Online Toxicity Analysis: Comparative Evaluation of Text Classification and Topic Modeling Techniques

Kevin Garofalo[1], Giulio Lonati[1], Vincenzo Siano[1]

Abstract
Online toxic discourse poses a significant challenge for large-scale content moderation, requiring automated methods capable of handling language. In this work, we conduct a comparative study of text mining techniques on the Jigsaw Toxic Comment Classification dataset, addressing both comment classification and topic modeling. For classification, we benchmark traditional statistical models based on TF-IDF representations (such as Logistic Regression and Linear SVM) against a fine-tuned DistilBERT model, evaluating their robustness under strong class imbalance using ROC-AUC and macro-averaged F1 scores. For topic modeling, we compare classical bag-of-words approaches (LDA and NMF) with BERTopic, an embedding-based method leveraging contextual sentence representations and clustering. Our analysis highlights the limitations of lexical models in capturing nuances and the advantages of contextual embeddings in discovering coherent and well-separated themes.

Keywords
Text Mining — Preprocessing — BERT — Text Classification — Topic Modeling

[1] *CdLM in Data Science, Università degli Studi di Milano-Bicocca*

Contents

1. Introduction

Discussing topics that matter can be challenging in online environments. Online harassment and hate speech severely degrade online discourse, forcing many platforms to limit user interaction. As user-generated content grows, manual moderation becomes operationally unfeasible, necessitating robust automated systems capable of detecting offensive content at scale.

This project addresses this challenge through a comparative study of Text Mining techniques using the **Jigsaw Toxic Comment Classification Challenge** dataset [1]. Unlike standard sentiment analysis, detecting toxicity involves capturing complex nuances like indirect aggression and identity-based hate. The analysis specifically tackles the dataset's severe *class imbalance* and *multi-label* structure, where a single comment often belongs to multiple toxicity categories simultaneously.

The primary objective is to evaluate the trade-offs between **statistical machine learning** and modern **Deep**

Learning architectures across two complementary NLP tasks:

- **Text Classification**: we benchmark traditional classifiers (e.g., *Logistic Regression, Linear SVC*) using TF-IDF representations against a fine-tuned *DistilBERT* transformer. The study focuses on handling class imbalance through targeted training strategies and utilizes robust metrics, such as *ROC-AUC* and *macro-averaged F1*, to accurately assess multi-label performance.
- **Topic Modeling**: we compare classical generative models, specifically *Latent Dirichlet Allocation (LDA)* and *Non-negative Matrix Factorization (NMF)*, with *BERTopic*, a neural embedding-based approach. While classification assigns predefined labels, topic modeling uncovers the latent thematic structures of hate speech. This section investigates whether deep learning approaches produce more interpretable and semantically coherent topics compared to traditional probabilistic methods.

2. Dataset

The dataset used in this project is sourced from the **Jigsaw Toxic Comment Classification Challenge**, a widely used benchmark for toxicity detection. It consists of user-generated comments extracted from Wikipedia talk pages, each annotated by human raters for the presence of different forms of toxic behavior. Each comment is associated with six binary labels representing distinct toxicity categories: **toxic**, **severe_toxic**, **obscene**, **threat**, **insult**, and **identity_hate**. The task is formulated as a **multi-label classification problem**, meaning that a single comment may simultaneously belong to multiple categories.

The dataset is divided into a training split containing 159,571 comments and a test split containing 153,164 comments. Unlike many standard benchmarks, the test set also includes ground-truth annotations; however, some label values are set to -1, indicating that the corresponding labels were excluded from scoring during the original Kaggle competition. These values were introduced by the dataset providers to avoid manual imputation and to prevent participants from exploiting partially annotated labels.

2.1 Data Cleaning

To ensure a consistent and fully supervised learning setup (i.e., only for text classification task), a series of *data cleaning steps* were applied. In particular, we removed all comments containing at least one label with value -1, keeping only instances with complete and reliable annotations across all toxicity categories. After this filtering step, we ended up with a test set made up of 63,978 comments.

By contrast, given the unsupervised nature of topic modeling, we aggregated the training and testing datasets to maximize the corpus size for this task. We then restricted the dataset to comments exhibiting at least one form of toxicity (i.e., at least one label set to 1), resulting in a subset of 22,468 comments. This choice was motivated by the highly imbalanced nature of the dataset: modeling topics on the full corpus would be dominated by neutral language and would obscure thematic patterns specific to toxic discourse. At a later time, to support a more fine-grained investigation of toxic discourse, we furthermore restricted the dataset to comments labeled as *identity_hate*, obtaining a subset of 2,117 instances.

2.2 EDA

After cleaning and converting the data into the desired format, an initial *exploratory analysis* was performed on the training dataset. For visualization purposes, we introduced a temporary "clean" label to group comments whose original labels were all set to zero. This allowed for a clearer comparison between toxic and non-toxic comments. The analysis confirmed a severe class imbalance, with approximately 90% of the comments categorized as non-toxic. Such imbalance poses a significant challenge for classification models, as naive predictors might achieve high accuracy by simply favoring the majority class while failing to detect rare but critical forms of toxicity (e.g., threats). The distribution of labels is summarized in Table 1.

Table 1. Distribution of toxicity labels in the training set.

Label	Count	Percentage
toxic	15,294	9.58%
severe_toxic	1,595	1.00%
obscene	8,449	5.29%
threat	478	0.30%
insult	7,877	4.94%
identity_hate	1,405	0.88%
clean (non-toxic)	*143,346*	*89.83%*

We also analyzed comment length, showing that clean comments tend to have a higher median length and

a wider dispersion compared to toxic comments. However, the actual overlap between the two distributions indicates that comment length alone is insufficient as a discriminative feature for toxicity detection. Finally, we generated *word clouds* for both toxic and non-toxic subsets (Fig. 1) to qualitatively inspect the most frequent terms. While non-toxic comments are dominated by neutral and discussion-oriented vocabulary, toxic comments exhibit a higher concentration of offensive language, further motivating the use of modeling approaches which capture semantics.

Figure 1. Word clouds contrasting the vocabulary of toxic and non-toxic subsets.

3. Text Preprocessing

Text preprocessing plays a crucial role in reducing noise and ensuring that textual data is suitable for modeling. In this project, a general preprocessing pipeline was applied uniformly across all experiments to remove sources of noise unrelated to linguistic content, including:

- removal of HTML tags and markup artifacts;
- removal of URLs, IP addresses, and user mentions;
- normalization of repeated characters to mitigate misspellings;
- elimination of redundant whitespace, including newlines and tab characters.

These operations preserve the semantic structure of the text while reducing variability caused by formatting and non-linguistic elements. After that, preprocessing strategies diverged according to the architectural characteristics of the models employed. Minor variations in preprocessing were introduced when supported by empirical validation.

3.1 Statistical-based Approaches

Traditional statistical models [2] (e.g., *Logistic Regression* and *LDA*) rely on sparse, frequency-based representations (i.e., *Bag of Words* or *TF-IDF*) and therefore benefit from stronger normalization of the input text to reduce feature dimensionality. For these models, we applied:

1. **tokenization**, to segment text into individual terms;
2. **lemmatization**, reducing words to their canonical dictionary forms;
3. **stop-word removal**, filtering out high frequency terms (e.g., “the”, “and”) with limited discriminative power.

This pipeline is aligned with common practices for sparse vector representations and helps reduce vocabulary size while emphasizing semantically meaningful tokens.

3.2 Deep Learning-based Approaches

For **deep learning** experiments involving *DistilBERT* and *BERTopic*, no additional linguistic preprocessing was applied beyond the initial cleaning stage. Transformer-based architectures rely on **subword tokenization algorithms** (e.g., *WordPiece* for DistilBERT and *SentencePiece*/*WordPiece* for the sentence-transformers used in BERTopic) and are designed to capture complex semantic and contextual dependencies. As a result, operations such as stop-word removal or lemmatization were **intentionally avoided**, as they may degrade performance by breaking up contextual information relevant to toxicity detection, including sarcasm and implicitly abusive language [3].

4. Text Classification

4.1 Text Representation

The choice of **text representation** is an important factor in determining the effectiveness of classification models. In this work, different representation strategies were adopted depending on the modeling paradigm, reflecting the distinct assumptions underlying statistical machine learning methods and deep learning architectures. In this

way, this project enables a comparison between traditional **bag-of-words approaches** and **modern contextual language models** in a multi-label toxicity detection setting.

4.1.1 Bag of Words

For statistical classification approaches, text was converted into **Term Frequency (TF)** or **Term Frequency-Inverse Document Frequency (TF-IDF)** representations. More specifically:

1. **TF**: simply counts how many times a word appears in a specific comment.
2. **IDF**: checks the rarity of a word across the entire corpus.

This logic makes it more suitable for linear classifiers operating on high-dimensional feature spaces. Practically, the both representations were implemented using Python's **scikit-learn** library, which provides a robust and efficient framework for text vectorization and classification. The vectorizers were fitted on the training data and applied consistently across all baseline models to ensure fair comparison.

4.1.2 Contextualized Word Embeddings

In contrast to statistical models, the deep learning approaches rely on **contextualized word embeddings** derived from **Transformer** architectures. For the *DistilBERT-based classifier*, text is encoded into dense, contextualized representations that capture semantic meaning as well as word order and syntactic structure. These embeddings are learned jointly with the classification objective during **fine-tuning**. Unlike TF-IDF representations, these embeddings allow the model to distinguish word meaning based on surrounding context, which is especially important for detecting various forms of toxicity.

4.2 Machine Learning Classifiers

As a first baseline, we employed **Naive Bayes classifiers** using raw *unigram* counts as features. Both **Multinomial Naive Bayes** and **Complement Naive Bayes** were evaluated, the latter being specifically designed to improve robustness in highly imbalanced settings. Model performance was assessed using **5-fold cross-validation**, with averaged *ROC-AUC* as the primary evaluation metric.

Subsequently, raw token counts were replaced with **TF-IDF representations**, which weight terms based on their importance within a document relative to the entire corpus. Using this representation, we evaluated the following linear classifiers:

- **Logistic Regression**: trained with balanced class weights to mitigate label imbalance; here, the algorithm estimates the coefficients (β) that maximize the log-likelihood function.
- **Linear Support Vector Classification (Linear SVC)**: an implementation of *Support Vector Machines (SVM)* that learns a maximum-margin linear decision boundary.
- **Stochastic Gradient Descent (SGD)**: used as a scalable optimization framework for linear models. By selecting appropriate loss functions, both logistic regression (*log loss*) and linear SVM (*hinge loss*) formulations were implemented.

All models were trained under a *One-Vs-Rest* scheme and incorporated **balanced class weighting** to penalize misclassification of underrepresented toxicity labels more strongly. To improve performance, *hyperparameter optimization* was conducted using a **randomized search strategy**. The optimization pipeline jointly tuned both the text representation and classifier parameters, including the *n-gram range*, *document frequency thresholds*, and *regularization strength*. Then, **model selection** was performed via cross-validation using *ROC-AUC* as the scoring function.

Finally, given the multi-label and highly imbalanced nature of the task, a label-specific **threshold tuning** procedure was applied. Instead of relying on the default probability threshold (i.e., 0.5), optimal thresholds were selected independently for each label by maximizing the *F1 score* on **validation data**. This post-processing step enables more balanced precision-recall trade-offs, particularly for rare toxicity categories.

4.3 Deep Learning Classifier

While statistical models provide efficient and interpretable baselines, the semantic complexity of toxic language motivates the use of a **contextual deep learning approach** [4]. For this reason, we fine-tuned **DistilBERT**, a transformer-based architecture designed to balance performance and computational efficiency. DistilBERT is a compressed version of **BERT** that retains approximately **97% of its performance** while being **40% smaller and 60% faster**, making it well suited for large-scale text classification tasks [5]. The model consists of a pre-trained transformer backbone followed by a **task-specific classification head**. The backbone produces

contextualized token representations, while the classification head maps the pooled representations to **six output logits**, one for each toxicity label.

Fine-tuning was performed in a multi-label setting using the **binary cross-entropy loss with logits**. A *weighted variant* of the loss was also evaluated to address class imbalance, but it did not yield performance improvements. Optimization was carried out using **AdamW optimizer** with a **linear learning rate scheduler**. Alternative configurations, including standard *Adam* and learning rate *warmup*, were tested but provided no measurable gains.

Several optimization strategies were employed to ensure **stable and efficient training**. Batching was handled via dedicated PyTorch's **DataLoaders**, which implemented **dynamic padding**, which pads sequences only to the longest sequence within the current batch rather than the global maximum. A maximum sequence length of **256 tokens** was enforced; increasing this limit to 512 substantially increased training time with negligible performance improvements, suggesting that relevant toxicity indicators typically occur early in the text. **Gradient clipping** was applied to prevent **exploding gradients**. The model was fine-tuned for **two epochs**, which appeared to be sufficient to achieve convergence without overfitting.

4.4 Evaluation

Evaluating performance in toxic comment classification requires particular attention due to the **multi-label structure** of the task and the **severe class imbalance** present in the dataset. Consequently, standard **accuracy** was not considered an appropriate metric, as it would be dominated by the majority non-toxic class and fail to reflect performance on rare but critical labels.

As the primary evaluation metric, we adopted the **mean column-wise ROC-AUC**, computed as the average of the individual ROC-AUC scores obtained for each toxicity label. Formally, the AUC for a specific class c is defined as the probability that a randomly chosen positive instance x_c^+ is ranked higher than a randomly chosen negative instance x_c^-:

$$\mathrm{AUC}_c = P\left(s(x_c^+) > s(x_c^-)\right)$$

where $s(\cdot)$ denotes the predicted score, and x_c^+ and x_c^- represent positive and negative samples for class c, respectively. The final score is obtained by averaging across all C labels:

$$\text{ROC-AUC}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{AUC}_c$$

This metric is well suited for multi-label classification, as it evaluates the model's ability to rank positive instances above negative ones independently for each class, and is less sensitive to class imbalance. In addition to the aggregated score, **per-label ROC–AUC values** were also reported to observe performance variability across different types of toxicity.

To complement ROC-AUC, we also computed the **macro-averaged F1 score**, which provides a **threshold-dependent perspective** on classification performance. For each label c, the F1 score is defined as the harmonic mean of precision and recall:

$$\mathrm{F1}_c = \frac{2 \cdot \mathrm{Precision}_c \cdot \mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}$$

The macro-averaged F1 score is then computed as:

$$\mathrm{F1}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{F1}_c$$

It is important to note that the F1 scores for the statistical approaches were computed using the **optimal decision thresholds** derived from the tuning procedures described at the end of Section 4.2. Rather than applying a default threshold of 0.5, these label-specific thresholds were the ones that generated the maximized macro-averaged F1 score on the **cross-validated folds**, to ensure a fair comparison of the models' peak performance capabilities. Finally, by treating all labels equally, macro F1 highlights a model's ability to detect rare toxicity categories such as *threat* and *identity_hate*. Therefore, this metric is particularly informative when evaluating the impact of class weighting and post-training threshold tuning.

4.5 Results

Our experimental results are divided into two phases: the selection of the best statistical baseline model and the final comparative evaluation against the neural approach. Logistic Regression and Stochastic Gradient Descent (SGD) emerged as the top performers with respect to ROC-AUC, so we decided to tune their hyperparameters. As shown in Table 2, tuned Logistic Regression and tuned SGD, both trained on the TF-IDF representation, confirmed to be the best algorithms; although

SGD offered a competitive training speed, Logistic Regression achieved the highest validation performance. Consequently, we selected the tuned Logistic Regression model as our final machine learning model.

Table 2. Comparison of models during evaluation.

Model	ROC-AUC
Naive Bayes (TF)	0.8720
Logistic Regression (TF-IDF)	0.9806
LinearSVC (TF-IDF)	0.9688
SGD (Log Loss, TF-IDF)	0.9769
SGD (Hinge Loss, TF-IDF)	0.9797
DistilBERT	0.9922

We finally tested the tuned Logistic Regression model on the held-out test set: the observed results confirm the strong ranking capabilities of the model, although underlining its difficulty of making precise binary decisions on rare classes without deep contextual understanding. Accordingly, we moved towards a deep learning-based fine-tuning a DistilBERT model, which instantly **outperformed the statistical baseline**. Precisely, it reached high performance on the validation set and maintained it on the test set as well (Tab. 3). This highlights the enhanced capacity to capture semantics, leading to better precision and recall balance on rare labels.

Table 3. Comparison of the best models on the test set.

Model	ROC-AUC	F1 Macro
Logistic Regression	0.9618	0.50
DistilBERT	0.9865	0.61

5. Topic Modeling

5.1 Text Representation

As with text classification, different representation strategies were adopted for a meaningful comparison between probabilistic and modern embedding-based topic models.

For **Latent Dirichlet Allocation (LDA)** and **Non-negative Matrix Factorization (NMF)**, documents were represented using a **bag-of-words** formulation, thus without considering word order or contextual information. This approach is consistent with the assumptions of both LDA and NMF, which model topics as distributions over words and rely on word co-occurrence patterns to infer latent themes.

On the other hand, **BERTopic** relies on **dense contextual embeddings** derived from transformer-based language models. Each document is mapped to a high-dimensional embedding that captures semantic meaning and contextual relationships between words. These embeddings enable BERTopic to cluster semantically similar documents even when they do not share **explicit vocabulary**, addressing a key limitation of bag-of-words approaches.

5.2 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a **generative probabilistic topic modeling** method that represents documents as mixtures of latent topics, where each topic is defined as a probability distribution over words. Documents are assumed to be generated by first sampling topic proportions and then sampling words from the corresponding topic distributions [6]. We trained multiple LDA models using the **LdaMulticore** implementation on the TF representation, exploring different configurations to control topic granularity and sparsity. In particular, we varied the **number of topics**, as well as the **Dirichlet priors** governing the *document-topic* (α) and *topic-word* (η) distributions. Across the tested configurations, the best topic representations were obtained when both α and η were set to *symmetric*, combined with a relatively small number of topics. This result suggests that documents in the corpus tend to exhibit balanced mixtures of topics and that the inferred topics are characterized by broadly distributed vocabularies rather than highly sparse word associations.

5.3 Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) [7] is a topic modeling technique based on linear algebra that decomposes the **TF-IDF document-term matrix** into two low-rank, non-negative matrices representing document-topic and topic-word relationships. The non-negativity constraint often leads to more interpretable topics compared to probabilistic models. We trained NMF models with varying numbers of latent components to assess the impact of topic granularity, focusing on configurations with **5 and 10 topics**. These settings allowed us to examine the trade-off between raw and more fine-grained thematic structures, providing a direct comparison with LDA under similar settings.

5.4 BERTopic

We further explored **BERTopic** as a neural topic modeling approach that combines **contextual sentence embeddings** with clustering-based topic discovery. Document

representations were obtained using a pre-trained **SentenceTransformer** model, which maps each comment into a dense semantic embedding space capturing contextual and semantic similarity [8]. To facilitate clustering, the embeddings were projected into a lower-dimensional space using **UMAP**, which preserves local neighborhood structure while reducing noise. We experimented with UMAP hyperparameters controlling *neighborhood size* and *output dimensionality*, as well as clustering parameters in **HDBSCAN**, including *minimum cluster size* and *density thresholds*. Topic representations were then extracted by computing class-based term importance scores over the clustered documents using a *count-based vectorizer* with uni-grams and bi-grams. The final configuration employed a moderate *neighborhood size*, a *low-dimensional* UMAP projection, and a conservative *HDBSCAN setup*, favoring larger and denser clusters. This choice aimed to balance stability while limiting the formation of too specific and overly fragmented topics.

In addition to density-based clustering, we also evaluated **k-means** as a baseline alternative to HDBSCAN, experimenting with **5 and 10 clusters**, in order to compare it with the previous bag-of-word topic models. This comparison between clustering approaches provides insight into the differences between **centroid-based** and **density-based** topic discovery methods when applied to contextual embeddings.

5.5 Evaluation

Topic models were evaluated using complementary metrics that capture both semantic quality and model fit:

- **Topic Coherence**: computed using *Gensim* library, adopting the **c_v** metric due to its strong correlation with human interpretability. Coherence measures the semantic similarity between the most representative words of a topic.
- **Topic Diversity (Lexical)**: for **LDA** and **NMF**, quantifies the proportion of unique words among the top terms across all topics, measuring lexical separation and redundancy. Higher diversity indicates more distinct topic vocabularies, addressing cases where models achieve high coherence but generate overlapping topics.
- **Topic Diversity (Semantic)**: for **BERTopic**, diversity evaluates the separation between topics in the embedding space rather than at the lexical level. This metric captures how distinct topic representations are based on their underlying document embeddings. Higher values indicate better separation between topics in terms of meaning, even when similar vocabulary may be used.
- **Perplexity and Reconstruction Error**: for **LDA**, model fit was assessed using **perplexity**, derived from the logarithm version and converted via exponentiation. Although lower perplexity indicates better likelihood fit, it does not necessarily imply more interpretable topics, as models with many topics often get lower perplexity but worse topics. For **NMF**, which lacks a probabilistic formulation, **reconstruction error** was used as an analogous measure of fit, with lower values indicating better approximation of the original TF-IDF matrix.

In addition to these quantitative metrics, we further assessed topic models' effectiveness and quality through **eye balling** by examining **top-10 words**, as well as visualizing topics and documents.

5.6 Results

5.6.1 Toxic-only Subset

We report results for both **bag-of-words–based models** (LDA, NMF) in Table 4, and **embedding-based approaches** (BERTopic) in Table 5, evaluated on the toxic-only subset using *coherence*, *topic diversity* (*lexical* for LDA and NMF, *semantic* for BERTopic), and model-specific fit metrics.

Table 4. Comparison of bag-of-words topic models.

Model	Coherence	Diversity	Perplexity
LDA (k=5)	0.514	0.96	0.0010
LDA (k=10)	0.515	0.97	0.0011
Model	**Coherence**	**Diversity**	**Rec. Error**
NMF (k=5)	0.490	0.92	147.1227
NMF (k=10)	0.457	0.69	145.6265

LDA achieved **moderate coherence scores** across both configurations, with comparable performance for 5 and 10 topics. Perplexity values were similarly low, indicating a stable likelihood fit, while **lexical diversity remained high**, suggesting limited redundancy among top words. Qualitatively, LDA produced **broad and interpretable topics**, often centered around high-frequency lexical patterns. Several topics capture recognizable themes such as *identity hate, sexual content,* and *generic insults*. However, substantial vocabulary overlap was observed. Topics tend to occupy **overlapping regions**, and the distribution of documents across topics is highly unbalanced. This behavior reflects LDA's reliance on

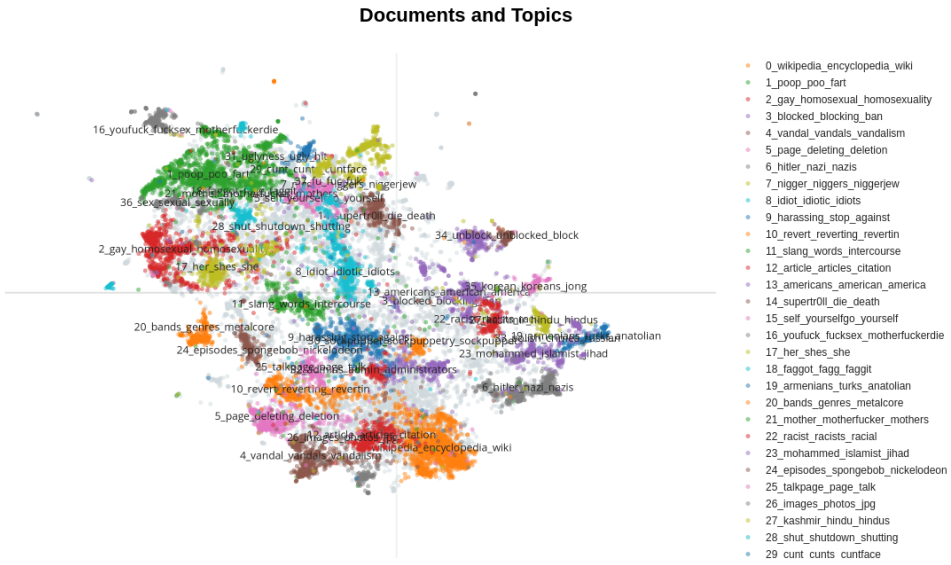

Figure 2. Documents inside the topics of BERTopic with HDBSCAN, useful to see if they make sense.

bag-of-words representations, which limits its ability to disambiguate semantically similar terms used in different contexts.

NMF exhibited **lower coherence and diversity than LDA** both in the 10-topic and 5-topic configurations. Compared to LDA, NMF generated **topics with sharper and more localized word distributions**, often improving word-level interpretability and yielding slightly better visual separation. However, several topics remain dominated by shared high-frequency terms, indicating continued reliance on surface-level lexical patterns rather than deeper semantic structure.

Table 5. Comparison of deep learning-based topic models (*toxic-only subset*).

BERTopic	Coherence	Semantic Diversity
HDBSCAN	0.367	0.528
k-means (k=5)	0.454	0.276
k-means (k=10)	0.455	0.329

Before experimenting with k-means clustering, we employed **BERTopic with HDBSCAN** (its default clustering method) using an automatic topic count to assess the model's natural behavior. While BERTopic achieved a **lower coherence score** compared to bag-of-words models, it exhibited a **moderate semantic diversity**, indicating well-separated and distinct topics in the embedding space. UMAP projections reveal **clearly defined cluster structures**, and HDBSCAN effectively isolates dense semantic regions while identifying noisy or ambiguous documents as outliers. Despite the presence of many outliers (over 8,000 documents even after hyperparameter tuning), as we can see from Fig. 2, **BERTopic demonstrated the strongest qualitative performance**, producing 38 semantically meaningful topics with reduced overlap. Many clusters capture **nuanced forms of toxic language** that are difficult to isolate using purely lexical models. A subset of topics showed reduced semantic coherence due to noise-related patterns such as usernames, timestamps, or metadata-like tokens.

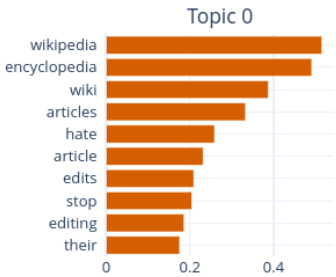

Figure 3. Bar chart of Wikipedia's topic terms.

Additionally, a prominent topic related to **Wikipedia-specific editorial language** emerged (Fig. 3), reflecting moderation discussions and page-edit commentary rather than toxicity.

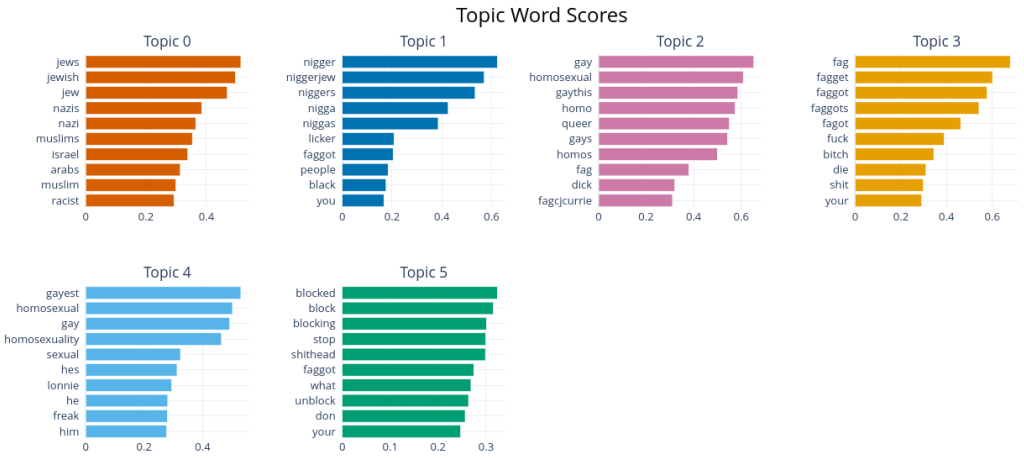

Figure 4. Bar chart of *identity_hate*'s topic terms.

This highlights BERTopic's sensitivity to contextual and structural patterns beyond abusive language alone. BERTopic models using **k-means clustering** on contextual embeddings produced **moderate coherence but low semantic diversity**, indicating substantial topic overlap. While useful as a baseline, centroid-based clustering proved less effective than density-based methods in capturing the heterogeneous structure of toxic discourse.

Overall, **LDA and NMF** provide interpretable baseline topics but struggle with semantic overlap, while **BERTopic** offers the most expressive and well-separated topic representations, at the cost of increased sensitivity to noise and outliers. Furthermore, BERTopic achieves comparatively lower coherence, since it is primarily designed for statistical and probabilistic topic models and does not fully capture the semantic structure learned by embedding-based approaches. In fact, these results highlight the advantages of **contextual embeddings** for modeling complex and nuanced forms of toxic language.

5.6.2 Identity Hate Subset

To further investigate fine-grained patterns of toxic discourse, we applied **BERTopic** to the subset of comments labeled as **identity_hate**, enabling a more focused analysis of this specific toxicity category. Despite the substantially smaller dataset, the resulting topics were **highly interpretable** and captured distinct sub-themes within the identity hate domain. We evaluated BERTopic using both **HDBSCAN** and **k-means** clustering strategies, with results reported in Tab. 6.

Similarly as before, the **HDBSCAN model** achieved greater semantic diversity at the cost of lower coherence, reflecting its ability to adaptively identify dense clusters

Table 6. Comparison of deep learning-based topic models (*identity_hate subset*).

Model	Coherence	Semantic Diversity
HDBSCAN	0.3399	0.4352
k-means (k=10)	0.3778	0.3913
k-means (k=5)	0.3670	0.3341

while isolating ambiguous or heterogeneous comments as outliers. This suggests that while centroid-based clustering may produce more internally consistent topics, it tends to group semantically adjacent themes together, leading to **increased overlap**. These differences are remarkably smaller than in the previous example.

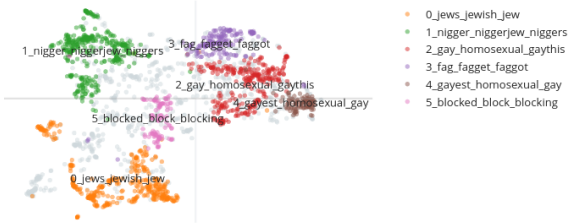

Figure 5. Documents inside the topics of BERTopic with HDBSCAN on *identity_hate*.

The topics extracted from this subset correspond to recognizable **identity-hate dimensions**, including ethnicity, religion, nationality, and gender-related hostility, as shown in Fig. 4 and Fig. 5. Overall, these results confirm that **contextual embedding topic modeling** is effective even on small and highly specialized datasets, and that **density-based clustering** is better suited for uncovering heterogeneous forms of identity-based hate.

6. Conclusions

The results discussed throughout this report offer a comprehensive comparison of traditional statistical methods and modern deep learning approaches for toxic content analysis. In the text classification task, we examined how different text representations and learning strategies perform under the challenges posed by severe class imbalance and the multi-label nature of the dataset, highlighting the advantages of contextual embeddings over purely lexical features. Complementing this quantitative evaluation, the topic modeling analysis provided qualitative insights into the structure of toxic comments, revealing recurring themes and nuanced patterns that are not directly observable through classification alone.

Future development could include incorporating ensemble strategies and model stacking, exploring approaches that combine classification and topic modeling, or scaling the models to multilingual datasets.

References

[1] cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. Toxic comment classification challenge. https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge, 2017. Kaggle.

[2] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[3] Michael Wiegand, Josef Ruppenhofer, and Elisabeth Eder. Implicitly abusive language – what does it actually look like and why are we not getting there? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 576–587, Online, June 2021. Association for Computational Linguistics.

[4] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. A survey on text classification: From shallow to deep learning. *arXiv preprint arXiv:2008.00364*, 2020.

[5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

[8] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.