Hate Speech Classification: Effectiveness–Efficiency Trade-offs on Twitter

Davide Marchi

Master's Degree Student, University of Pisa

Email: d.marchi5@studenti.unipi.it

TABLE I
LABEL DISTRIBUTION AND TOKEN LENGTH IN THE TRAINING DATA.

label	count	percent	tokens mean	tokens std
0	29720	92.985	12.517	5.644
1	2242	7.015	13.319	5.160

***Abstract*—We compare five text classifiers for hate speech detection on Twitter: a regex/keyword baseline, linear SVM with TF–IDF features, MiniLM sentence embeddings with logistic regression, an LSTM sequence model, and a zero-shot classifier based on DistilBERT fine-tuned on MNLI. Effectiveness is measured by macro-F1 and accuracy; efficiency is measured by estimated carbon emissions (kg CO_2e) for training and testing, alongside time. Pairwise McNemar tests assess statistical significance. A Pareto analysis of macro-F1 vs test emissions highlights practical trade-offs between effectiveness and environmental cost.**

I. Introduction

We study the effectiveness–efficiency trade-off in hate speech classification with five representative models, from keyword rules to pretrained transformers. We report macro-F1 and accuracy alongside estimated emissions, and test for significance with McNemar's test. Code and artifacts: github.com/davide-marchi/hate-speech-model-comparison.

II. Data

We use the Twitter hate speech dataset (training split) as provided on Kaggle [8]. Table I shows class distribution with token length statistics. We apply standard text cleaning (URLs, mentions, lowercasing).

III. Methods

We compare the following models:

- **Regex/keyword**: a lexical baseline built from indicative keywords and patterns; see early rule-based toxic language detection [1].
- **SVM + TF–IDF**: a linear SVM on word/character TF–IDF features, a strong classical baseline for text classification [2].
- **MiniLM + Logistic Regression**: sentence embeddings from `all-MiniLM-L6-v2` [3] followed by logistic regression.
- **LSTM**: a recurrent model over token sequences [4].
- **Zero-shot (DistilBERT-MNLI)**: using the pretrained model `distilbert-base-uncased-mnli` [5] for zero-shot classification via NLI.

We perform grid search over model-specific hyperparameters; performance is reported on the test set (see Table II).

IV. Experimental Setup

All runs are performed on CPU only for consistency across models. We track both *time* and *carbon emissions (kg CO_2e)* during training and testing. Emissions are estimated via a tracker implemented with the CodeCarbon library [7], which accounts for runtime and power draw; therefore, we emphasize emissions as an efficiency metric rather than wall-clock time alone [6].

V. Results

A. Effectiveness and Emissions

Table II reports test macro-F1 and accuracy alongside estimated test and train emissions. The SVM+TF–IDF model achieves the highest macro-F1 while keeping emissions low relative to neural alternatives. The keyword baseline is extremely efficient but lags in macro-F1. The zero-shot DistilBERT provides modest macro-F1 and comparatively higher emissions.

Figure 1 shows the Pareto frontier of macro-F1 vs test emissions. We prefer emissions-based comparisons over time-based ones, given emissions encode both runtime and CPU utilization under our CPU-only protocol.

B. Statistical Significance

We apply McNemar's test to paired predictions. Table III lists pairwise results; SVM+TF–IDF significantly outperforms the neural baselines at $\alpha = 0.05$ on this test set.

C. Error Analysis

We inspect confusion matrices and misclassified examples. Due to space, we include the SVM+TF–IDF confusion matrix in Figure 2; other model-specific reports are similar and are available in the artifacts.

TABLE II
EFFECTIVENESS AND EMISSIONS SUMMARY.

model	macro-F1	accuracy	test time (s)	train time (s)	test emissions (kg)	train emissions (kg)
SVM+TF–IDF	0.864	0.965	0.114	7.633	8.212 121e–7	2.566 488e–5
MiniLM+LR	0.728	0.888	12.070	274.935	2.229 687e–5	0.000 388
LSTM	0.717	0.886	0.500	800.165	1.335 282e–6	0.001 126
Regex/Keyword	0.705	0.936	0.075	1.398	7.687 961e–7	2.590 393e–6
Zero-shot	0.593	0.860	253.772	257.134	0.000 452	0.000 455

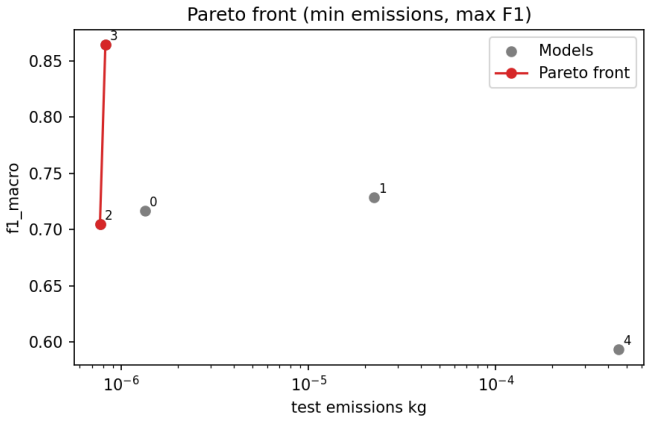

Fig. 1. Macro-F1 vs test emissions with a Pareto frontier.

TABLE III
MCNEMAR PAIRWISE TESTS (P-VALUE AND SIGNIFICANCE).

model_a	model_b	p_value	significant
SVM+TF–IDF	Zero-shot	2.356e–128	True
LSTM	SVM+TF–IDF	3.540e–92	True
MiniLM+LR	SVM+TF–IDF	1.687e–86	True
Regex/Keyword	Zero-shot	3.594e–66	True
LSTM	Regex/Keyword	6.524e–29	True
Regex/Keyword	SVM+TF–IDF	4.319e–28	True
MiniLM+LR	Regex/Keyword	3.763e–26	True
MiniLM+LR	Zero-shot	1.970e–7	True
LSTM	Zero-shot	1.150e–6	True
LSTM	MiniLM+LR	0.627	False

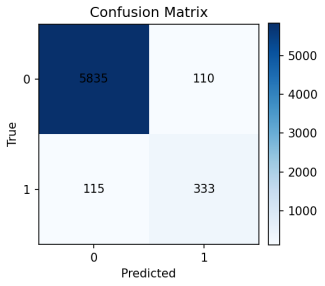

Fig. 2. Confusion matrix for SVM+TF–IDF on the test set.

VI. DISCUSSION

Our results suggest that strong linear baselines remain competitive on short social media text: SVM+TF–IDF provides the best macro-F1 at a favorable emissions footprint. Pretrained embeddings (MiniLM) and LSTM narrow the effectiveness gap but incur higher emissions per test example. Zero-shot classification offers convenience and label flexibility without task-specific training, but its emissions are comparatively higher under CPU-only inference. The keyword baseline is the most efficient but sacrifices recall.

Generalization to other hate speech datasets is plausible for relative rankings between simple linear baselines and larger neural models, though dataset shift (lexical, topical, and label schema) can alter absolute metrics. The emissions trends should persist under CPU-only execution; on GPUs, relative emissions may change depending on batch size and utilization.

VII. LIMITATIONS

Class imbalance and domain-specific language may bias metrics. Emissions are estimates (e.g., based on hardware utilization and regional energy intensity) and should be interpreted as relative comparisons rather than exact measurements. We restrict to small models and CPU to keep runs comparable; larger models or GPUs may change the Pareto frontier.

VIII. CONCLUSION

On this dataset, SVM+TF–IDF is a strong effectiveness–efficiency choice, dominating neural alternatives under CPU-only constraints. Emissions-aware evaluation provides a more informative perspective than time alone because it incorporates utilization. Code and full artifacts are available at github.com/davide-marchi/hate-speech-model-comparison.

REFERENCES

[1] E. Spertus, "Smokey: Automatic Recognition of Hostile Messages," in IAAI, 1997.

[2] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in ECML, 1998.

[3] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers," arXiv:2002.10957, 2020.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, 9(8):1735–1780, 1997.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv:1910.01108, 2019.

[6] A. Lacoste, A. Sasha Luccioni, V. Schmidt, and T. Dandres, "Quantifying the Carbon Emissions of Machine Learning," arXiv:1910.09700, 2019.

[7] CodeCarbon, “CodeCarbon: Track Carbon Emissions from Machine Learning,” available at: https://github.com/mlco2/codecarbon (accessed Nov. 2025).
[8] “Twitter Hate Speech,” Kaggle, available at: https://www.kaggle.com/vkrahul/twitter-hate-speech (accessed Nov. 2025).