# Extended Literature Review: Lightweight Emotion Detection from Text

2025-09-29 04:41 UTC

## 1. Task & “Lightweight” Scope

Emotion detection refers to identifying emotional categories from short texts. It is typically framed as a multi-label classification problem. The GoEmotions dataset (Demszky et al., 2020) is the dominant benchmark with 27 fine-grained emotion classes + neutral. By “lightweight,” we refer to models that reduce: • Parameter count (e.g. <70M) • Inference latency (e.g. <15ms per batch) • Memory footprint This is particularly relevant for real-time applications (chatbots, mobile apps, on-device ML).

## 2. Datasets

| Name | Description |
|---|---|
| GoEmotions | Reddit-based, 58k examples, 27 emotions + neutral (raw) or 6 + neutral (simplified) |
| dair-ai Emotion | Twitter dataset with 6 basic emotions |
| EmoContext | SemEval 2019 dataset for dialog-based emotion classification (3-turn chat data, 3 class |

## 3. Lightweight Models Considered

| Model | Key Property |
|---|---|
| DistilBERT | 40% fewer parameters, ~60% faster than BERT |
| ALBERT | Parameter sharing + factorized embeddings |
| ModernBERT | Optimized encoders (e.g., TinyBERT, MiniLM, MobileBERT) (e.g., TinyBERT, MiniLM, M |
| Distilled GoEmotions Checkpoints | Pre-trained student models available on Hugging Face |

## 4. Techniques for Lightweighting

**Knowledge Distillation (KD)**
Transfer knowledge from a large “teacher” model (e.g., BERT-base) to a small “student” model (e.g., DistilBERT). Improves performance without increasing size.
*Recent paper:* Hussain et al., 2025 — Optimized knowledge distillation for emotion detection in social media.

**Quantization**
Reduces weight precision (e.g., from float32 → int8). Useful for hardware speedup with minimal performance drop.
*Recent paper:* Zhou et al., 2024 — Efficient emotion recognition via expansion quantization.

## 5. Evaluation Best Practices

• Report micro, macro, and weighted F1 scores to cover both frequent and rare emotion classes.
• Report latency using forward-pass timing (e.g., using `time` in PyTorch) for realistic batch sizes (e.g., 32 examples).
• Include model size (trainable parameters) and optionally memory footprint.
• Threshold tuning (e.g., sweeping 0.2–0.7) is often necessary to improve macro-F1 due to class imbalance.

• Evaluate both the raw (27-class) and simplified (6-class) schema to understand trade-offs between granularity and performance.

## 6. Recommended Setup

| Component | Details |
|---|---|
| Baseline Models | DistilBERT, ALBERT, ModernBERT |
| Dataset | GoEmotions (raw + simplified), optionally EmoContext |
| Experiments | Train each model; run distillation & quantization variants |
| Metrics | F1 scores, latency, model size, label-wise performance |

## 7. Related Work & Broader Context

Recent studies in NLP increasingly focus on balancing model efficiency and classification accuracy — especially for emotion detection tasks in constrained environments (e.g., chatbots, mobile UIs, multilingual assistants). While this review focuses on text-only models, some recent work also extends into multimodal and cross-lingual emotion detection, where lightweight models remain a critical need due to deployment constraints on edge devices or low-resource languages.