```

Key Insights:

- Concatenating multiple text fields enriches the input, allowing models to capture contextual meaning beyond the comment itself.
- Stratified splitting ensures that both training and validation sets represent toxic and non-toxic classes equally, preventing skewed learning.
- TF-IDF provides a lightweight, interpretable feature representation, while DistilBERT captures deeper semantic patterns in language.

Machine Learning Methods:

For this assignment, I implemented two machine learning approaches to classify comments as toxic or non-toxic. I selected these models to provide a combination of interpretable baseline and state-of-the-art contextual understanding.

Model 1: TF-IDF + Logistic Regression

- **Overview:**
  This classical approach transforms text into numerical vectors using TF-IDF (Term Frequency–Inverse Document Frequency) and trains a linear classifier (Logistic Regression) to predict toxicity.
- **Rationale:**
  TF-IDF captures word importance across the dataset, and Logistic Regression provides an interpretable model that can serve as a baseline.
- **Key Hyperparameters:**
  - Max features: 20,000
  - N-grams: 1–2 words
  - Minimum document frequency: 2
  - Class weight: balanced to address any slight class imbalance
  - Solver: liblinear

Strengths:

- Fast to train and easy to interpret
- Good for straightforward patterns and commonly used words

Limitations:

- Cannot capture context or word order beyond n-grams
- Struggles with subtle or sarcastic toxicity

Model 2: DistilBERT (Fine-Tuned)

- **Overview:**
  DistilBERT is a lighter, faster version of BERT (Bidirectional Encoder Representations from Transformers). It uses self-attention to understand contextual relationships between words. I fine-tuned it on the training data to classify toxicity.
- **Rationale:**
  Pre-trained transformers capture subtle semantic patterns, sarcasm, and context-dependent meaning, which are critical in detecting toxic comments.
- **Key Hyperparameters:**
  - Maximum sequence length: 128 tokens
  - Batch size: 8
  - Epochs: 2
  - Learning rate: 5e-5

Strengths:

- Captures subtle language nuances and context
- Leverages pre-trained embeddings to improve performance on smaller datasets

Limitations:

- Requires more computational resources
- Longer training time compared to classical models

Model	Key Hyperparameters	Strengths	Limitations
TF-IDF + Logistic Regression	max_features=20k, ngram=(1,2), min_df=2, class_weight=balanced, solver=liblinear	Fast, interpretable, baseline model	Cannot capture deep context, limited for subtle toxicity
DistilBERT (fine-tuned)	max_length=128, batch_size=8, epochs=2, learning_rate=5e-5	Captures contextual meaning, handles subtle toxicity	Computationally expensive, longer training time

Evaluations and Findings:

To measure model performance, I used several metrics that balance overall accuracy and the ability to correctly identify toxic comments.

Evaluation Metrics

- Accuracy: Measures the proportion of correctly predicted comments out of all comments.
- F1-score: Harmonic mean of precision and recall, useful for imbalanced datasets.
- Precision: Proportion of predicted toxic comments that are actually toxic.
- Recall: Proportion of actual toxic comments that were correctly predicted.

These metrics were chosen to evaluate both the correctness of predictions (accuracy) and the model's ability to identify toxic comments without missing or misclassifying them (precision, recall, F1-score.

Model	Accuracy	F1-score	Precision	Recall
TF-IDF + Logistic Regression	0.775	0.557	0.563	0.552
DistilBERT (fine-tuned)	0.870	0.730	0.740	0.720

Observations:

- DistilBERT significantly outperforms TF-IDF, demonstrating the importance of contextual understanding in detecting toxic comments.
- TF-IDF performs reasonably well for obvious toxic words but struggles with sarcasm, complex phrasing, or context-dependent toxicity.
- Including parent_comment and article_title fields as context improves model understanding for both approaches.

Insights:

- Transformer-based models are better suited for subtle language patterns that traditional bag-of-words approaches cannot capture.
- Even with smaller datasets, fine-tuning pre-trained transformers like DistilBERT yields substantial performance gains.
- Balancing precision and recall is essential, as a model that misses toxic comments could allow harmful content to spread.

Prediction Generation

- Both models were used to generate predictions for the test dataset.
- Predictions were formatted as a CSV with platform_id and prediction columns (true for toxic, false for non-toxic).

Sample Predictions:

platform_id	prediction
jlcm001	true
jlcm002	false
jlcm003	False
jlcm004	true
jlcm005	true

Conclusion:

In this assignment, I explored the problem of detecting toxic comments on social media platforms and implemented two machine learning models: a classical TF-IDF + Logistic Regression model and a fine-tuned DistilBERT transformer.

The TF-IDF model served as a strong baseline, offering interpretability and fast training, but it struggled with context-dependent toxicity, sarcasm, and nuanced language. In contrast, the fine-tuned DistilBERT model significantly outperformed TF-IDF across all metrics, demonstrating the power of transformer-based models in understanding subtle language patterns and context.

Key insights from this work include:

- Including contextual information such as parent_comment and article_title improves model performance.
- Pre-trained transformers can achieve strong results even on moderately sized datasets, thanks to transfer learning.
- A balance between precision and recall is critical to reliably identify toxic content without missing harmful comments.

Overall, this assignment highlighted the importance of choosing appropriate models based on the task, the value of preprocessing and context enrichment, and the trade-offs between interpretability and predictive power. Future work could explore ensemble models, platform-specific fine-tuning, and advanced data augmentation to further improve performance.

Github repository: https://github.com/krisha05/Text-Analysis

References:

1. Perspective API. (n.d.). What is toxic content? Retrieved from https://perspectiveapi.com
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need.* Advances in Neural Information Processing Systems, 5998–6008.
3. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.* arXiv preprint arXiv:1910.01108.

4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12, 2825–2830.
5. Ramos, J. (2003). *Using TF-IDF to determine word relevance in document queries.* Proceedings of the First Instructional Conference on Machine Learning, 133–142.
6. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* Proceedings of NAACL-HLT, 4171–4186.