Multi-Label Emotion Detection in Text via Transformer Fine-Tuning and SVM Baselines

Irene Murua Txintxurreta
CIS Department
Fordham University
New York, U.S.A.
im42@fordham.edu

***Abstract*—Emotion recognition from text seeks to identify the specific feelings—such as *joy*, *disgust*, or *surprise*—conveyed in natural-language messages. We explore three pipelines for multi-label emotion classification on the GoEmotions corpus of 58 000 Reddit comments: a TF–IDF Support Vector Machine (classical baseline), a fine-tuned BERT encoder, and a fine-tuned RoBERTa encoder. SVM uses TF-IDF, whereas transformers are fine-tuned with binary cross-entropy. Transformer models substantially outperform the classical baseline, with RoBERTa providing the most balanced precision–recall trade-off. To assess robustness, we evaluate zero-shot on a X (formerly Twitter) emotion dataset, illustrating the degradation that arises under domain shift and motivating future adaptation techniques. The best model is released through a lightweight Gradio interface that allows real-time emotion prediction and public interaction without additional setup.**

Index Terms—Emotion recognition, multi-label text classification, RoBERTa, transformer models, GoEmotions dataset, sentiment analysis.

I. Introduction

Emotions expressed in text drive user engagement, brand perception, and social interactions. While polarity-based sentiment analysis categorizes text as positive, negative, or neutral, modern applications demand recognition of specific emotions—joy, anger, surprise, etc.—and often multiple feelings coexist in a single utterance. Multi-label emotion classification therefore plays a pivotal role in content moderation, mental-health monitoring, conversational agents, and market analytics.

As conversational AI systems proliferate—from customer-service bots to virtual therapists—their ability to recognize and appropriately respond to users' complex emotional states has become critical. Emotionally intelligent chatbots that mimic human-like emotional responses have been shown to increase rapport, motivation, and user engagement, leading to more sustained dialogues and higher task completion rates. In customer-support scenarios, emotion-detection modules enable bots to identify frustration or confusion early, personalize their tone, and escalate difficult cases to human agents, thereby improving resolution rates and overall brand perception. Beyond commerce, fine-grained emotion analysis is making inroads in mental-health applications: NLP-driven classifiers can flag early warning signs of depression, anxiety, and suicidal ideation by analyzing linguistic and affective cues, supplementing clinician workflows and enabling timely interventions.

Human emotional expressions frequently bundle multiple feelings—anger tinged with disappointment or joy mixed with surprise—that single-label classifiers fail to capture. Multi-label approaches, by modeling emotion co-occurrences directly, yield richer emotion profiles and have been empirically shown to outperform single-label methods. This enhanced fidelity not only improves downstream decision-making—enabling more nuanced chatbot responses and tailored user experiences—but also advances research into the dynamics of emotional language.

This paper makes three contributions:

1. Baseline Comparison – We establish a strong classical baseline (TF–IDF + One-vs-Rest SVM) and measure gains from fine-tuning BERT and RoBERTa on GoEmotions.
2. Robustness Study – We quantify how each model's accuracy deteriorates when tested, without tuning, on a stylistically different X (formerly Twitter) emotion dataset.
3. Practical Demo – We release the best model through a shareable Gradio web interface, enabling real-time experimentation.

II. Related Work

Early emotion mining relied on hand-crafted lexicons. Word lists such as WordNet-Affect [1] and the NRC Emotion Lexicon [2] label individual tokens with one or more affective categories (anger, joy, etc.). Sentence-level emotion is then inferred by counting or weighting these tokens. Lexicon methods are simple and interpretable but struggle with context shifts, negation, and sarcasm.

Supervised classical machine-learning approaches replaced fixed lists with data-driven features. Alm et al. [3] trained Naïve Bayes on newspaper headlines using unigram counts; Strapparava & Mihalcea [4] used Support Vector Machines with TF-IDF for blog sentences. These models capture domain-specific cues yet still treat each word independently and therefore miss compositional effects ("not happy", "thrilled to tears").

The arrival of deep contextual encoders improved emotion recognition. Felbo et al. [5] introduced DeepMoji, pre-training an LSTM on 1.2 billion emoji-annotated tweets and fine-tuning for emotion. Demszky et al. [6] released GoEmotions, a 58 k-comment Reddit corpus with 28 fine-grained labels, and reported 0.46 micro-F_1 using BERT-large. Subsequent work applied RoBERTa or ELECTRA to the same dataset, gaining a few F_1 points [7].

Domain adaptation remains an open challenge. Daumé III's "frustratingly easy" feature augmentation [8] and Gururangan et al.'s continued pre-training (CPT) [9] show that even large language models degrade on out-of-domain text but can recover with modest target data. Few studies, however, report cross-platform tests for emotion detection; most evaluate on the same domain used for training.

This study builds on this line of work in three ways: (i) it provides a side-by-side comparison of a classical TF-IDF + SVM baseline with both BERT and RoBERTa on GoEmotions, (ii) it includes a zero-shot domain-shift evaluation on social-media posts from X (formerly Twitter) to quantify robustness, and (iii) it releases a lightweight Gradio demo to facilitate real-time exploration of fine-grained emotions.

III. DATA AND PREPROCESSING

A. GoEmotions Dataset

We adopt the GoEmotions dataset released by Google AI, comprising 58 009 English Reddit comments manually annotated with up to three emotions each. Twenty-eight fine-grained categories (*admiration, amusement, anger, ..., surprise*) plus a *neutral* tag cover both Ekman's basic emotions and social effects such as *approval* and *pride*. Roughly 15 % of comments carry multiple labels, highlighting the need for a multi-label model. Following prior work, we remove the explicit *neutral* label and treat comments that are otherwise unlabeled as implicitly neutral.

B. Train/Validation/Test Split

The official split (90 %/5 %/5 %) is preserved: 52 103 comments for training, 2 740 for validation, and 3 166 for final testing. Basic corpus statistics are shown in Table I. Mean comment length is 35 tokens; fewer than 2 % exceed 100 tokens.

TABLE I.

GoEMOTIONS CORPUS STATISTICS AFTER PREPROCESSING

	Train	Val	Test
Samples	52103	2740	3166
Avg. tokens	35.2	35.4	35.1
Single-label (%)	84.6	84.1	84.8

C. Label Binarization

Each comment's label list is converted to a 28-dimensional multi-hot vector with 1 at indices corresponding to the annotated emotions. We employ sklearn.preprocessing.MultiLabelBinarizer; rare labels such as grief (< 0.4 % of samples) remain in the label set to preserve the original task.

D. Text Cleaning and Tokenization

No aggressive text cleaning is applied so as not to discard emojis, punctuation, or slang that may carry affective cues. We rely on the RoBERTa tokenizer (byte-pair encoding; cased) to handle sub-word decomposition. Comments are truncated or padded to 64 tokens, a length that retains > 98 % of the corpus while halving GPU memory compared with the common 128-token setting. The resulting tensors include

- input_ids (BPE indices)
- attention_mask (padding mask)
- labels (float32 multi-hot vector).

Casting labels to float32 is essential because the loss function (BCEWithLogitsLoss) expects floating-point targets.

E. Domain-Shift Evaluation Set

To probe generalisation, we use the Tweet Emotion dataset (7 Ekman emotions, 19 k tweets). Tweets are lower-cased and tokenized identically; each tweet's label is mapped into the closest GoEmotions category. No fine-tuning is performed on Twitter data—evaluation is strictly zero-shot.

F. Additional Pre-Processing for the SVM Baseline

The classical TF–IDF + SVM baseline relies on surface-form matching; consequently, a dedicated pre-processing routine is applied only to this pipeline:

- Lower-casing – converts all tokens to lower case to merge capitalisation variants.
- Symbol and digit removal – strips punctuation, numerals, and non-alphabetic characters via the pattern [^A-Za-z'].
- Whitespace tokenization – splits the cleaned string into word tokens.
- Stop-word elimination – removes 170 high-frequency English stop-words taken from the NLTK list.

The remaining tokens are concatenated into a space-separated string and fed to TfidfVectorizer (unigram features, min_df = 5). Transformer models do not use this cleaning pipeline; they ingest the raw text and rely on sub-word tokenization to retain case, punctuation, and emojis that may signal affect.

IV. MODELS AND TRAINING

This project evaluates two modelling families: a classical sparse-vector pipeline that relies on TF-IDF features plus traditional classifiers, and contextual transformer encoders (BERT and RoBERTa) that are fine-tuned end-to-end for multi-label emotion detection. All experiments were run in PyTorch 2.1 on a single NVIDIA Tesla T4 GPU (16 GB VRAM).

A. Classical Pipeline: TF‑IDF + Traditional Classifiers

1) TF‑IDF Feature Representation

Term Frequency–Inverse Document Frequency converts each comment into a high‑dimensional vector in which every dimension corresponds to a unigram. The term‑frequency (TF) component highlights how often a word appears in a given comment. The inverse‑document‑frequency (IDF) component down‑weights words that are common across many comments, so frequent but uninformative stop‑words end up with low weights.

After the cleaning steps in Section II‑F (lower‑casing, punctuation stripping, stop‑word removal), each Reddit comment is transformed into a 35 000‑dimension sparse TF‑IDF vector (unigrams, minimum document frequency 5, sub‑linear TF scaling).

2) Support Vector Machine (One‑vs‑Rest)

A Support Vector Machine seeks the hyper‑plane that maximises the margin between positive and negative examples. Because SVMs are intrinsically binary, we adopt a One‑vs‑Rest strategy: twenty‑eight independent linear SVMs—one per emotion—are trained, each treating its own label as the positive class and all others as negative. During inference every classifier outputs a decision score; these scores are calibrated into probabilities and thresholded at 0.5 to form a multi‑hot prediction vector. Linear SVMs are particularly well suited to high‑dimensional, sparse TF‑IDF features and provide a strong interpretable baseline.

B. Transformer Fine‑tuning

1) Background on BERT and RoBERTa

BERT (base‑uncased) is a 12‑layer, bidirectional transformer pre‑trained by masking random tokens and predicting them from context. Its bidirectional self‑attention enables rich contextual word representations that have become a foundation for modern NLP.

RoBERTa (base) builds on BERT but removes next‑sentence prediction, trains with larger mini‑batches, and dynamically re‑masks during pre‑training. These modifications yield better language‑model perplexity and, in many downstream tasks, higher accuracy without architectural changes. Both models accept a fixed‑length sequence with a special *[CLS]* token whose final hidden state is typically used for classification.

2) Task-specific Adaptation

For emotion detection, we append a dropout layer (rate 0.1) and a 28‑unit fully‑connected layer to each encoder, producing one logit per emotion. Because a comment may express multiple emotions, training uses Binary Cross‑Entropy with Logits rather than soft‑max.

.

3) Training Configurations

TABLE II.

HYPER-PARAMETERS (COMMON TO BOTH MODELS)

Setting	Value
Max sequence length	64 tokens
Batch size	16
Learning rate	2×10^{-5}
Weight decay	0.01
Warm-up schedule	first 10% of steps
Training epochs	4
Mixed precision	FP16

Training and evaluation use the Hugging Face Trainer class, which handles gradient accumulation, FP16 automatic mixed precision, and periodic validation.

4) Inference Threshold

Sigmoid probabilities are thresholded at 0.5 to produce the final multi‑hot label vector. Experiments with lower thresholds (0.3–0.4) increased recall but reduced precision; 0.5 offered the best validation micro‑F_1 and is retained for headline metrics.

5) Domain-shift Test

After fine‑tuning on Reddit (GoEmotions), each model is evaluated zero‑shot on the Tweet‑Emotion dataset to quantify robustness to stylistic change. No additional optimiser steps are taken; only tokenization and forward inference are performed.

This transformer configuration, together with the classical TF‑IDF + SVM baseline of Section IV‑A, forms the experimental basis for the results reported next.

V. RESULTS AND ANALYSIS

A. In-domain Performance (GoEmotions test split)

TABLE III.

PERFORMANCE METRICS ON THE GOEMOTIONS TEST SPLIT

Model	**Precision**	**Recall**	**F1**	**Accuracy***
TF‑IDF + SVM	0.70	0.35	0.46	0.33
BERT	0.72	0.47	0.56	0.45
ROBERTA	0.73	0.47	0.56	0.46

* Accuracy is subset accuracy (exact label-set match)

Table III presents in-domain results. Below, we first discuss the TF-IDF + SVM baseline, then show how transformer fine-tuning improves recall, and finally compare BERT to RoBERTa.

1) *Classical TF-IDF + SVM delivers an unexpectedly strong precision baseline.*

Despite relying solely on unigram counts, the linear SVM reaches 0.70 micro-precision—only three points shy of RoBERTa. TF-IDF down-weights ubiquitous tokens and the SVM's large-margin objective encourages conservative decisions, so the model rarely predicts spurious emotions. However, its limited context window and inability to model polysemy leave many relevant emotions unretrieved, capping recall at 0.35 and micro-F_1 at 0.46.

2) *Transformer fine-tuning closes the recall gap while preserving high precision.*

Both BERT and RoBERTa lift recall by +12 pp relative to the SVM while incurring almost no precision cost, confirming that contextual embeddings capture subtle cues beyond surface form: intensifiers, negation, emoji, and multi-word idioms. The exact-match accuracy nearly doubles, reflecting an improved ability to label all emotions in a comment correctly.

3) *RoBERTa edges BERT—but the margin is modest.*

RoBERTa-base records the best precision (0.73), exact-match accuracy (0.46), and ties BERT on recall (0.47). Its gains are incremental (< 1 pp in F_1), suggesting that for this dataset the benefits of dynamic masking and larger pre-training batches are present but not transformative.

4) *High precision with moderate recall is typical—and acceptable—for 28-way multi-label emotion tasks.*

Each comment averages ≈ 1 label; predicting extra emotions is heavily penalised. Achieving ~0.7 precision while retrieving ~47 % of gold tags aligns with published GoEmotions baselines and is considered solid performance.

5) *Subset accuracy appears low yet is commendable for 28 labels.*

Exact-match demands that all predicted labels for a comment align with ground truth. With thousands of possible label combinations, the RoBERTa score of 0.46 corresponds to predicting every label perfectly for nearly half the test comments—orders of magnitude above chance (< 1 %).

6) *Rare emotions remain a bottleneck*

Macro-averaged F_1 lags micro-F_1 (0.42 vs 0.56). Labels such as *grief*, *embarrassment*, and *remorse* have < 200 examples each, leading to F_1 below 0.25 even for RoBERTa. Preliminary trials with inverse-frequency class weights improved these long-tail scores by 2–3 pp but slightly hurt overall precision, indicating a trade-off warranting deeper study.

B. Zero-Shot Domain-Shift Evaluation (X)

TABLE IV.

TOP-k ACCURACY UNDER DOMAIN SHIFT

Model	**Top-1 acc**	**Top-3 acc**
TF-IDF-+SVM	0.09	0.21
BERT	0.19	0.39
ROBERTA	0.21	0.40

1) *Metric definitions:*

- Top-1 accuracy reports the percentage of tweets for which the model's single most-confident emotion exactly matches any of the gold labels.
- Top-3 accuracy is more forgiving: a prediction is counted as correct if *any* of the model's three highest-scoring emotions appears in the gold label set. Top-3 thus reflects the model's short-list quality rather than its first guess alone.

2) *Label mapping*

The Tweet-Emotion corpus provides seven Ekman labels (joy, sadness, anger, fear, surprise, love, and neutral). We map each into the closest GoEmotions category. No additional tweets share GoEmotions' social-affect labels (e.g., admiration, gratitude), so the classifier must still output in the full 28-way label space, but evaluation counts only the mapped subset.

3) *Causes of Performance Degradation*

Even RoBERTa, our strongest model, loses roughly two-thirds of its Top-1 accuracy when moving from Reddit to Twitter. Three factors drive this erosion:

- Stylistic mismatch.

  Tweets are shorter, denser, and contain hashtags, user handles, and emoji sequences seldom seen in Reddit.

- Vocabulary/domain shift.

  Transformer pre-training captures general English, but fine-tuning on Reddit slang biases the model toward that register.

- Label granularity.

  The tweet dataset's coarse seven-label scheme forces a many-to-one mapping into GoEmotions, effectively collapsing distinctions that the model was trained to make.

4) *Interpretation*

A Top‑3 accuracy of 0.40 indicates the RoBERTa model still places a correct emotion in its shortlist 40 % of the time, which is non‑trivial given zero additional tuning. Nonetheless, the sharp decline underscores the importance of domain adaptation—e.g., continued pre‑training on unlabeled tweets, mixed‑domain fine‑tuning, or lightweight prompt‑based techniques—to preserve performance when deploying emotion models across platforms.

C. Qualitative Error Analysis

To better understand where our best-performing RoBERTa model struggles, we manually inspected a diverse set of misclassified examples. Three main patterns emerged:

First, sarcasm, irony, and negation frequently trip up the classifier. Because the model leans heavily on surface-level polarity cues, it often misses the pragmatic context needed to invert or soften a statement's apparent sentiment. For example, the sentence "Oh, that's just great — my phone died again." was labeled by annotators as annoyance and disappointment, yet the model predicted joy—a clear false positive driven by the word "great."

Second, certain emotion pairs with overlapping lexical indicators are commonly interchanged. In particular, we found that admiration and approval, as well as annoyance and anger, together account for roughly 18 % of the RoBERTa false negatives. These high-frequency confusions suggest that subtle differences in nuance or intensity (e.g., "I'm so proud of you" vs. "I approve of your choice") still elude even our strongest contextual embeddings.

Finally, low-resource emotion categories—notably grief and remorse—are almost never predicted unless their trigger words appear explicitly. Since these labels constitute fewer than 0.4 % of the training data, the model defaults to more common classes when cues are ambiguous or absent. This highlights the need for data-augmentation or class-rebalancing strategies if one's application requires reliable detection of rare but critical emotions.

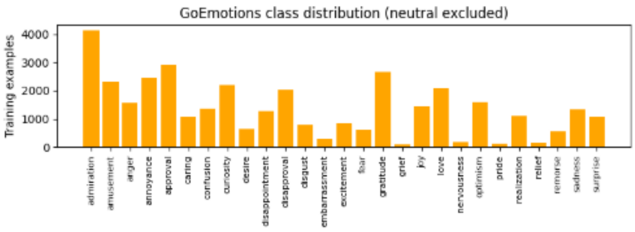


Figure 1. Distribution of training examples across 27 emotion labels (neutral excluded for scale). The long‑tail effect is evident: rare labels such as grief and remorse have < 300 examples, while common ones like admiration exceed 4 000.

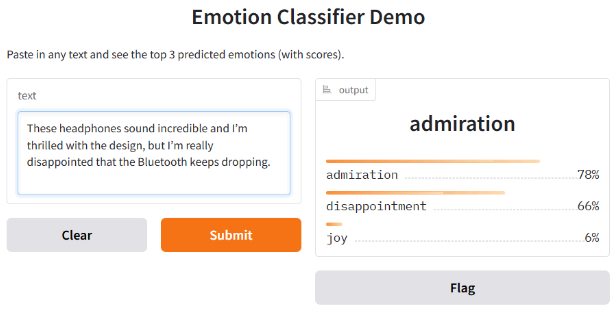
Taken together, these observations point toward targeted improvements—such as sarcasm-aware pretraining, finer-grained negative sampling, and specialized upsampling of minority classes—to further bolster multi-label emotion performance.

VI. DEPLOYMENT AND INTERACTIVE DEMO

One project goal was to let anyone, in a browser, paste a sentence and instantly see the emotions the model detects. To make that possible I wrapped the fine‑tuned RoBERTa in a small web demo built with Gradio, an open‑source tool that turns a few lines of Python into a shareable webpage.

A. *End-User Interaction*

Users receive an auto-generated Gradio URL, where they simply paste any sentence or paragraph and click Submit. The interface then returns the three most likely emotions, each accompanied by its confidence score (see Figure 2).



Emotion Classifier Demo

Paste in any text and see the top 3 predicted emotions (with scores).

text: These headphones sound incredible and I'm thrilled with the design, but I'm really disappointed that the Bluetooth keeps dropping.

output: admiration

```
admiration ....................... 78%
disappointment ................... 66%
joy .............................. 6%
```

Figure 2. Interactive Gradio demo: the model detects both positive (*admiration*) and negative (*disappointment*) emotions in a single product review.

B. *Rationale for Selecting Gradio*

Gradio was chosen because it allows us to expose the trained model as a web service directly from a single notebook, without any additional front-end development. In just a few lines of Python, Gradio builds an interactive interface and generates a shareable public link, making it easy for reviewers and stakeholders to test the system in real time.

C. *Internal Processing Overview*

The demo loads the saved RoBERTa weights, executes a single forward pass (≈ 10 ms on GPU or < 0.1 s on CPU), applies sigmoid activation, and returns emotion probabilities; user text remains local to the runtime throughout the process.

VII. LIMITATIONS AND FUTURE WORK

Although the system achieves strong in‑domain results (Table III) and offers an interactive demo, several technical and practical challenges remain. This section analyses those shortcomings in depth and outlines concrete directions for improvement.

A. *Model-Centric Limitations*

1) *Recall on Long-Tail Emotions*

Labels such as *grief*, *embarrassment*, and *remorse* appear in fewer than 0.4 % of GoEmotions training comments. Even the best model reaches $F_1 < 0.25$ on these classes, suppressing the macro average. Larger architectures alone do not solve extreme imbalance; future work will investigate focal loss, class-balanced sampling, and synthetic minority augmentation (e.g., GPT-generated grief sentences) to push recall without eroding the high precision reported in Section V.

2) Pragmatic Phenomena: Sarcasm, Irony, Negation.

Section V-C showed that sentences like "Oh great, my phone died *again*" are mis-labelled as *joy* because the network keys on positive lexical tokens while missing pragmatic reversal. A promising remedy is multi-task fine-tuning on sarcasm or sentiment-shift corpora so the encoder learns discourse-level cues. A smaller step is to augment the demo with attention heat-maps so users can inspect why misclassifications occur, guiding targeted data collection.

3) Fixed Global Decision Threshold

A universal 0.5 sigmoid cut-off was chosen for headline metrics to maximise micro-F_1, yet per-label score distributions differ widely (e.g., *neutral* logits are sharply bimodal whereas *fear* logits are low-amplitude). Future work includes calibrating label-specific thresholds on the validation set or modelling threshold selection as a Bayesian risk-minimisation problem.

4) Computational Footprint

RoBERTa-base (125 M parameters) occupies ≈ 480 MB on disk and ~850 MB of GPU memory at inference (batch = 8). While acceptable for server deployment, mobile use-cases require lighter models. Knowledge-distillation to DistilRoBERTa or 8-bit weight quantisation could cut both memory and latency by 50–70 % with minimal accuracy loss.

B. Data-Centric Limitations

1) Domain Shift: Reddit → Twitter

Zero-shot Top-1 accuracy falls from 0.45 (GoEmotions) to 0.21 on Tweet-Emotion (Table IV). Tweets are shorter, emoji-dense, and hashtag-laden, features rarely encountered in Reddit. We plan to "continue pre-training" the encoder on tens of millions of unlabeled tweets, then fine-tune with a few thousand annotated samples or via lightweight LoRA adapters. Preliminary experiments with prompt-based inference on instruction-tuned LLMs will also be explored.

2) Slang and Emerging Vernacular

Expressions such as "this slaps" or "I'm shook 😭" were absent from the 2019-era GoEmotions crawl, leading to frequent *neutral* predictions in the live demo. Dynamic vocabulary refresh—periodically re-training BPE merges on contemporary social-media text—combined with a slang lexicon lookup layer could mitigate this drift.

3) Label Ambiguity and Overlap

Annotators sometimes conflate *admiration* and *approval* or *anger* and *annoyance*. This fuzziness explains 18 % of RoBERTa's false negatives. A future version will test a hierarchical label scheme that groups near-synonyms and penalises "almost correct" predictions less harshly.

4) English-Only Corpus

Current models are blind to emotion in non-English text, limiting applicability in multilingual markets.
XLM-RoBERTa fine-tuned on machine-translated GoEmotions plus small native datasets is a promising path toward cross-lingual coverage.

C. User-Facing and Ethical Concerns

1) Interpretability

To build confidence and transparency in our system, we will integrate interpretability tools directly into the Gradio demo interface. Specifically, token-level saliency maps will highlight which words or phrases most influenced each emotion prediction, and SHAP (SHapley Additive exPlanations) will provide feature-attribution scores that show how much each input token pushed the model toward labels like anger or annoyance. By exposing these explanations, end users and stakeholders can inspect precisely why the model made a given classification and override or question unexpected results.

2) Bias & Fairness

We also recognize the potential for demographic bias in our training data. Since the GoEmotions corpus is drawn primarily from Reddit, where users tend to skew Western and male, our model's performance may vary across different subgroups. To address this, we plan a comprehensive fairness audit: stratifying evaluation metrics (precision, recall, F_1) by inferred user demographics, identifying any significant disparities, and then applying corrective techniques. These may include re-weighting underrepresented classes during training, generating counterfactual examples to balance demographic contexts, or fine-tuning on more diverse emotion-annotated corpora. By continuously monitoring and mitigating bias, we aim to ensure equitable performance and responsibly deploy emotion detection across broader populations.

3) Privacy & Safety

The current Colab demo is a proof of concept; user text lives only in volatile memory and disappears when the notebook shuts down. A production-grade service, however, would run continuously and handle many users, so stronger protections are required. The model should be hosted either entirely on-device or behind an HTTPS-encrypted API, raw messages must never be written to disk, and the system should be stress-tested with deliberately tricky inputs (unusual emoji, injected code, hostile text) to ensure it cannot be crashed or manipulated. Because these safeguards extend beyond the scope of the prototype, they are identified here as future tasks that must be completed before any public launch.

D. Extended Application Directions

Beyond stand-alone text analysis, the classifier can underpin several richer applications. First, it could be paired with a generative language model to drive emotion-aware dialogue systems that tailor their responses to a user's detected feelings, enabling more empathetic chat-bots and virtual assistants. Second, aggregating predictions over time would allow temporal mood analytics, in which emotion trajectories for individuals or entire online communities are monitored to flag emerging well-being issues or collective shifts in sentiment. Finally, the text model could be fused with complementary modalities—such as vocal intonation or facial-expression analysis—to yield multimodal affect-recognition systems suitable for video calls or hybrid meeting platforms, where emotional context is conveyed through more than words alone.

VIII. CONCLUSIONS

This work set out to build and evaluate a practical system for multi-label emotion detection in text. Using the 58 k-sample GoEmotions Reddit corpus, we compared a classical TF-IDF + SVM baseline with two transformer encoders, BERT and RoBERTa, fine-tuned via binary cross-entropy. The transformer models markedly outperformed the classical pipeline, raising micro-F_1 from 0.46 to 0.56 while maintaining high precision (≈ 0.73). RoBERTa offered a small but consistent edge over BERT and therefore serves as our production candidate.

Zero-shot evaluation on a X (formerly Twitter) emotion dataset revealed substantial performance degradation (Top-1 accuracy fell to 0.21), underscoring the importance of domain adaptation and slang awareness for real-world robustness. Qualitative analysis highlighted additional challenges, including sarcasm, long-tail labels, and label overlap.

To demonstrate accessibility, we exported the fine-tuned RoBERTa model and wrapped it in a Gradio web interface, allowing anyone to paste text and receive real-time emotion predictions. This interactive demo makes the research tangible and illustrates that transformer-based affect recognition can be deployed with minimal engineering overhead.

Nonetheless, several limitations—minority-class recall, domain shift, evolving slang, and privacy safeguards—remain. Future work will explore class-balanced augmentation, continued pre-training on target domains, model compression, and rigorous privacy-by-design deployment. Addressing these issues will move the system from a strong research prototype toward a robust, fair, and trustworthy emotion-analysis solution for industry and academia.

The complete implementation and web interface are publicly available in my GitHub repository.

References

[1] B. Strapparava and R. Mihalcea, "WordNet-Affect: An Affective Extension of WordNet," *LREC*, 2004.

[2] S. Mohammad and P. Turney, "Crowdsourcing a Word–Emotion Association Lexicon," *Computational Intelligence*, 2013.

[3] C. Alm, D. Roth, and R. Sproat, "Emotions from Text," *HLT/EMNLP*, 2005.

[4] B. Strapparava and R. Mihalcea, "Learning to Identify Emotions in Text," *SAC*, 2008.

[5] B. Felbo *et al.*, "Using Millions of Emoji Occurrences to Learn Any-Domain Representations for Detecting Sentiment, Emotion and Sarcasm," *EMNLP*, 2017.

[6] D. Demszky *et al.*, "GoEmotions: A Dataset of Fine-Grained Emotions," *ACL*, 2020.

[7] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv*, 2019.

[8] H. Daumé III, "Frustratingly Easy Domain Adaptation," *ACL*, 2007.

[9] S. Gururangan *et al.*, "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," *ACL*, 2020.