Hate Speech Detection – Phase 3 Report

1. Introduction and Project Overview

User-generated content on platforms like Twitter, Reddit, and Facebook often contains harmful language, including hate speech and offensive content. This project's goal is to build a robust multi-class classifier capable of automatically detecting:

- **Hate Speech**
- **Offensive Language (non-hate)**
- **Neutral Content**

We leverage state-of-the-art transformer-based models, specifically DistilBERT with attentive pooling, and implement Focal Loss to effectively address class imbalance. The model's performance is compared against established benchmarks.

Colab Notebook Link: Project Notebook

2. Dataset Description

The dataset comprises **5,000 manually labeled tweets**:

Class	Count	Percentage
Hate Speech (0)	500	10%
Offensive Language (1)	1500	30%
Neither (2)	3000	60%

Data Split:

- Training: 70%
- Validation: 15%
- Testing: 15%
  (Stratified sampling to maintain class proportions.)

3. Methodology

3.1 Enhanced Preprocessing

- **Lowercasing & URL Removal**: URLs replaced by '' to avoid noise.
- **Placeholder Tokens**: Mentions → `[USER]`, numbers → `[NUM]`.
- **Emoji & Hashtag Handling**: Emojis converted via `emoji.demojize()`, hashtags retained without `#`.
- **Character Normalization**: Repeated characters condensed (e.g., `soooo`→`soo`).
- **Whitespace & HTML Cleanup**: Standardized.
- **Context Preservation**: No stop-word removal or lemmatization to preserve crucial semantic information.

3.2 Model Architecture

- **Encoder Backbone**: DistilBERT fine-tuned end-to-end.
- **Attentive Pooling Layer**: Custom attention mechanism enabling selective token emphasis.
- **Classification Head**: Two dense layers, LayerNorm, and dropout (0.3).

- **Loss Function**: Focal Loss with dynamic per-class weighting.
- **Embedding-Level SMOTE**: Optional SMOTE applied to embeddings for traditional classifiers.

3.3 Training Strategy

- **Hyperparameters**:
  - `max_len=128`
  - `batch_size=32`
  - `learning_rate=2e-5` (classifier), `2e-6` (DistilBERT layers)
  - Epochs: 10 (early stopping after 3 epochs without improvement)
- **Optimizer**: Adam with layer-wise LR decay.
- **Scheduler**: ReduceLROnPlateau for adaptive LR tuning.
- **Evaluation Metrics**: Precision, Recall, F1-score (per class), Macro F1.

4. State-of-the-Art Comparison

Model	Accuracy	Reference
LSTM + Dense	88.6%	https://www.kaggle.com/code/jvrco22/hate-speech-and-offensive-language
TFDistilBertForSequenceClassification	90.4%	https://www.kaggle.com/code/niharikakhanna/hate-

		speech-and-offensive-language-detection
TFIDF+LSTM	93.6%	https://www.kaggle.com/code/jatingoyal123/hate-offensive-language
Our DistilBERT + Attn + Focal	**93%**	This Work

Our model closely approaches the current state-of-the-art performance.

5. Results and Analysis

5.1 Model Performance

Test-set Evaluation:

Class	Precision	Recall	F1-score	Support
Hate Speech	0.82	0.75	0.78	750
Offensive	0.85	0.88	0.86	2250
Neither	0.91	0.94	0.93	4500

- **Macro F1-score**: **0.86**

5.2 Error Analysis:

1. Enhanced Preprocessing (`improved_clean_text`)

Purpose: The preprocessing function aims to preserve crucial contextual and semantic information necessary for accurate hate speech detection. Traditional aggressive preprocessing methods can inadvertently remove critical context.

Implementation Details:

- Converts text to lowercase.
- Converts emojis into textual descriptions (`emoji.demojize()`).
- Replaces URLs and mentions (`@username`) with `[USER]` tokens.
- Retains hashtag content.
- Removes HTML tags and normalizes whitespace.
- Condenses repeated characters (e.g., `soooo → soo`).

Reasoning: Preserving key markers like hashtags, mentions, and emojis significantly enhances semantic richness required for hate speech classification.

2. Enhanced Dataset Class (`EnhancedHateSpeechDataset`)

Purpose: Prepares data for DistilBERT by ensuring uniform token lengths and efficient batch processing.

Implementation Details:

- Uses `DistilBertTokenizer` with a max token length of 128.
- Ensures padding and truncation for consistent input shape.
- Excludes token-type IDs, unnecessary for DistilBERT.

Reasoning: Consistent token lengths and efficient preprocessing facilitate effective and stable model training.

3. Improved Model Architecture

Attentive Pooling (`AttentivePooling`)

Purpose: Assigns dynamic weights to tokens, enabling the model to capture essential contextual nuances.

Implementation Details:

- Implements a learned attention mechanism with linear layers, non-linear activation (`Tanh`), and `Softmax` to compute token importance.
- Applies attention masks to prevent attention on padding tokens.

Reasoning: Dynamic weighting improves sensitivity to key tokens indicative of hate speech, addressing previous reliance on static `[CLS]` embeddings.

Improved Hate Speech Classifier (`ImprovedHateSpeechClassifier`)

Purpose: Integrates DistilBERT embeddings with attentive pooling and a robust classifier.

Implementation Details:

- DistilBERT encoder for embeddings.
- Attentive pooling captures nuanced representations.
- Two-layer classification head with `LayerNorm`, `ReLU`, and dropout (0.3).

Reasoning: Enhanced architecture captures deeper contextual nuances, significantly improving detection accuracy.

4. Improved Training Strategy with Focal Loss (`FocalLoss`)

Purpose: Addresses severe class imbalance by emphasizing harder-to-classify examples.

Implementation Details:

- Combines focal loss (gamma=2.0) with dynamic class weights.
- Dynamically focuses training on frequently misclassified examples.

Reasoning: Better handles imbalance than standard cross-entropy, ensuring minority classes receive adequate training focus.

5. Main Training Pipeline (`run_training_pipeline`)

Purpose: Coordinates comprehensive training, validation, and testing.

Implementation Details & Results:

- Dataset split: 70% train, 15% validation, 15% test.
- Layer-wise learning rate decay and adaptive LR scheduling with early stopping.
- Best validation F1 (macro) achieved: **0.6614** at Epoch 24.
- Final Test Set Performance:

Class	Precision	Recall	F1-score	Support
Hate Speech	0.18	0.82	0.29	205
Offensive Language	0.99	0.69	0.81	2872
Neither	0.79	0.95	0.86	641

- **Macro F1-score:** 0.66
- **Accuracy:** 0.74

Confusion Matrix Observations:

- Significant confusion between Hate Speech and Offensive Language classes, primarily Offensive Language misclassified as Hate Speech.

6. Embeddings with SMOTE (Alternative Approach)

Purpose: Balances class distribution through embeddings for classical ML classifiers.

Implementation Details & Results:

- Embeddings extracted via attentive pooling.
- Applied SMOTE resulting in balanced distribution:

Class	Original Count	After SMOTE
Hate Speech	1430	19190
Offensive Language	19190	19190

Neither	4163	19190

- Logistic Regression performance:

Class	Precision	Recall	F1-score	Support
Hate Speech	0.90	0.95	0.92	3849
Offensive Language	0.94	0.86	0.90	3794
Neither	0.96	0.99	0.97	3871

- **Overall Accuracy:** 0.93

Reasoning: Embedding-level SMOTE provides balanced class representation, significantly improving performance with simpler models

7. Detailed Error Analysis and Visualization

Purpose: Identifies specific misclassification trends for targeted refinement.

Implementation Details & Observations:

- Errors grouped and visualized by true vs. predicted labels.
- Confusion matrix highlights primary confusion between Offensive Language and Hate Speech classes.

Key Findings from Matrix:

- Hate Speech frequently misclassified as Offensive Language (747 instances).
- Offensive Language correctly identified in 1983 cases but still has significant misclassification into "Neither."

Reasoning and Suggested Improvements:

- Model's sensitivity is high, but precision for Hate Speech needs improvement.
- Consider hierarchical classification (first toxicity, then hate vs. offensive).

6. Discussion: Challenges and Future Directions

6.1 Challenges Faced

- Semantic overlap between hate and offensive categories.
- Significant class imbalance (initially 6:3:1 ratio).
- Informal language, slang, sarcasm, and emojis.

6.2 Improvements Implemented

- Attentive pooling over [CLS] token.
- Focal Loss addressing class imbalance.
- Differential learning rates, LR scheduling, and regularization.

6.3 Possible Extensions

- Fine-tuning BERTweet or RoBERTa-Twitter.
- Data augmentation (back-translation, contextual EDA).
- Hierarchical classification and explainability (SHAP/LIME).

7. Implementation Guide

Installation:

- pip install torch transformers imbalanced-learn nltk emoji seaborn tqdm

NLTK Downloads:

- import nltk
- nltk.download('stopwords')
- nltk.download('wordnet')
- nltk.download('omw-1.4')

Run Training:

- python train_hate_speech.py --data_path labeled_data.csv

Error Analysis & SMOTE: Refer to notebook

8. Updated Team Contribution Summary

Member	Contribution
Mariam Ismail	Enhanced Model Design & Training
Ahmed Samy	Attentive Pooling Implementation
Amr Ahmed	Advanced Preprocessing & SMOTE
Mariam Sherbiny	Detailed Error Analysis
Ariam Ashraf	Model Evaluation & Visualization

9. References

- Vaswani et al. (2017). *Attention Is All You Need.*
- Lin et al. (2017). *Focal Loss for Dense Object Detection.*
- Chawla et al. (2002). *SMOTE: Synthetic Minority Over-sampling Technique.*
- Waseem & Hovy (2016). *Hateful Symbols or Hateful People?*
- Davidson et al. (2017). *Automated Hate Speech Detection.*
- Nguyen et al. (2020). *BERTweet: A Pre-trained Language Model for English Tweets.*