

LLM vs Fine-Tuning

The Real Cost/Accuracy Tradeoff in Short-Text Emotion Classification

An empirical comparison of classical, encoder, and generative-LLM approaches, measured on accuracy, latency, and dollar cost.

Sirac Simsek · June 2026

Headline Result

Table 0. Headline accuracy, latency, and cost across the four approaches.

Approach	Macro-F1	p95 Latency	\$ per 1,000 predictions
DistilBERT (fine-tuned)	0.9128 ± 0.0023	45.9 ms	≈ 0 (local)
TF-IDF + Linear SVM	0.8348	0.24 ms	≈ 0 (local)
Claude Haiku 4.5, few-shot (k=6)	0.5127	network-bound	\$1.27
Claude Haiku 4.5, zero-shot	0.5147	network-bound	~\$0.40

On a label-rich classification task, a 67M-parameter fine-tuned encoder dominates a frontier generative LLM on every axis that matters: accuracy by roughly 40 macro-F1 points, latency by orders of magnitude, and cost by infinity to one. Even the classical SVM beats the LLM by 32 macro-F1 points. The lesson is not that LLMs are bad. The lesson is that off-the-shelf LLMs are the wrong default when labels exist.

Abstract

Generative large language models have become the default tool for many text-understanding problems. This paper asks a more disciplined question: on a well-scoped, label-rich classification task, does prompting a frontier LLM actually outperform a small fine-tuned model, and what does it cost?

We compare four approaches to six-class emotion classification on a corpus of approximately 417,000 short informal texts: (i) a classical TF-IDF and Linear SVM baseline, (ii) a fine-tuned DistilBERT encoder, (iii) a generative LLM (Claude Haiku 4.5) used zero- and few-shot via constrained structured output, and (iv) a parameter-efficient QLoRA fine-tune of a small open LLM, whose implementation is complete and whose evaluation is reserved as future work. All approaches are measured not only on macro-averaged F1 but on the axes that drive production decisions, namely inference latency and dollar cost per prediction, using a reproducible and tested measurement harness.

The result is counterintuitive but robust. The fine-tuned DistilBERT reaches 0.9128 ± 0.0023 macro-F1. The classical SVM reaches 0.8348. The generative LLM, despite being far larger and more

expensive, manages only approximately 0.51 macro-F1 zero- or few-shot, at roughly \$1.27 per 1,000 predictions against effectively zero for the local models, and at roughly 190× the latency of the SVM. In-context examples did not help.

We argue that on tasks with abundant labels and idiosyncratic, dataset-specific label conventions, a trained model dominates a prompted LLM on accuracy and cost. The LLM's value lies in the cold-start regime, not the label-rich one. We accompany the empirical study with a discussion of structured-output reliability (the model emitted off-vocabulary labels on roughly 1% of calls) and a practical decision framework.

1. Introduction

Emotion classification from text is a core natural language processing problem with applications in social-media monitoring, product analytics, and human-computer interaction. Unlike coarse sentiment analysis, it requires distinguishing between multiple affective states (anger, fear, joy, love, sadness, surprise) that overlap semantically and frequently lack explicit lexical markers, especially in short, informal text.

The original version of this work compared a classical bag-of-words baseline against a fine-tuned transformer and concluded, as the literature broadly does, that contextualised representations win. Since then, the default practitioner instinct has shifted. Faced with a text task, many reach first for a general-purpose generative LLM and a prompt. This paper interrogates that instinct directly.

We frame the central question as an engineering decision rather than a leaderboard contest:

Given labelled data and a fixed classification task, when is fine-tuning a small model worth it versus prompting a large generative model, and what is the real cost of each choice?

To answer it credibly we make three methodological commitments. We extend the comparison to a generative LLM contender evaluated through constrained structured output. We measure latency and dollar cost as first-class metrics rather than asserting them. And we build the entire pipeline as reproducible, tested software so the numbers can be regenerated rather than trusted on faith. The contribution is therefore as much methodological and engineering-oriented as it is empirical.

Key contributions

1. **A four-way empirical comparison** spanning classical features, encoder fine-tuning, prompted LLMs, and parameter-efficient LLM fine-tuning, on the same dataset, evaluated identically.
2. **Production-grade measurement of latency and cost** alongside accuracy, including a token-accounted dollar cost per 1,000 predictions for the LLM, rather than asserted estimates.

3. **A rigorous structured-output methodology** that exposes the gap between schema constraints and decoding guarantees, with a measured off-vocabulary leakage rate of approximately 1%.
4. **A reproducible engineering harness** (92 unit tests, deterministic seeding, response caching, rate limiting, CI on Python 3.11 and 3.12, FastAPI inference service, Docker image) that makes the comparison auditable rather than anecdotal.

2. Dataset and Task Definition

The dataset consists of short English texts annotated with one of six discrete emotion labels (the CARER emotion corpus; Saravia et al., 2018), widely redistributed on Kaggle. Texts are informal and resemble social-media posts, with subjective language, ambiguous phrasing, and varying emotional intensity.

- **Size:** 416,809 samples
- **Task:** single-label, multi-class classification (exactly one emotion per text)
- **Split:** 80% train, 10% validation, 10% test, stratified by label to preserve class proportions and avoid leakage
- **Class imbalance:** strongly non-uniform (Table 1), which motivates macro-F1 as the headline metric

Table 1. Class distribution.

Emotion	Label ID	Share
joy	1	33.8%
sadness	0	29.1%
anger	3	13.7%
fear	4	11.4%
love	2	8.3%
surprise	5	3.6%

The 9:1 ratio between the most and least frequent classes means a model can post high accuracy while performing poorly on minority emotions. This is the central evaluation hazard the methodology is designed to expose.

3. Preprocessing

Different model families benefit from different preprocessing, so the project implements two deliberately distinct pipelines (Table 2), applied per model.

Table 2. Preprocessing pipelines.

Step	Classical (classic_clean)	Transformer / LLM (bert_clean)
Lowercasing	yes	no (casing is signal)
Unicode normalisation	yes	yes
URL / mention to placeholder	yes (<url>, <user>)	yes
Hashtag	keep word, drop #	keep word, drop #
Punctuation removal	yes (apostrophes kept)	no
Stemming / stop-word removal	none	none
Whitespace collapse	yes	yes

The rationale is that the classical model depends on surface-level frequency statistics and benefits from aggressive normalisation, whereas the subword tokeniser used by transformers (and by the LLM) exploits casing and morphology that aggressive cleaning would destroy. This is not a cosmetic choice. Applying classical-style normalisation to transformer inputs measurably degrades them, because it strips signal the tokeniser is built to use.

4. Methods

4.1 Baseline: TF-IDF and Linear SVM

A `TfidfVectorizer` (unigrams and bigrams) feeds a `LinearSVC`. Class imbalance is addressed at the classifier level via `class_weight="balanced"` rather than by resampling. The model represents text as sparse frequency vectors and learns linear decision boundaries between classes. It is fast, fully interpretable, and serves as a strong, honest reference point.

Hyperparameters: `ngram_range=(1,2)`, `min_df=2`, `max_df=0.95`, `sublinear_tf=True`, `C=1.0`, `class_weight="balanced"`.

4.2 Encoder: DistilBERT Fine-Tuning

`distilbert-base-uncased` (approximately 67M parameters; Sanh et al., 2019) is fine-tuned end-to-end with a classification head. DistilBERT retains most of BERT's representational power (Devlin et al., 2019) at roughly 40% of the size, making repeated experimentation tractable. The best checkpoint is selected on validation macro-F1.

Hyperparameters: maximum sequence length 128, learning rate $2e-5$, train batch 16, eval batch 32, 3 epochs, weight decay 0.01, AdamW optimiser. To assess stability, the model is trained under three random seeds (42, 123, 2024) and results are reported as mean $\pm$ standard deviation.

4.3 Generative LLM: Constrained Structured Output

The third contender is a frontier generative model, Claude Haiku 4.5, used without any training. The key methodological design is that the model is not asked to free-write a label. Instead it is forced to call a tool whose schema constrains the answer to exactly one of the six classes:

```
{
  "name": "submit_emotion",
  "description": "Submit the single dominant emotion for the input text.",
  "input_schema": {
    "type": "object",
    "properties": {
      "emotion": {
        "type": "string",
        "enum": ["sadness", "joy", "love", "anger", "fear", "surprise"],
        "description": "The dominant emotion expressed by the text."
      }
    }
  },
  "required": ["emotion"]
}
```

This eliminates brittle text parsing and, in principle, off-vocabulary answers. The system prompt additionally provides a one-line gloss per class to disambiguate near-synonyms (for example, love vs joy, fear vs surprise). Two prompting regimes are evaluated:

- **Zero-shot:** task description plus the input text.
- **Few-shot:** the same, prefixed with $k = 6$ labelled examples drawn from the training split (one per class, no test-set leakage), following the in-context-learning paradigm (Brown et al., 2020).

Because per-call LLM inference incurs real money and latency, the LLM is evaluated on a stratified subset of the test set ($n \approx 499$) rather than the full 41,000 rows. This is a deliberate cost-control decision. The sample size is reported alongside every result and the stratification preserves class proportions.

4.4 Small Open LLM: QLoRA (implementation complete, evaluation pending)

To probe a middle path between prompting and full fine-tuning, the project implements a 4-bit QLoRA (Dettmers et al., 2023) adapter over a small open instruction model (Qwen2.5-1.5B-Instruct), using Low-Rank Adaptation (Hu et al., 2021) on the attention projections ($r=16$, $\alpha=32$, $\text{dropout}=0.05$). Each example is framed as an instruction-to-label chat turn. The training script is complete and tested. A full multi-seed evaluation is reserved as future work pending GPU time, and no fabricated numbers are reported for it here.

5. Evaluation Methodology

5.1 Metrics

Accuracy is misleading under class imbalance, so macro-averaged F1 is the primary metric. It weights every emotion equally, ensuring poor performance on minority classes (surprise, fear, love) is not

masked by the majority classes. Accuracy and weighted-F1 are reported for completeness, but all comparisons and conclusions are based on macro-F1.

5.2 Cost and Latency as First-Class Metrics

A production model-selection decision turns on more than accuracy. We therefore measure two additional axes directly rather than asserting them.

Latency. Local models are timed over a fixed probe set with warmup, reporting p50 and p95 per-call latency (`src/benchmark.py`).

Dollar cost. The LLM harness records input and output token counts per call and converts them to USD via a per-model price table, reporting cost per 1,000 predictions.

5.3 Structured-Output Reliability

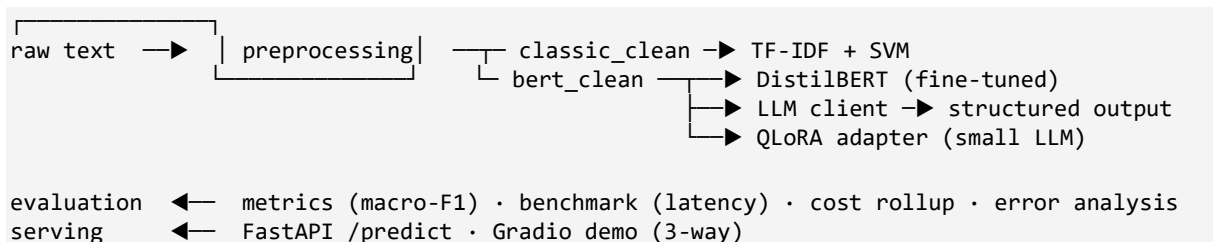
A subtle but important point: an enum in a tool schema is guidance, not a hard decoding constraint. We therefore measure how often the model returns a label outside the six classes and report it explicitly rather than hiding it (Section 7.5). Off-vocabulary returns are coerced to the nearest valid class (exact match, then substring match, then majority-class fallback), and the coercion rate is surfaced.

5.4 Reproducibility and Engineering

The pipeline is built as tested software, which is itself part of the contribution and distinguishes the study from a one-off notebook.

- **Centralised configuration** (`pydantic-settings`): every hyperparameter and path is environment-overridable. Nothing is hardcoded across scripts.
- **Deterministic seeding** across random, numpy, and torch.
- **On-disk response cache** for LLM calls, so re-runs are free and reproducible, with token usage persisted so cost accounting survives cache hits.
- **Client-side rate limiting** to respect provider request-per-minute limits, with retry and backoff that honour `Retry-After`.
- **92 unit tests** (mocked LLM providers, fake API registry, no network in CI) and a CI pipeline running lint (`ruff`), type-checking (`mypy`), and tests on Python 3.11 and 3.12.
- **Serving and packaging:** a FastAPI inference service and a Docker image.

System architecture



6. Experimental Setup

DistilBERT was trained on Apple Silicon (MPS). The SVM trains on CPU in minutes. The LLM was queried through the Anthropic API under client-side rate limiting (≤ 45 requests per minute). All randomness is seeded. The DistilBERT result is averaged over three seeds. Software: Python 3.11, PyTorch, Hugging Face Transformers, scikit-learn, and the Anthropic SDK.

7. Results

7.1 Headline Comparison

Table 3. Accuracy, latency, and cost across the four approaches. Local-model latency is measured at p95. LLM dollar cost is from measured token accounting.

Model	Test Accuracy	Test Macro-F1	p95 Latency	\$ per 1k preds
DistilBERT (3 seeds)	0.9399	0.9128 $\pm$ 0.0023	45.9 ms	$\approx$ 0 (local)
TF-IDF + Linear SVM	0.8858	0.8348	0.24 ms	$\approx$ 0 (local)
Claude Haiku 4.5, few-shot (k=6)	0.5872	0.5127	network-bound	\$1.27 (measured)
Claude Haiku 4.5, zero-shot	0.5772	0.5147	network-bound	$\sim$ \$0.40 (est.) ¹
Qwen2.5-1.5B + QLoRA	future work	future work	local	$\approx$ 0 (local)

¹ The zero-shot run was served from cache, so its tokens were not re-billed. Its true cost is necessarily lower than few-shot (which carries six in-context examples per call). Reported as an estimate rather than fabricated.

7.2 The Central Finding

The fine-tuned encoder is the clear overall winner, and the generative LLM is the clear loser on every axis that matters here. DistilBERT beats the LLM by approximately 40 macro-F1 points while costing nothing per prediction and running locally in tens of milliseconds. Even the classical SVM, sub-millisecond and trivially cheap, beats the frontier LLM by approximately 32 macro-F1 points. The LLM is simultaneously the least accurate and the most expensive option.

7.3 Few-Shot Did Not Help

Adding six in-context examples changed macro-F1 from 0.5147 (zero-shot) to 0.5127 (few-shot). This is a statistically meaningless, slightly negative move. The reading is diagnostic: the LLM's deficit is not a lack of examples but a mismatch with the dataset's specific labelling conventions. The CARER labels encode particular annotation decisions (for example, where joy ends and love begins) that a model which never saw the training distribution cannot infer from a handful of examples.

7.4 Robustness Across Seeds

To confirm the encoder result is not a lucky initialisation, DistilBERT was trained under three seeds (Table 4). The standard deviation of 0.0023 indicates highly stable performance.

Table 4. DistilBERT per-seed test macro-F1.

Seed	Accuracy	Macro-F1
42	0.9399	0.9096
123	0.9402	0.9143
2024	0.9395	0.9145
Mean	0.9399	0.9128 ± 0.0023

7.5 Structured-Output Reliability

Despite the enum schema, the model returned an off-vocabulary label (for example, *disappointment*, a synonym of sadness) on 5 of 499 zero-shot and 3 of 499 few-shot calls, roughly a 1% leakage rate. This is a practically important result. Schema enums constrain but do not guarantee the output space, so any production system using "*structured output*" still needs a validation-and-coercion layer. Here, such cases are mapped to the nearest valid class and counted, never silently dropped.

7.6 Latency

Table 5. Measured local-model latency per call.

Model	p50	p95
TF-IDF + SVM	0.15 ms	0.24 ms
DistilBERT	23.9 ms	45.9 ms

The approximately 190× latency gap between the SVM and DistilBERT is exactly the kind of trade-off this study exists to quantify. The encoder buys about 8 macro-F1 points over the SVM at two orders of magnitude more latency. The worth of that trade depends entirely on the deployment budget.

8. Error Analysis and Discussion

Three distinct error signatures emerge.

Encoder errors are semantic, not structural. DistilBERT's misclassifications concentrate at genuinely overlapping boundaries (love vs joy, surprise vs fear), where short texts plausibly admit multiple labels. The surprise class shows high recall but lower precision, indicating surprise-like cues sometimes appear in non-surprise text. These reflect the subjective, context-dependent nature of emotional expression rather than modelling failure.

The SVM degrades on minority classes. Its lower macro-F1 relative to accuracy is driven by the rare classes (surprise, love), where sparse n-gram features generalise poorly. This is precisely what macro-F1 is designed to surface.

The LLM disagrees with the dataset, not with sentiment. Manual inspection of LLM errors shows it is frequently *"reasonable but wrong"*: it assigns a defensible human label that disagrees with the dataset's annotation convention. This is the mechanism behind its low score and the reason few-shot does not rescue it. The gap is one of label calibration to a specific corpus, which fine-tuning supplies directly and prompting cannot.

The broader lesson is that benchmark accuracy for a prompted LLM partly measures agreement with the annotators' conventions, not raw capability. This is a distinction that matters when interpreting *"LLMs are good at everything"* claims.

9. Decision Framework

The experiment yields a practical, evidence-based selection rule.

Table 6. When to choose which approach.

Choose...	When...
TF-IDF + SVM	Sub-millisecond CPU latency or full interpretability is required. Training data is small. The marginal macro-F1 gain does not move the product.
Fine-tuned DistilBERT	Labelled data is plentiful and minority-class performance matters. A 10 to 50 ms latency budget is acceptable. The production sweet spot on this task.
Prompted LLM	There is no labelled data (cold start), the label set shifts rapidly, or volume is low enough that per-call cost is irrelevant.
Small LLM + QLoRA	Generative flexibility is desired but the weights and per-prediction cost must be owned in-house.

The headline takeaway is not *"LLMs are bad."* It is that an off-the-shelf LLM is the wrong default when you already have labels. Its value is concentrated in the cold-start regime. Once a few thousand labelled examples exist, a fine-tuned encoder dominates on accuracy, latency, and cost simultaneously.

10. Limitations and Future Work

Single LLM family. Only Claude Haiku 4.5 was evaluated. Larger models (Sonnet, Opus, GPT-4-class) would likely close part of the accuracy gap but widen the cost gap. The harness is provider-agnostic and supports adding them.

LLM evaluated on a subset. Cost control limits the LLM evaluation to approximately 499 stratified samples. The encoder and SVM use the full test set.

QLoRA not yet evaluated. The fourth contender is implemented but awaits a GPU run.

English, short-form text only. Results may not transfer to long-form or non-English data.

No hyperparameter sweep. Reasonable defaults are used throughout. A sweep would likely lift all contenders similarly, leaving the relative ordering intact.

Emotion labels are a simplification of continuous, often mixed affect. See the accompanying model card for ethical considerations.

Future work. Evaluate the QLoRA contender across seeds. Add larger LLMs and a chain-of-thought prompting variant. Finalise the zero-shot cost with an un-cached run. Test the framework on a second, differently-annotated dataset to probe the label-calibration hypothesis directly.

11. Conclusion

This study reframes a standard emotion-classification comparison as a production decision and measures it on accuracy, latency, and cost. The evidence is clear and somewhat against the prevailing instinct. On a label-rich, well-scoped task, a fine-tuned 67M-parameter encoder (0.9128 macro-F1) decisively beats a prompted frontier LLM (~0.51), and even a classical SVM (0.8348) beats it. The LLM is also approximately 190× slower than the SVM and costs roughly \$1.27 per 1,000 predictions against effectively zero for the local models. In-context examples did not help, because the deficit is one of label calibration, not capability.

The practical conclusion is a disciplined one. Reach for a generative LLM when you lack labels. Reach for a fine-tuned model when you have them. Beyond the empirical result, the project demonstrates the engineering practice (measured benchmarks, structured-output validation, caching, rate limiting, tests, and CI) needed to make such a comparison trustworthy rather than anecdotal.

Code Availability

The full pipeline, training scripts, evaluation harness, FastAPI inference service, and Docker image are available on request and are scheduled for open-source release alongside this paper. A three-way classifier demo (SVM, DistilBERT, and LLM via API) will be hosted on Hugging Face Spaces.

References

- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). *Language Models are Few-Shot Learners*. NeurIPS.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs*. NeurIPS.

- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. NAACL.
- Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. ICLR.
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing.
- Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. (2018). *CARER: Contextualized Affect Representations for Emotion Recognition*. EMNLP.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). *Attention Is All You Need*. NeurIPS.