PROJECT ABSTRACT

Hate Speech Detection using LSTM

Bidirectional LSTM text classifier (GloVe embeddings) trained on the Davidson et al. 2017 Twitter dataset (24,783 tweets) for hate speech vs offensive language vs neither — delivered as a built-to-order student project with code, notebook, report, PPT and viva kit.

01. Title

Hate Speech Detection using LSTM

02. Introduction / Background

Platforms face impossible arithmetic: millions of posts per day, human moderators able to review a fraction. Keyword blocklists fail exactly where it matters — context. Davidson et al.'s 2017 study made the distinction the field still uses: hate speech (hostility toward a protected group) is not the same as offensive language (profanity without group targeting), and a useful classifier must separate the two. Their 24,783-tweet labeled dataset — only 5.8% actual hate speech — is the standard teaching benchmark. Recurrent networks that read word order and context, rather than bags of keywords, are the natural modeling choice.

03. Problem Statement

Keyword filters both over-block (reclaimed language, “I hate Mondays”) and under-block (group-targeted hostility without profanity). The project addresses this by training a bidirectional LSTM on the Davidson dataset with class-weighted learning, evaluated on per-class F1 — the metric that exposes the “always predict offensive” shortcut raw accuracy would reward — as a moderation-assistance prototype that triages content for human review.

04. Objectives

- Build tweet-specific preprocessing: URL/mention stripping, hashtag splitting, elongation normalization, tokenization.
- Train a bidirectional LSTM (128 units/direction) over GloVe Twitter embeddings with dropout and a 3-way softmax, using class weights against the 5.8% hate-speech minority.
- Reach a design target of ~75%+ macro-F1, with the notebook logging per-class precision/recall/F1 and the confusion matrix on the held-out split.
- Build a demo UI that classifies typed or sample posts with word-level contribution highlighting.

- Deliver the complete student kit: source code, notebook, report, PPT and viva Q&A document.

05. Proposed Methodology

Tweets are cleaned, tokenized and mapped to indices, then to 100-dimensional GloVe Twitter embeddings. The bidirectional LSTM reads the sequence forward and backward, building a context-aware representation; dropout regularizes it and a dense softmax outputs hate/offensive/neither probabilities. Training uses class-weighted cross-entropy. Evaluation centers on per-class F1, macro-F1 and the confusion matrix, with dedicated error analysis of the hate-vs-offensive boundary — the dataset's hardest distinction.

06. System Architecture / Working

Raw tweet → cleaning/tokenization → index sequence → GloVe embedding lookup → BiLSTM (both directions) → dropout → dense softmax → three class probabilities. The training notebook logs loss and macro-F1 curves; the evaluation stage produces per-class reports and the confusion heatmap; the Flask demo serves the exported model with token highlighting computed from embedding-gradient contributions.

07. Hardware / Software Requirements

- Software: Python 3.10, TensorFlow/Keras, GloVe Twitter vectors (100-d), tweet preprocessing utilities, NumPy, pandas, scikit-learn, Matplotlib, Seaborn, Flask, Jupyter.
- Hardware: any modern laptop/desktop; the BiLSTM trains on CPU in minutes.
- Dataset: Davidson et al. 2017 labeled_data.csv (public) + GloVe Twitter embeddings (public).

08. Expected Outcomes

A trained 3-class tweet classifier with notebook-measured per-class F1 and macro-F1 on the held-out split (design target ~75%+ macro-F1); a confusion matrix showing the hate-vs-offensive boundary; documented failure cases (sarcasm, coded language); and a demo that classifies posts live with word-level explanations. Every metric comes from the student's own training run.

09. Applications

- Moderation-assistance triage queues (prototype scope — human review required).
- Teaching material for RNNs/LSTMs, embeddings and imbalanced classification.
- Reference pipeline for other short-text classification tasks (sentiment, toxicity).
- Baseline for the documented BERT fine-tuning extension experiment.

10. Future Scope

- Transformer fine-tuning (BERT/RoBERTa) comparison on the same split.
- Multilingual extension with language-specific labeled data.
- Character-level modeling for heavy slang and misspellings.
- Conversation-context modeling beyond single tweets.