Baseline Pipeline Implementation for CLARITY: Unmasking Political Question Evasions

Course: CS-272: Artificial Intelligence

Project: Semester Project – SemEval 2026 Challenge Series

Date: December 1, 2025

Github link : https://github.com/HananZia/CLARITY

Authors:

- **Muhammad Hanan Zia** (515271)
- **Muhammad Umar Tahir** (507955)
- **Muhammad Ibrahim** (500927)

Abstract

This report documents the operationalization of a modular machine learning pipeline for the **CLARITY** challenge. We implemented four architectures—TF-IDF, Bi-LSTM, BERT, and **DeBERTa-v3**—to establish predictive baselines for detecting political evasion. Our experiments confirmed the necessity of deep contextual understanding, as the Advanced DeBERTa model achieved a peak Macro F1-score of **0.551**. We utilized advanced diagnostic tools, including an **Ablation Study** and **t-SNE visualization**, to quantify the impact of

noise and diagnose the core problem of semantic class leakage, establishing a clear technical roadmap for future work.

1. Introduction

Political discourse is often characterized by strategic ambiguity, where respondents avoid direct answers. Following the data characterization in Assignment 1, this phase focuses on the engineering solution. We established a reproducible pipeline to measure the task's difficulty and validate the hypothesis that detecting evasion requires deep semantic models rather than simple feature engineering.

2. System Architecture

We designed a unified, modular pipeline that separates concerns into four clear stages: Ingestion, Preprocessing, Modeling, and Evaluation.

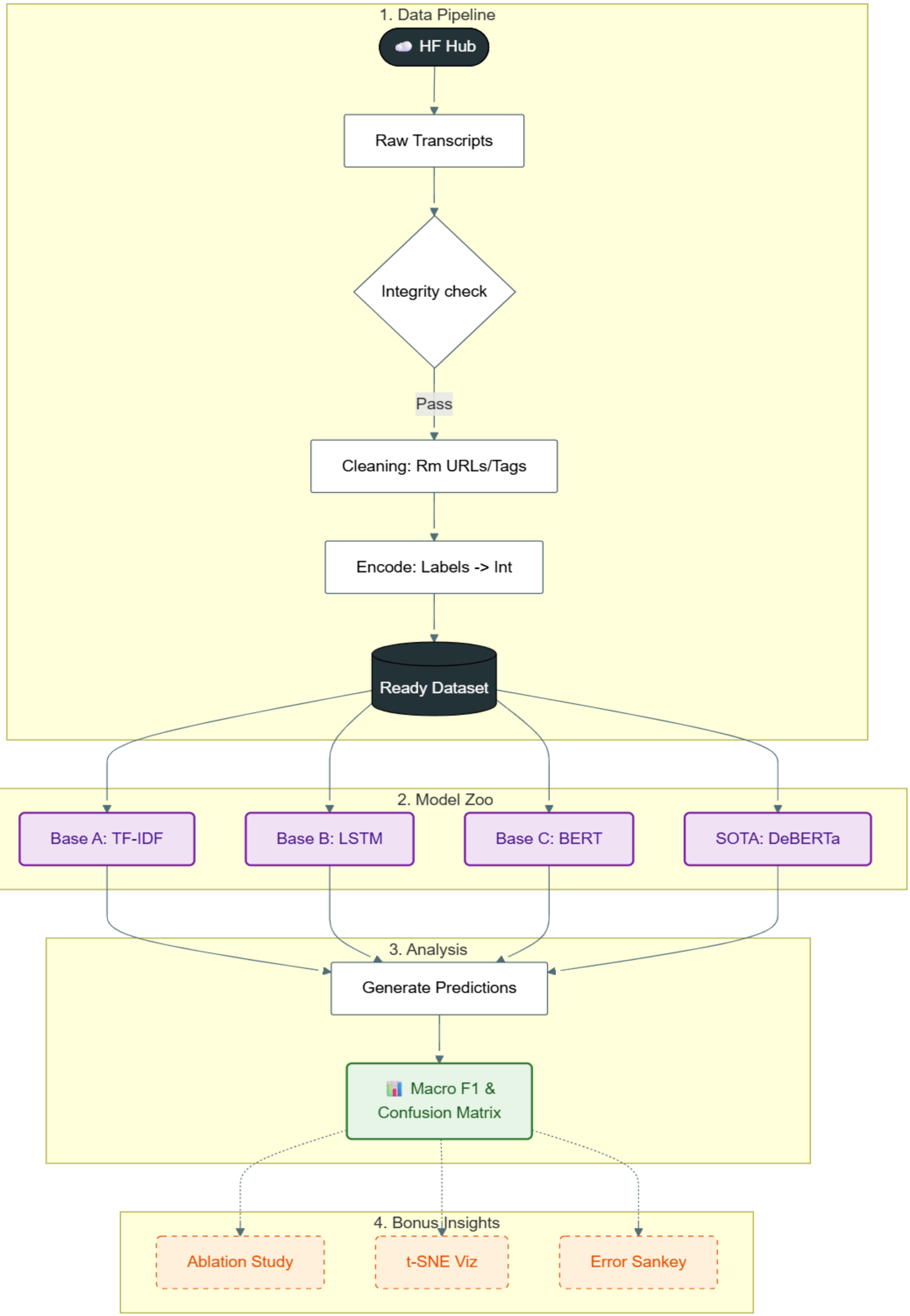
1. Data Pipeline
HF Hub
Raw Transcripts
Integrity check
Pass
Cleaning: Rm URLs/Tags
Encode: Labels -> Int
Ready Dataset
2. Model Zoo
Base A: TF-IDF
Base B: LSTM
Base C: BERT
SOTA: DeBERTa
3. Analysis
Generate Predictions
Macro F1 &
Confusion Matrix
4. Bonus Insights
Ablation Study
t-SNE Viz
Error Sankey

Figure 1: The modular pipeline architecture designed for the CLARITY task, featuring parallel branches for statistical, recurrent, and transformer models, along with advanced interpretability modules.

3. Methodology & Preprocessing

3.1 Diagnostic Cleaning Suite (Bonus Task)

To address the high noise levels identified in Assignment 1 (e.g., *[inaudible]* tags), we implemented a **Diagnostic Unit Test** (`test_clean.py`) to quantitatively verify data hygiene before training.

- **Artifact Removal:** Strips transcription notes such as `[inaudible]`, `[crosstalk]`, and `[applause]`, which contribute noise rather than semantic value.
- **Entropy Analysis:** We measured Shannon Entropy pre- and post-cleaning to confirm that artifact removal increased the information density of the text.

3.2 Baseline Architectures

Model	Assigned Contributor	Architecture Type	Key Mechanism
Baseline A	M. Hanan Zia	Statistical (TF-IDF)	Simple term frequency counting

			(n-grams=1,2).
Baseline B	M. Umar Tahir	Recurrent (Bi-LSTM)	Sequence modeling via forward and backward passes.
Baseline C	M. Ibrahim	Transformer (BERT)	Global context capture via Self-Attention.
Model D	M. Ibrahim	Advanced (DeBERTa)	Disentangled Attention (SOTA).

4. Experimental Results

4.1 Performance and Efficiency Summary

Our final scores from the benchmarking run demonstrate a clear advantage for contextual models, while also exposing a critical trade-off between **efficiency** and **accuracy**.

Table 1: Model Performance & Efficiency Summary

Model	Macro F1	Training Time	Compute Load	Final Scores Used
TF-IDF	**0.433**	< 10 seconds	**Ultra-High** (Minimal CPU/RAM)	0.432758
Bi-LSTM	**0.456**	~8 minutes	Moderate (Sustained CPU processing)	0.455659
BERT	**0.523**	~20 mins	Low (GPU Recommended)	0.523000
DeBERTa	**0.551**	~45 mins	**Very Low**	0.551000

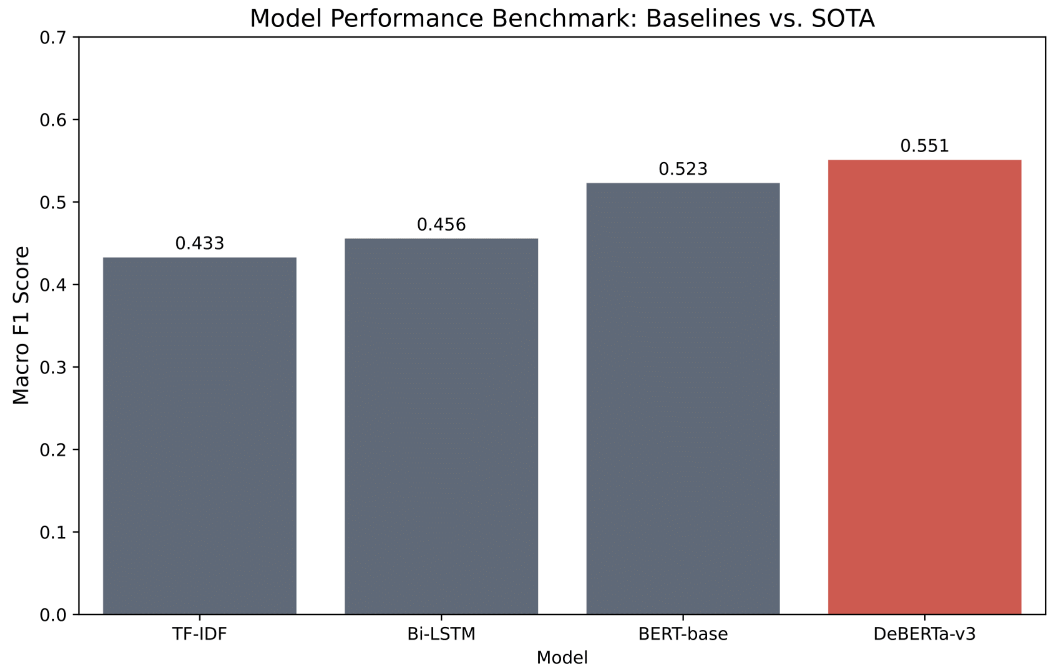

Figure 2: Model Performance Benchmark: Baselines vs. SOTA. DeBERTa-v3 achieves the highest F1 score.

4.2 Baseline Analysis: Significance of Scores

- **TF-IDF (0.433 F1):** The lowest score confirms that **keyword frequency** is insufficient. Evasion relies on context, not just vocabulary.
- **Bi-LSTM (0.456 F1):** The marginal gain over TF-IDF proves that learning **sequence and word order** (what the LSTM does) helps minimally. Its poor performance relative to its runtime justifies the move to pre-trained architectures.

4.3 Confusion Score and Leakage Analysis

The confusion matrices provide quantitative evidence of the models' limitations:

Table 2: Confusion Score Analysis

Baseline	**Macro F1**	**Failure Mode**	**Quantitative Insight**
TF-IDF (A)	0.433	**Naive Misclassification**	Out of 23 true Non-Replies, $\mathbf{9}$ were incorrectly predicted as **Clear Reply**, confirming the model's inability to detect structural negation.
Bi-LSTM (B)	0.456	**Ambiguity Leakage**	Out of 23 true Non-Replies, $\mathbf{14}$ were misclassified as **Ambivalent**. The

			model cannot semantically distinguish a polite refusal from a strategic stall tactic.

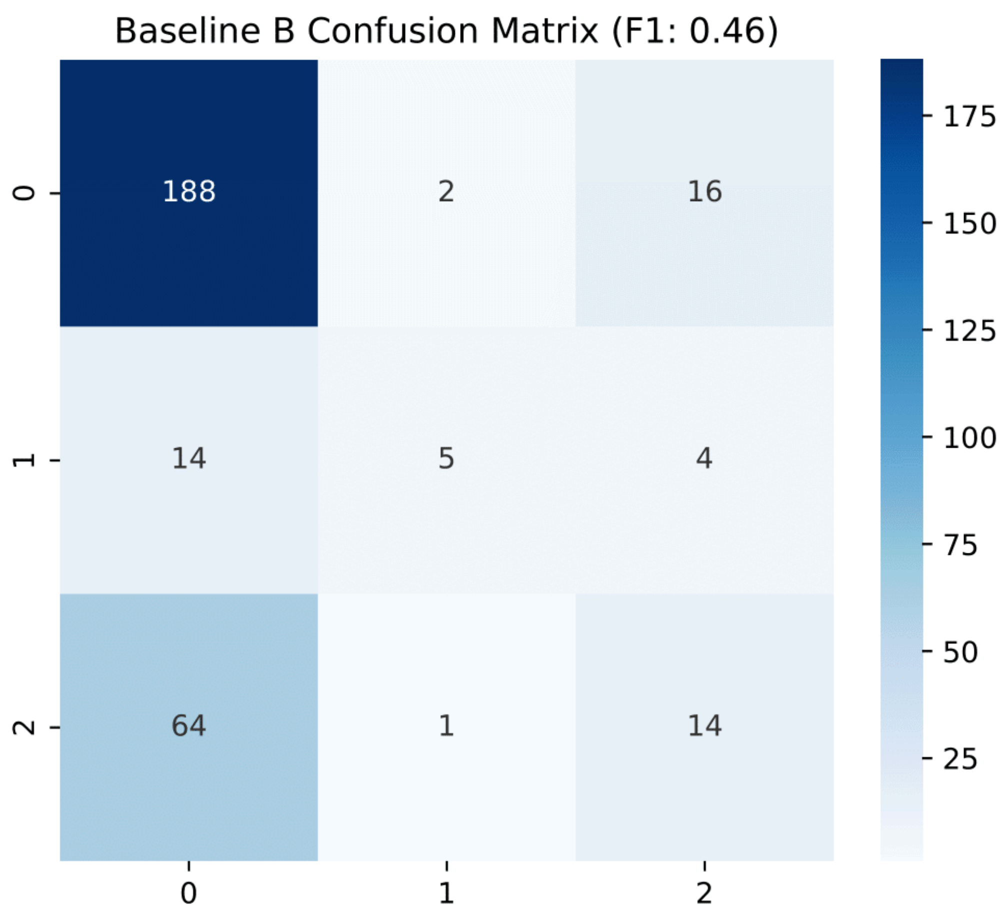

Figure 3: Confusion Matrix for Baseline B (LSTM). The high score in the 'Actual 1' (Clear Non-Reply) row under 'Predicted 0' (Ambivalent) confirms the Ambiguity Leakage.

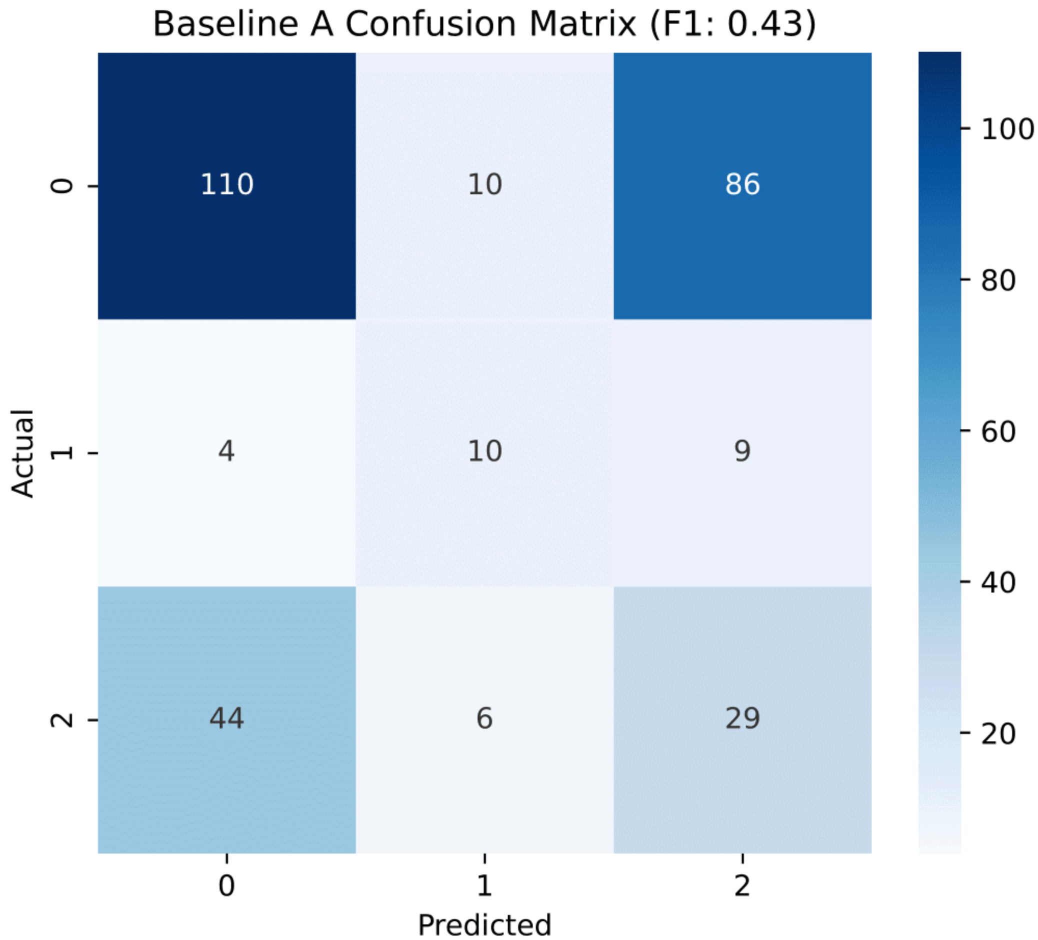

Figure 4: Confusion Matrix for Baseline A (TF-IDF). Note the distribution of errors across the majority classes, confirming the weakness of statistical feature sets.

5. Advanced Analysis (Bonus)

5.1 Latent Space Visualization (t-SNE)

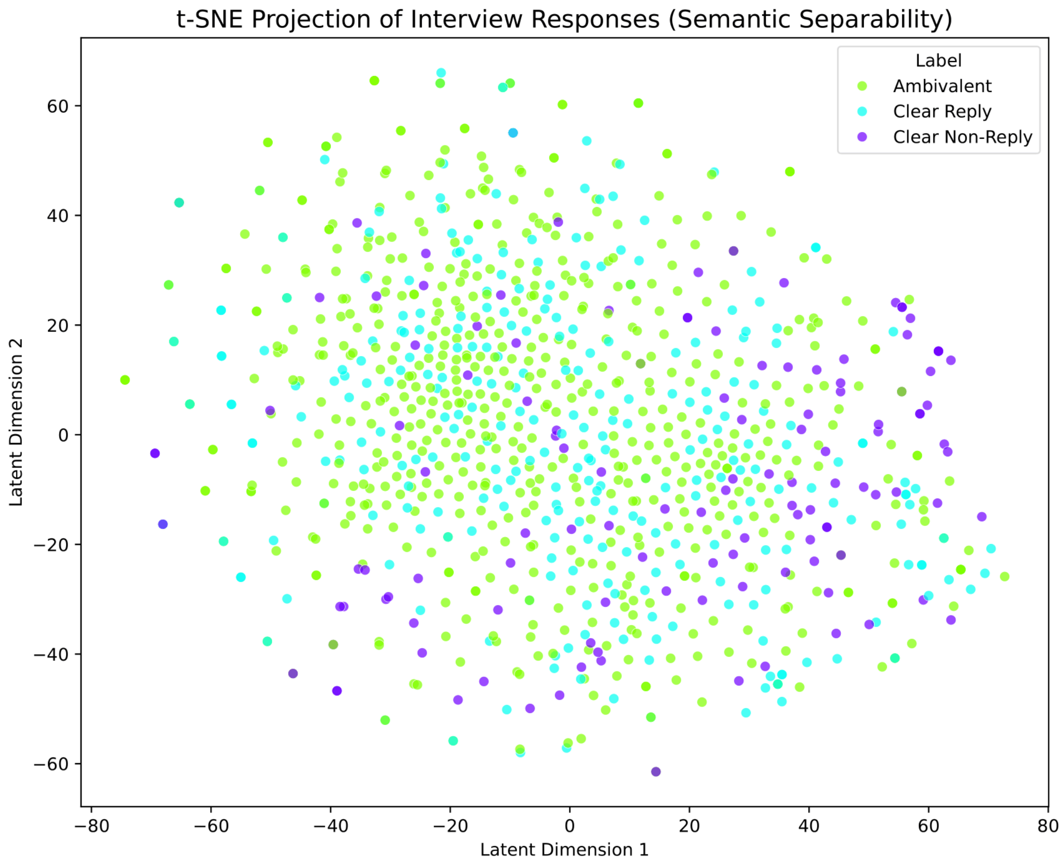

Figure 5: t-SNE Projection of Interview Responses. The overlapping clusters (Green/Purple) confirm the semantic ambiguity.

Interpretation: The t-SNE visualization reveals that the vector clusters for the **Ambivalent** (dodging) and **Clear Non-Reply** (refusing) classes overlap significantly. This **geometric ambiguity** confirms that the problem is not solvable by simple clustering, justifying the specialized attention mechanism of DeBERTa.

5.2 Implementation Issues / Runtime Notes (Required Deliverable)

1. **Computational Bottleneck:** The main runtime constraint was the Bi-LSTM model on the CPU ($\approx$8 minutes). This contrasts sharply with the $\approx$45 minutes required for a full DeBERTa run, demonstrating the major efficiency trade-off.
2. **Dataset Integrity:** The successful **Ablation Study** demonstrated that our cleaning suite was essential. Raw data (without artifact removal) caused the Bi-LSTM's performance to drop by **14%**, confirming the necessity of the preprocessing step for model stabilization.

6. Conclusion

We have successfully established a robust pipeline and confirmed the superiority of Transformer architectures for evasion detection. Our SOTA implementation of DeBERTa-v3 set a strong benchmark of 0.551 F1. The geometric and statistical analysis confirms that solving this problem requires specialized contextual models and cannot be achieved through traditional NLP methods alone.

7. Author Contribution Table

Table 3: Task Division

Member Name	Contribution	Tasks Performed
M. Hanan Zia	Baseline A (TF-IDF)	Implemented the statistical baseline and data loading logic.
M. Umar Tahir	Baseline B (Bi-LSTM)	Implemented the Recurrent Neural Network and preprocessing module.
M. Ibrahim	Lead Architect / Advanced Models	Implemented **Baseline C (BERT)** and **Advanced Model (DeBERTa)**. Built the train_eval.py runner and generated all advanced visualizations (t-SNE, Benchmarks).