MULTI-LABEL TOXIC COMMENT DETECTION

Using Supervised Classification and Unsupervised Clustering

Aarohi Mistry (925352)
Any Das (922710)

| PROJECT OVERVIEW & MOTIVATION

- Online hate speech is growing fast.
- Checking it manually is impossible
- Traditional keyword filters fail to detect implicit toxicity (e.g., sarcasm, context).

Our Solution (Dual-Strategy):

- **Classification**: For real-time, precise detection.
- **Clustering**: To discover hidden or emerging toxic patterns.

Objective: To quantify the "Contextual Premium" of Transformer models over traditional baselines.

| DATASET OVERVIEW & CLASS IMBALANCE

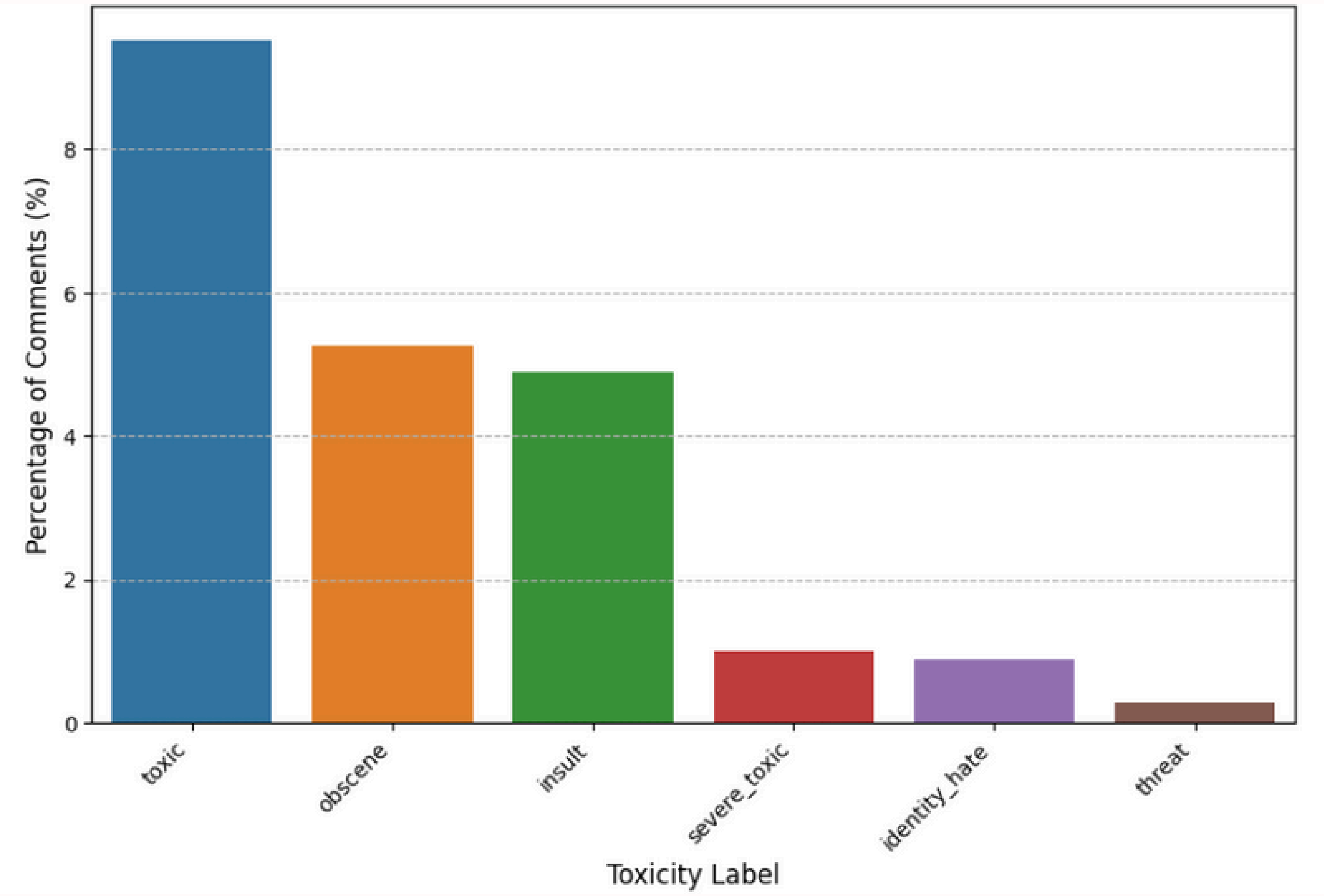
- Jigsaw Toxic Comment Classification Dataset (Wikipedia Talk Pages, Kaggle).
- Contains 159,571 real-world user comments.
- Multi-label Classification (e.g., Toxic + Insult).

Target Categories [6 Labels]:

- toxic, severe_toxic, obscene, threat, insult, identity_hate.

Challenge:

- Severe Class Imbalance.
- Dangerous classes like threat and identity_hate represent less than 1% of the data.



Label Distribution

| TEXT PRE-PROCESSING

- Data Integrity: Merged test labels and removed unscored entries (-1).
- Final Valid Test Set: 63,978 samples.

Statistical Pipeline (TF-IDF)

- Heavy Cleaning
- Aggressive Removal: Removed emojis, punctuation, URLs, and Stop-words.
- Normalization: Applied POS-aware Lemmatization to reduce words to base forms.
- Filtering: Removed duplicates.
- Result: Higher Data Loss (~3.68%)

Neural Pipeline (TextCNN / DistilBERT)

- Minimal Intervention
- Cleaning: Removed Emojis, URLs, HTML tags, and User Mentions.
- Context Preservation: Kept Stop-words and punctuation (Grammatical anchors).
- Integrity: Used Label-aware duplicate handling (kept duplicates if labels differed).
- Result: Minimal Data Loss (~0.1%)

| TEXT REPRESENTATION

Aspect	Statistical (TF-IDF)	Static Neural (FastText)	Contextual (Transformers)
Type	Sparse Matrix	Static Dense Embeddings	Dynamic Contextual Embeddings
Dimensions	100,000	300	768 (DistilBERT) / 384 (SBERT)
Key Feature	Used N-grams (1, 2) to capture toxic phrases (e.g., "shut up").	Sub-word information (Character n-grams)	Self-Attention Mechanism adapts to context.
Why used	Captures specific keywords and explicit slurs.	Handles OOV (Out-of-Vocabulary), slang, and misspellings (e.g., "fck"*).	Understands Polysemy (e.g., "kill the process" vs "kill you").

| TEXT CLASSIFICATION

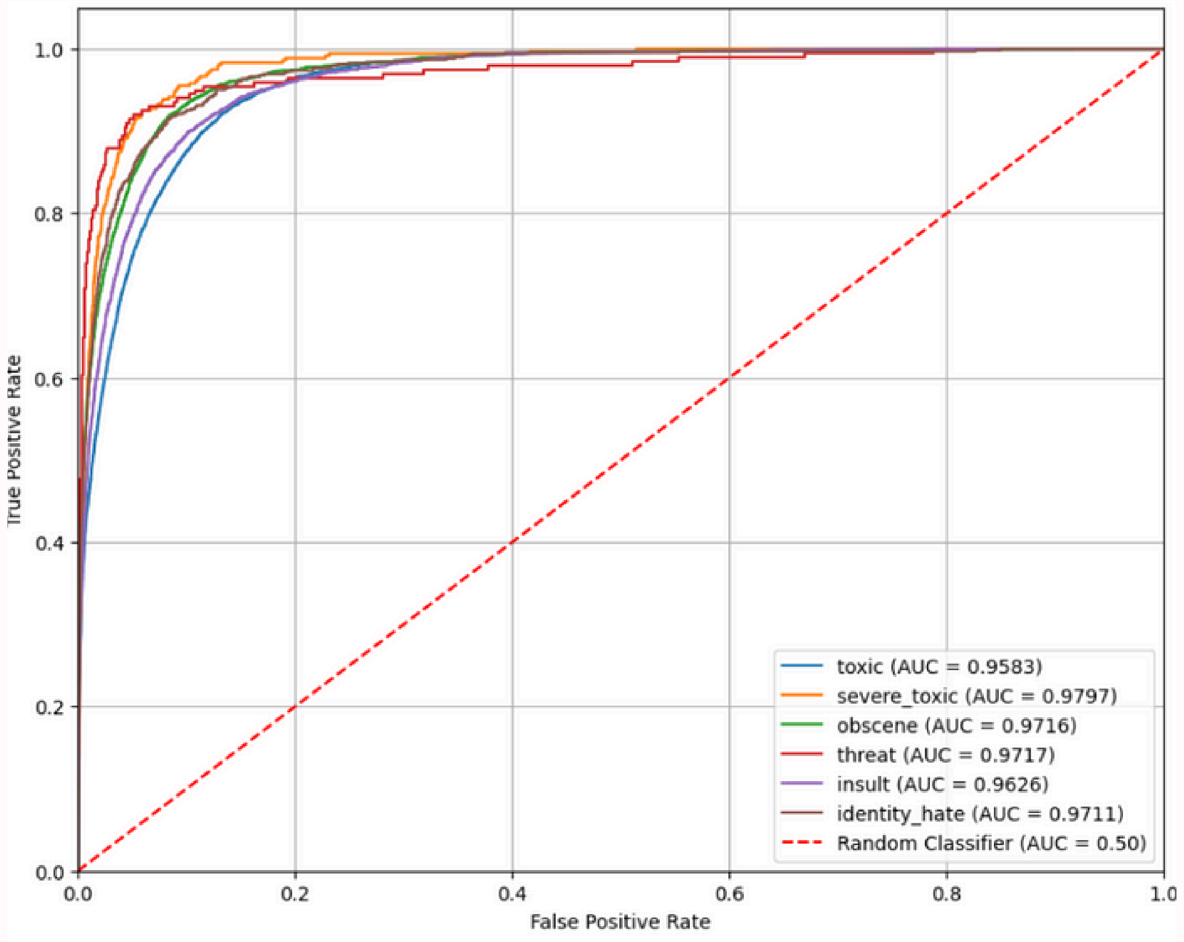
Model	Representation	Architecture	Macro ROC-AUC	Macro F1 Score
Ridge Classifier (Statistical)	Sparse (TF-IDF)	One-vs-Rest (L2 Regularization)	0.95	0.48
LinearSVC (Statistical)	Sparse (TF-IDF)	One-vs-Rest (Linear Kernel)	0.96	0.55
Text CNN (Deep Learning)	Static (FastText)	3 Parallel Conv Layers (Kernels 3,4,5)	0.95	0.47
DistilBERT (Transformer)	Contextual Dense	Fine-tuned [CLS] Token Classification	**0.98**	**0.62**

```
--- Classification Report (using Optimized Thresholds) ---
               precision    recall  f1-score   support

        toxic       0.56      0.90      0.69      6087
 severe_toxic       0.31      0.66      0.42       367
      obscene       0.65      0.78      0.71      3688
       threat       0.47      0.70      0.56       211
       insult       0.66      0.75      0.70      3425
identity_hate       0.67      0.60      0.63       712

    micro avg       0.59      0.81      0.68     14490
    macro avg       0.55      0.73      0.62     14490
 weighted avg       0.60      0.81      0.68     14490
  samples avg       0.08      0.08      0.07     14490
```

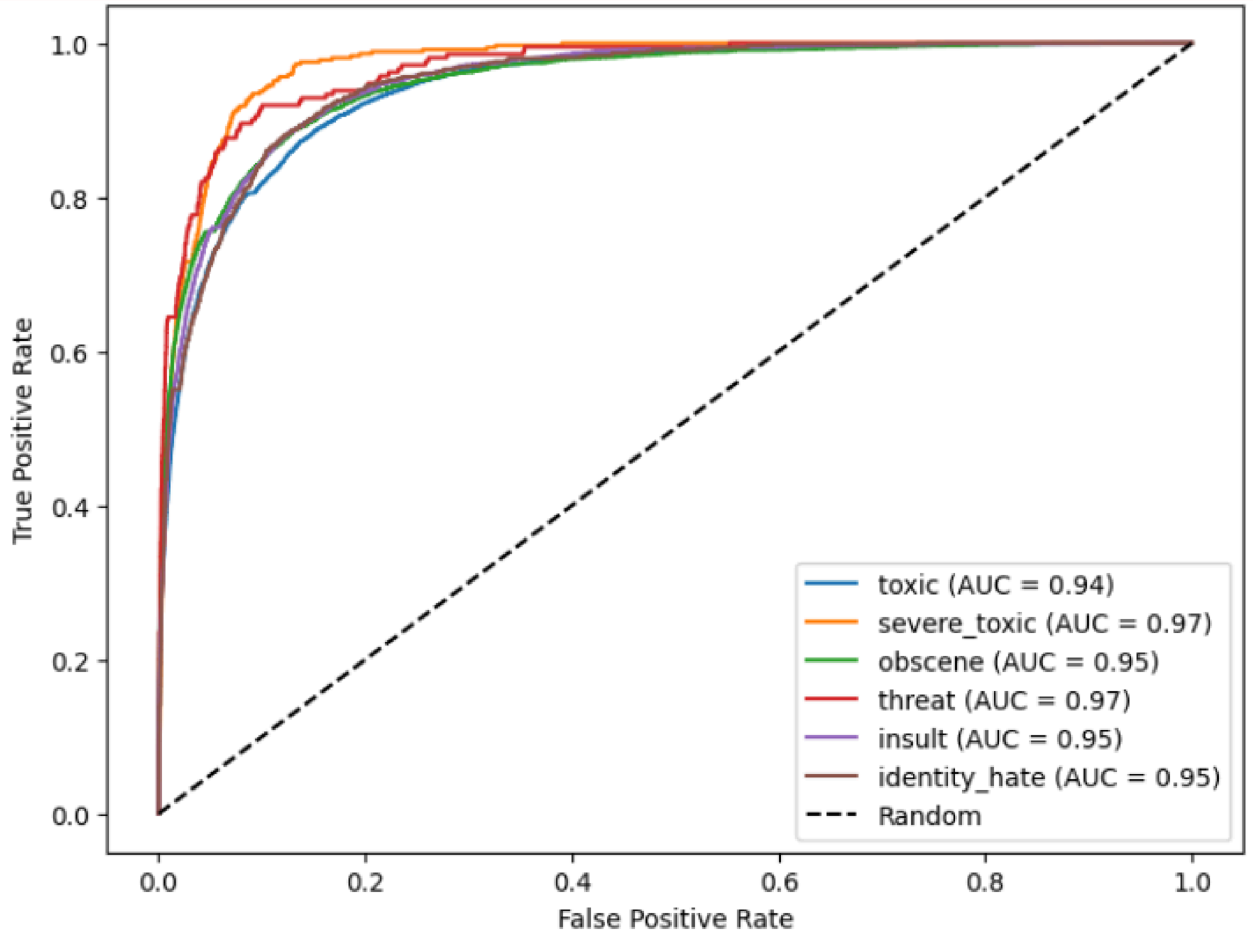
DistilBERT (State-of-the-Art)

- Neural models (Text CNN & DistilBERT) were optimized using Focal Loss to prioritize rare classes (like Threat & Identity Hate).

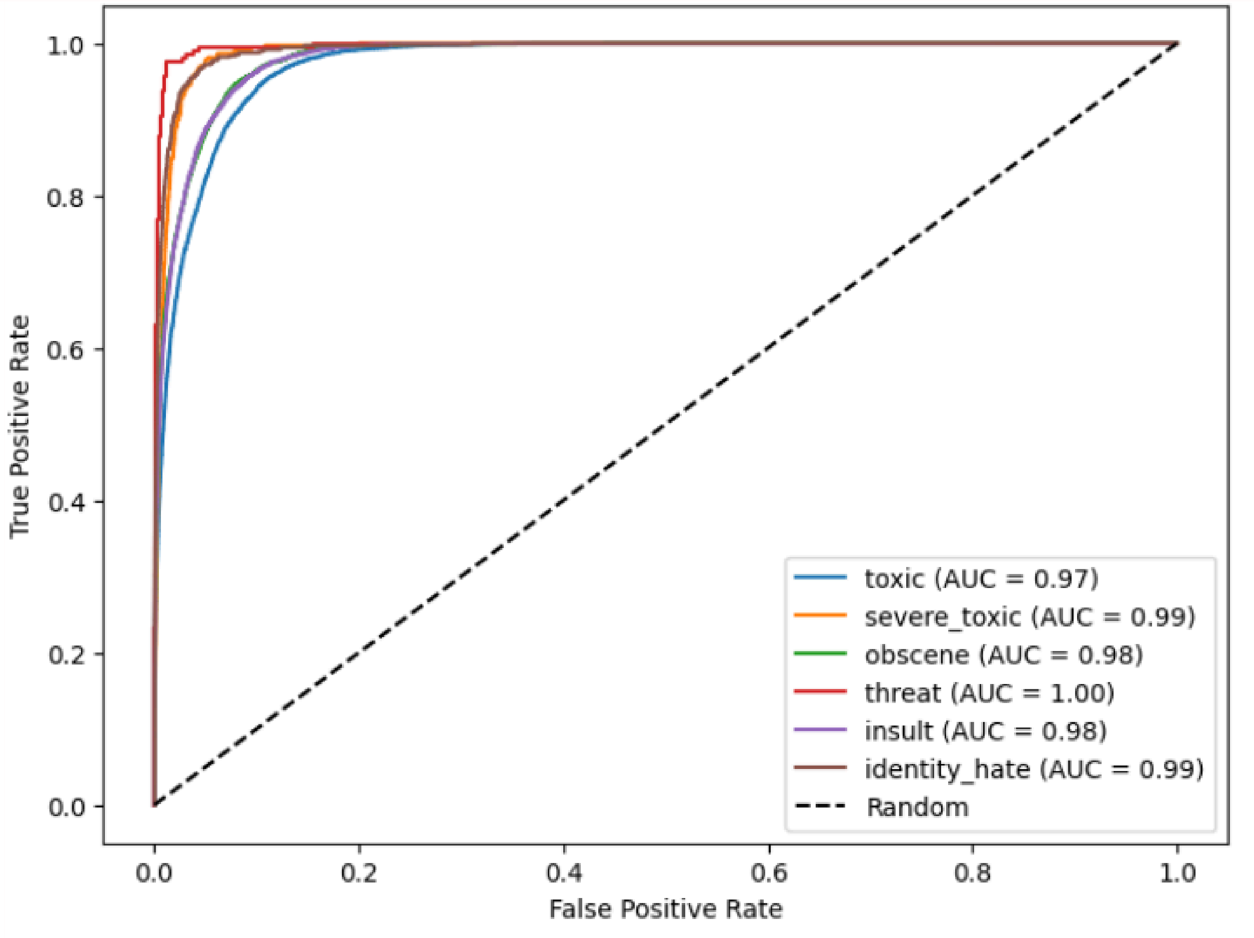
| ROC CURVE COMPARISON



LinearSVC (Statistical Baseline)



Text CNN (Deep Learning)



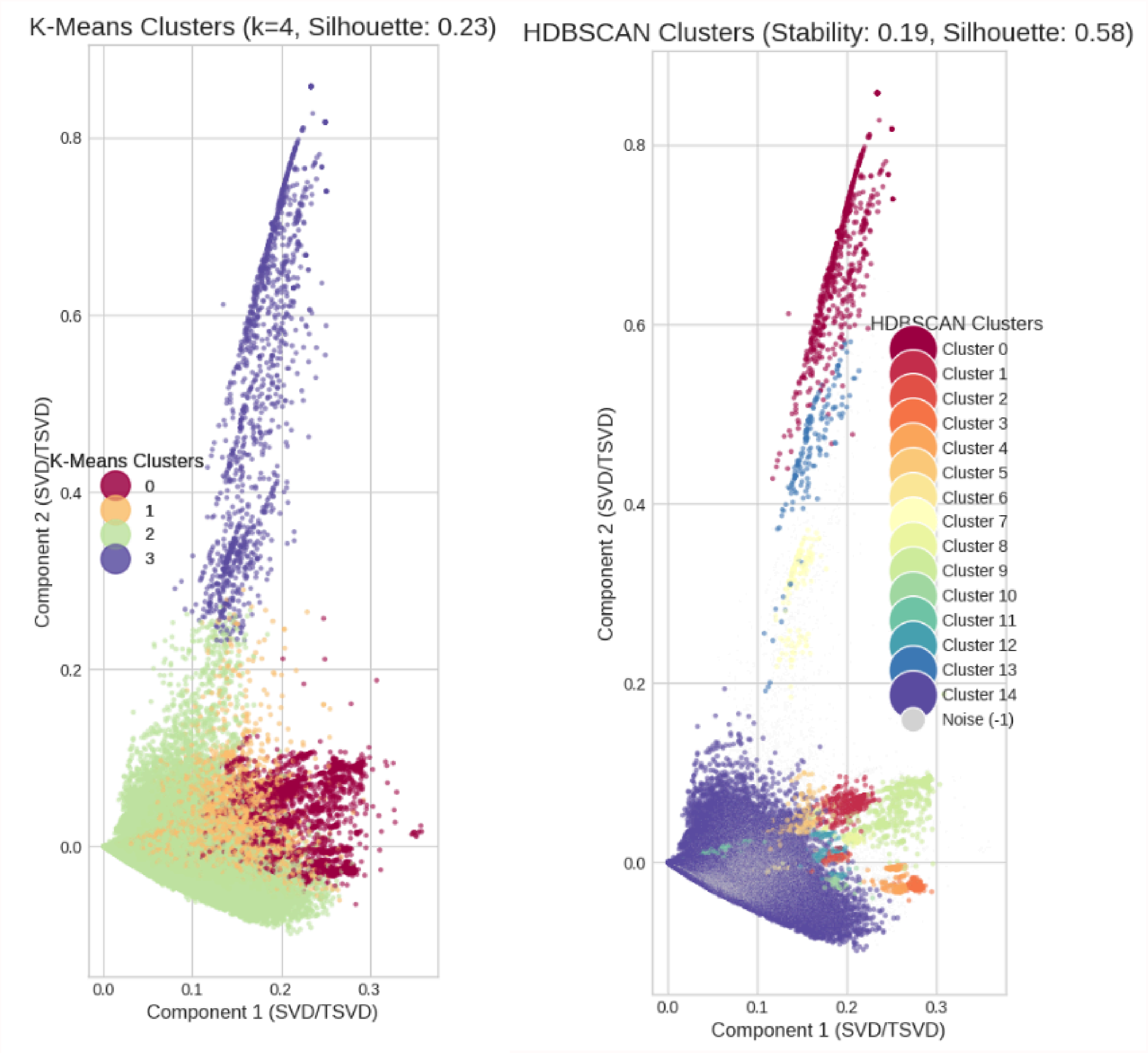
DistilBERT (State-of-the-Art)

| TEXT CLUSTERING PERFORMANCE

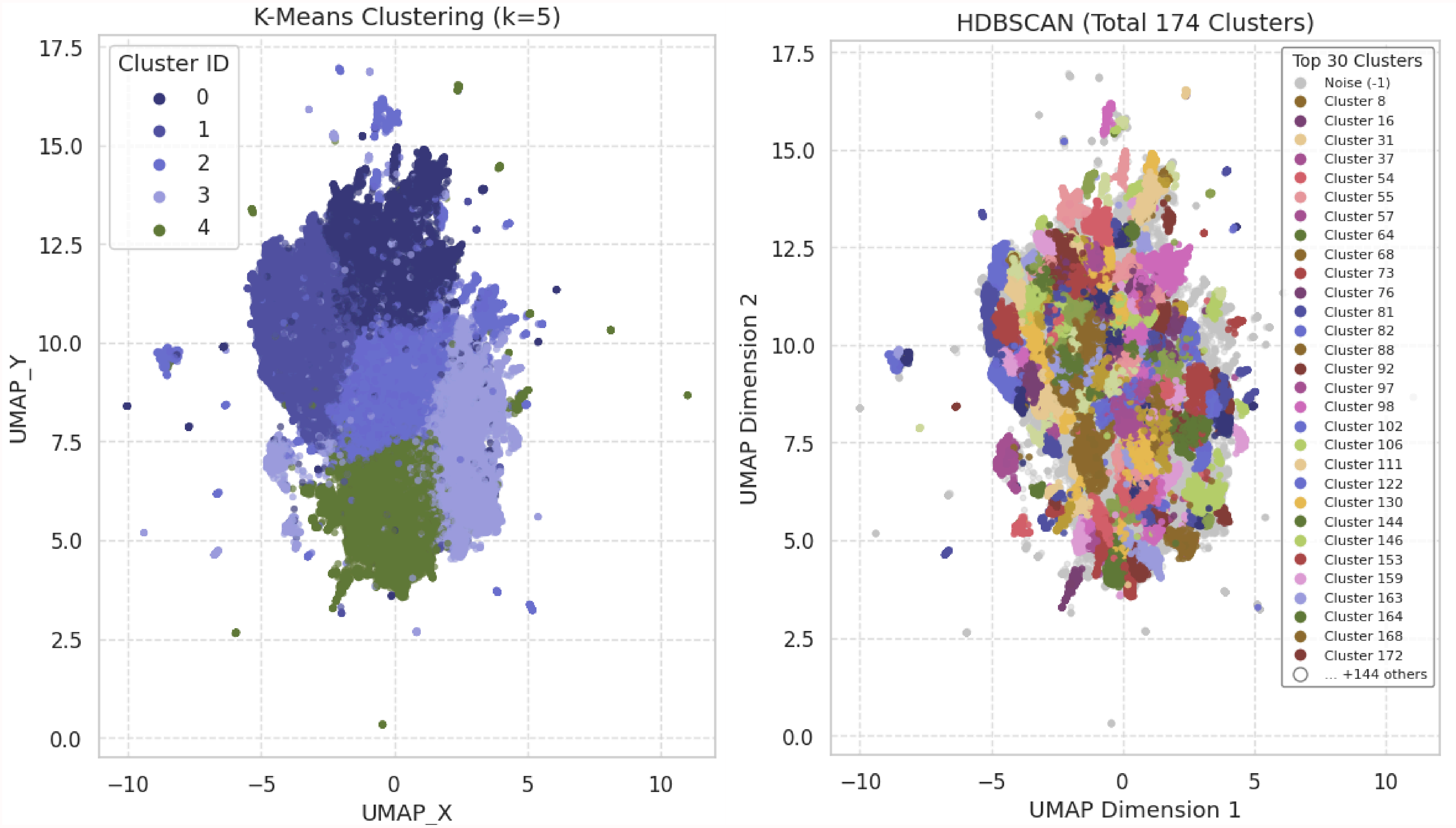
Feature Space	Algorithm	Silhouette Score	NMI Score	ARI Score	Cluster Purity
TF-IDF + TSVD (Statistical)	K-Means	0.2328	0.0145	-0.0665	90.48%
TF-IDF + TSVD (Statistical)	HDBSCAN	0.5826	0.0125	-0.049	90.56%
SBERT + UMAP (Semantic)	K-Means	0.2783	0.0276	0.0096	89.42%
SBERT + UMAP (Semantic)	HDBSCAN	0.5177	0.0376	0.001	**94.15%**

- HDBSCAN treated ambiguous data as 'Noise' (17% - 56%), significantly improving cluster coherence compared to K-Means.

| CLUSTER VISUALIZATION COMPARISON



K-MEANS VS HDBSCAN (TF-IDF)



K-MEANS VS HDBSCAN (SBERT)

| STATISTICAL CLUSTER PROFILING

K-means Cluster Profiling

Cluster	Sample Count	Top Keywords	Toxic Purity
2	134,284	article, wiki, page, one	1.12%
1	13,933	talk, page, redirect, user	0.18%
0	4,098	page, deletion, image	0.00%
3	1,386	blocked, vandalize, edit	0.00%

HDBSCAN Cluster Profiling

Cluster	Sample Count	Top Keywords	Toxic Purity
14	122,025	article, page, talk, would	0.93%
11	1,233	redirect, list, film, album	0.08%
0	673	vandalize, blocked, edit	0.00%
1	399	test, sandbox, welcome	0.00%

K-Means: Toxic vs. Non-Toxic

HDBSCAN: Top 2 Most Toxic Clusters

| SEMANTIC CLUSTER PROFILING

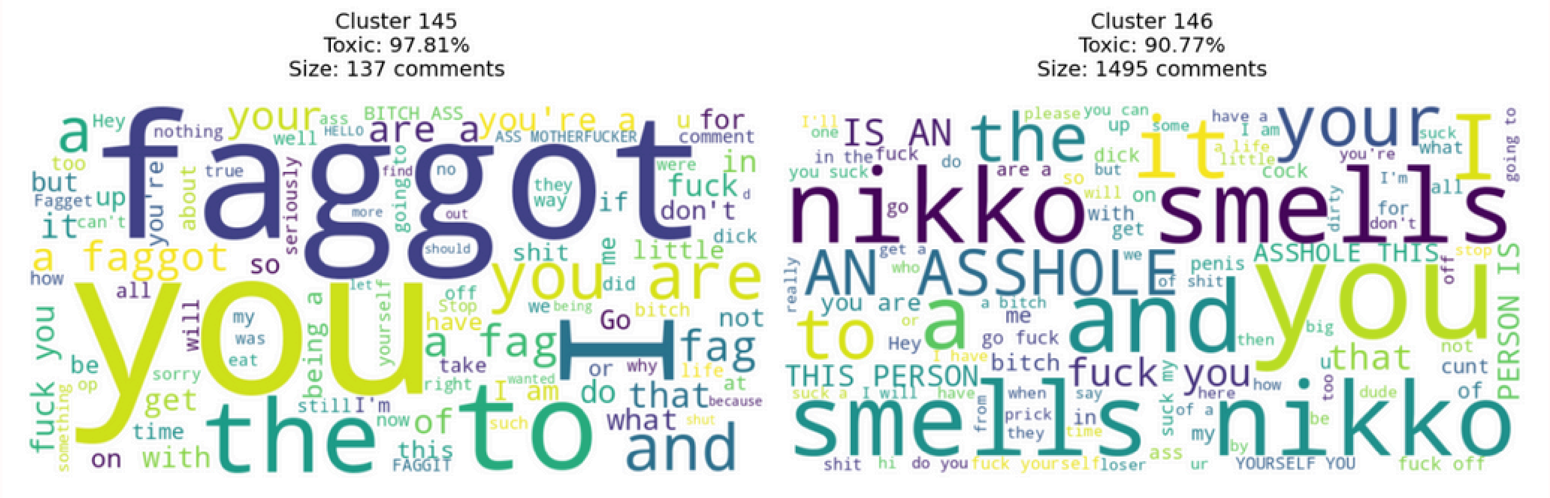
K-means Cluster Profiling

Cluster	Size	Toxic Purity	Top Keywords
0	28,480	22.95%	fuck, bitch, shit, stupid
4	31,823	14.63%	page, wikipedia, blocked, edit
3	31,446	7.61%	article, people, think, know
2	37,646	4.12%	article, wikipedia, page, deletion
1	29,935	3.59%	image, article, talk, page

HDBSCAN Cluster Profiling

Cluster	Size	Toxic Purity	Toxicity Specialization
145	137	97.81%	Targeted LGBTQ+ harassment
146	1,495	90.77%	Gender-based attacks
140	171	86.55%	Sexual orientation harassment
132	214	85.05%	Broad-spectrum abuse
80	309	41.10%	Extremist rhetoric

K-Means Cluster Topic



HDBSCAN: Top 2 High-Purity Clusters

| CONCLUSION

- **DistilBERT** achieved the highest performance (Macro F1: 0.62, Macro ROC: 0.98) because it understands the true meaning of words better than older models.
- **Focal Loss** effectively captured rare classes (Identity Hate, Threat) missed by standard classifiers.
- Our Clustering method (**SBERT + UMAP + HDBSCAN**) acted like a detective and found hidden hate groups with 94.15% purity.
- We recommend a **Hybrid System** that uses DistilBERT for fast filtering and Clustering to find new types of attacks.

THANK YOU :)

QUESTIONS?