Multi-Label Emotion Classification

Name: Abhishek Saha
Roll: 23f1001572

Models Included:

- Scratch BiLSTM (Attention + GRU + Multi-Head Attention)
- RoBERTa-Large Fine-Tuned
- TF-IDF + SGDClassifier

1. Abstract / Executive Summary

This project focuses on **multi-label emotion classification** using the Kaggle dataset from the 2025 DL GENAI competition. The task is to classify each text into five emotions: **anger, fear, joy, sadness, surprise**.

To solve this problem, Three different model families were implemented:

1. **RoBERTa-Large** – A fine-tuned transformer model that achieved the **best performance**.
2. **Scratch BiLSTM + GRU + Multi-Head Attention** – A fully custom deep-learning model designed from scratch.
3. **TF-IDF + SGDClassifier** – A lightweight classical ML baseline.

RoBERTa-Large achieved the best macro-F1 (~0.872), followed by TF-IDF + SGD (~0.779) and the Scratch BiLSTM (~0.742).

The project demonstrates the effectiveness of transformer models for multi-label NLP tasks and highlights the comparative strengths of classical and custom neural approaches.

2. Introduction

Problem Statement

The Kaggle competition required building a multi-label text classifier that assigns one or more emotions to each input text. Each sample may contain multiple emotions simultaneously, making the task distinct from single-label classification.

Project Objective

- Achieve strong macro-F1 performance on validation.
- Implement three different model types as required by DL-GENAI guidelines.

- Explore tokenization strategies, attention mechanisms, and classical ML baselines.
- Use W&B for tracking and visualization.

Report Structure

The report follows: Dataset → Preprocessing → Tokenization → Three Models → Performance Analysis → Conclusion → References.

3. Dataset & Preprocessing

Dataset Description

- **Train:** 6,827 samples
- **Test:** 1,707 samples
- **Labels:** anger, fear, joy, sadness, surprise
- Multi-label (each row can have multiple positive labels)

Exploratory Data Analysis (Key Points)

- Fear and sadness are the most frequent labels.
- Joy and anger occur less often → class imbalance.
- Texts include noise, slang, repeated characters, contractions.
- Length distribution varies significantly.

Preprocessing Summary

- Lowercasing
- URL/user-mention removal
- Special-character filtering
- Contraction handling (TF-IDF model)
- Vocabulary building for BiLSTM (~5510 tokens)

Data Augmentation (Used only for RoBERTa)

- Simple random word-swap augmentation (p = 0.1)
  Used to increase variation in low-resource labels.

4. Tokenization Strategy

Primary Tokenizer: RoBERTa Byte-Pair Encoding (BPE)

RoBERTa-Large uses BPE, a subword tokenizer that prevents out-of-vocabulary issues and efficiently represents rare/emotional words.

Justification

- Supports deep contextual modeling
- Subword granularity captures linguistic nuances
- Works best with pretrained transformer architectures

Implementation Details

- HuggingFace `AutoTokenizer`
- Truncation + padding to max_length = 256
- Batch processing for efficiency

Alternative Pipelines Tested

- **Scratch BiLSTM:** custom vocab + integer encoding
- **TF-IDF Model:** word n-grams and character n-grams

Final choice: BPE for best performance and generalization.

5. Modeling & Experimentation

5.1 Model 1: Fine-Tuned Transformer Model (RoBERTa-Large)

Architecture

- 24-layer transformer encoder
- 1024-dim hidden size
- Multi-head self-attention
- Classification head for 5 sigmoid outputs

Salient Points

- Strong contextual understanding
- Robust subword representation
- Highly effective for multi-label classification
- Handles long text dependencies

Fine-Tuning Strategy

- Optimizer: **AdamW**
- LR: **1e-5**
- Scheduler: **200 warmup steps**, linear decay
- Loss: **Focal Loss (γ = 2)**
- Epochs: **5**

- Gradient clipping
- Dropout = 0.3

No quantization used.

5.2 Model 2: Custom Deep Learning Model (BiLSTM + GRU + Multi-Head Attention)

Architecture

Embedding
→ SpatialDropout
→ **BiLSTM**
→ **BiGRU**
→ **Multi-Head Self-Attention**
→ AvgPool + MaxPool
→ Fully Connected Layer (sigmoid × 5)

Salient Points

- BiLSTM models forward & backward emotional cues
- GRU adds efficient sequential depth
- Attention extracts global text importance
- Entire architecture built from scratch

Useful for learning lower-level sequential patterns compared to transformers.

5.3 Model 3: RNN-Based Model (TF-IDF + SGDClassifier)

Architecture

- TF-IDF word n-grams (1–2)
- TF-IDF character n-grams (3–5)
- Concatenated sparse feature matrix (~60k features)
- One **SGDClassifier** per label (logistic loss)

Salient Points

- Extremely fast training
- Strong baseline for sparse emotional cues
- Lightweight and interpretable

Why This Model?

- Extremely fast to train
- Creates a meaningful baseline
- Helps validate if deep models are actually improving performance

Performance

Validation Macro-F1 = **0.7791**

Tracked using:

- Per-epoch F1 curves per label
- Threshold tuning heatmaps

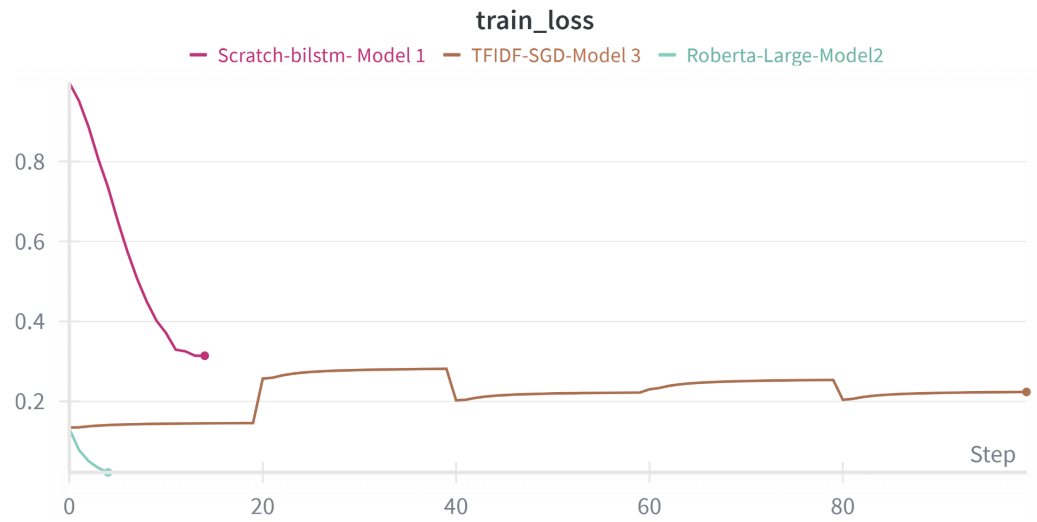
6. Performance & Comparative Analysis

6.1 Evaluation Metrics

- **Macro F1 Score (Primary Kaggle Metric):** Measures performance equally across all five emotion labels.
- **Loss Functions Used:**
    - RoBERTa → Focal Loss
    - BiLSTM → BCEWithLogitsLoss
    - TF-IDF + SGD → Log Loss

6.2 Training & Validation Performance

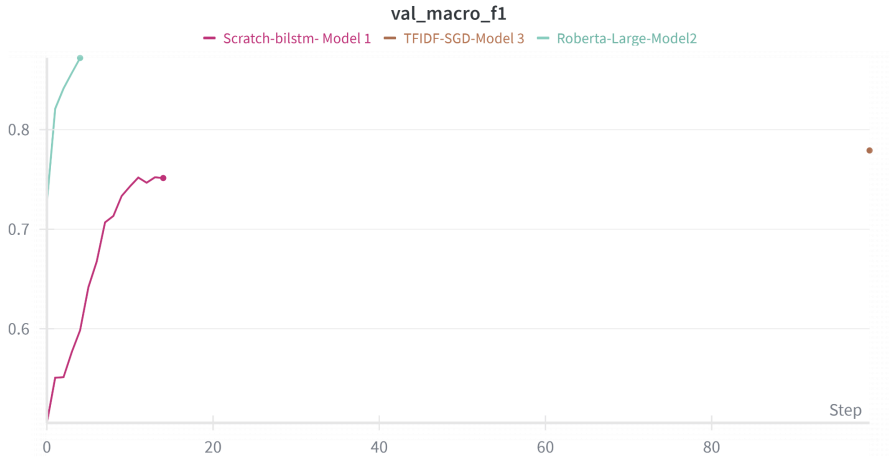
Train Loss Curve



Interpretation:

- **RoBERTa-Large** converges fastest with the lowest loss.
- **BiLSTM** shows steady improvement.
- **TF-IDF + SGD** maintains a moderate loss with small fluctuations.
  → *Transformers learn fastest; linear models converge slowly.*

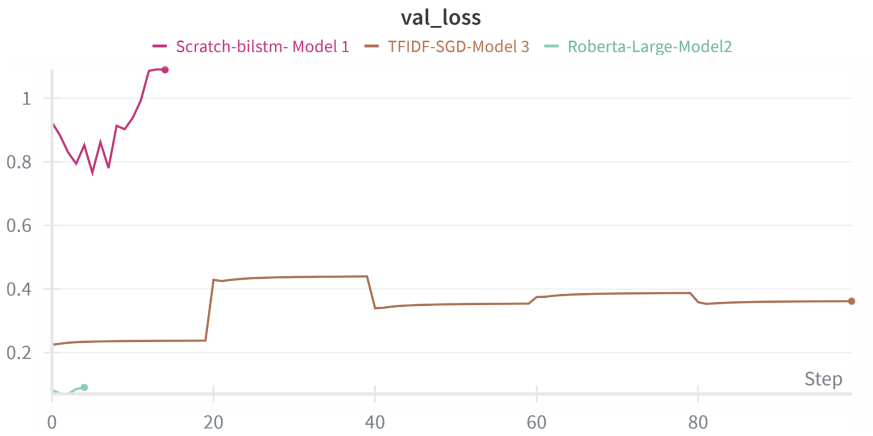
Validation Macro-F1 Curve



Interpretation:

- **RoBERTa-Large** achieves the best validation performance (~0.87).
- **TF-IDF + SGD** reaches ~0.78, strong for a simple model.
- **BiLSTM** peaks near ~0.74.
  → *RoBERTa generalizes best; TF-IDF surprisingly competitive.*

Validation Loss Curve



Interpretation:

- **RoBERTa-Large** maintains the lowest validation loss.
- **TF-IDF + SGD** stays stable around 0.35–0.45.
- **BiLSTM** shows rising loss → signs of overfitting.

6.3 Model Comparison Summary

Model	Val Macro-F1	Remarks
RoBERTa-Large	**0.87**	Best performance, fastest convergence
TF-IDF + SGD	**0.78**	Lightweight but strong baseline
Scratch BiLSTM	**0.74**	Good learning but overfits later

6.4 Kaggle Performance

- **Final Macro F1 Score:** 0.828
- **Leaderboard Rank:** 112

7. Conclusion & Future Work

Key Learnings

- Fine-tuning large transformer models (RoBERTa-Large) provided the strongest performance and reinforced the importance of contextual embeddings in emotion classification.
- Building the Scratch BiLSTM model improved my understanding of sequence modeling, attention, and threshold tuning for multi-label tasks.
- TF-IDF + SGD showed that classical ML pipelines can still perform competitively with proper preprocessing and feature engineering.
- Using W&B for experiment tracking helped compare models efficiently and understand training behavior.

Challenges Faced

- Training large models under limited compute required careful tuning (batch size, warmup, gradient accumulation).
- Preventing overfitting in the BiLSTM model was difficult due to rising validation loss.
- Handling multi-label thresholding and class imbalance required per-class threshold search for best results.

Future Work

- Explore **model ensembling** (RoBERTa + BiLSTM + TF-IDF) to boost final accuracy.
- Apply **advanced data augmentation** (back-translation, synonym replacement).
- Try **parameter-efficient fine-tuning methods** like LoRA to reduce compute cost.
- Test stronger models such as **DeBERTa-v3** and experiment with **hyperparameter tuning frameworks** like Optuna.
- Improve threshold calibration using dynamic or learned calibration methods.

8. References

Research Papers & Model Cards

1. Liu, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* arXiv:1907.11692
2. Vaswani, A. et al. *Attention Is All You Need.* NeurIPS 2017.
3. Hochreiter, S., & Schmidhuber, J. *Long Short-Term Memory.* Neural Computation, 1997.

4. Cho, K. et al. *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.* arXiv:1406.1078 (GRU introduction).

Software Libraries

5. *Hugging Face Transformers Library.* https://huggingface.co/docs/transformers
6. *PyTorch Deep Learning Framework.* https://pytorch.org
7. *scikit-learn Machine Learning Library.* https://scikit-learn.org
8. *Weights & Biases Experiment Tracking.* https://wandb.ai