FLARE: An Error Analysis Framework for Diagnosing LLM Classification Failures

Keerthana Madhavan, Luiza Antonie, Stacey D. Scott
School of Computer Science, University of Guelph
Guelph, Ontario, Canada
{kmadhava, lantoine, stacey.scott}@uoguelph.ca

Abstract

When Large Language Models return "Inconclusive" in classification tasks, practitioners are left without insight into what went wrong. This diagnostic gap can delay medical decisions, undermine content moderation, and mislead downstream systems. We present FLARE (Failure Location and Reasoning Evaluation), a framework that transforms opaque failures into seven actionable categories. Applied to 5,400 election-misinformation classifications, FLARE reveals a surprising result: Few-Shot prompting—widely considered a best practice—produced 38× more failures than Zero-Shot, with 70.8% due to simple parsing issues. By exposing hidden failure modes, FLARE addresses critical misunderstandings in LLM deployment with implications across domains.

1 Introduction

Large Language Models (LLMs) are now the workhorse for text classification across industry and academia[1, 13], handling hundreds of millions of calls each month in tasks from social-media filtering to biomedical triage and legal review [5]. Yet when an LLM responds with the catch-all label *"Inconclusive"*, it is difficult to know whether the prompt was too ambiguous for the model to understand, or whether the model incorrectly parsed it or simply failed [21]. This uncertainty stalls debugging and deployment.

Risks are most acute in high-stakes settings: a single unexplained "Inconclusive" can delay treatment [20], erode trust in moderation [7], distort sentiment analysis [8], or silently propagate errors in automated labelling [14]. Understanding *why* an LLM hesitates is therefore critical for responsible use.

In practice, "Inconclusive" emerges when models cannot confidently map input text to predefined categories—but this label provides no diagnostic information about why classification failed. Without understanding these failure modes, practitioners resort to trial-and-error prompt adjustments that may worsen rather than improve performance.

Prior work has pushed accuracy upward through prompt engineering—Zero-Shot (ZS), Few-Shot (FS) [3], and richer In-Context Learning (ICL)—and through calibration metrics. However, existing studies rarely examine the character of failures themselves. Taxonomies often collapse uncertainty into a single bucket and tacitly assume FS prompting is a safe upgrade over ZS. This leaves a methodological gap: practitioners lack a systematic way to diagnose failure modes hidden behind "Inconclusive" labels.

We close that gap with **FLARE** (Failure Location and Reasoning Evaluation)—a seven-category framework that distinguishes universal technical errors (e.g. parsing breakdowns) from domain-specific semantic errors (e.g. misclassification).

Our research questions are, *What specific failure modes trigger LLM "Inconclusive" classifications?* and *How do these failure modes vary across ZS, FS, and ICL prompting?*

To answer, we tasked GPT-4 Turbo with classifying 900 election-misinformation texts using Van der Linden's Six Degrees of Manipulation framework [16]. FLARE shows that Few-Shot prompting, contrary to belief, sharply increases error rates compared to Zero-Shot—mostly due to parsing rather than genuine ambiguity. These findings highlight misguided assumptions in LLM use.

Our contributions include:

1. **FLARE framework**, the first systematic error-analysis method for LLM classification failures.
2. **Empirical evidence** that Few-Shot prompting can degrade reliability by 38×.

2 Related Work

Error analysis has long helped linguists and engineers understand why NLP systems fail [6], but the advent of instruction-tuned LLMs introduces failure modes that classical, linguistically oriented taxonomies cannot capture [12]. Today's breakdowns often arise from prompt-induced biases or sensitivities, rigid output-format constraints, or inconsistent reasoning chains—phenomena absent from earlier work [19].

Most large-scale LLM evaluations remain performance-centric. Benchmarks such as HELM report aggregate accuracy, bias, and robustness

Proceedings of the First Interdisciplinary Workshop on OMMM associated with RANLP 2025,
pages 40–44, Varna, Bulgaria, Sep 8-10, 2025.
https://doi.org/10.26615/978-954-452-101-1-004

scores [9], while adversarial-trigger studies chart worst-case degradations [17]. Confidence-calibration research likewise stops at reliability curves rather than mapping specific errors [22]. Consequently, a model's ubiquitous *"Inconclusive"* output is treated as a single class of uncertainty, leaving practitioners blind to its underlying causes.

Prompting research further illustrates the gap. The seminal GPT-3 paper popularised Few-Shot prompting by highlighting accuracy gains [3], and subsequent surveys catalogue prompt patterns and macro-level improvements across datasets [15]. Yet these studies rarely dissect *how* the remaining errors differ from one prompting paradigm to another.

A parallel line of work explores LLMs as data annotators. Synthetic labels can complement scarce human annotations, especially for rare classes [11, 18], yet the evaluations still focus on aggregate scoreboards—overall accuracy, averaged F1, or raw agreement with humans—while leaving the underlying failure types unexplored.

Across these threads, researchers have examined *how well* LLMs classify or annotate, but little work has systematically investigated why these models fail—particularly in cases where the model self-reports an *"Inconclusive"* outcome. FLARE fills this research gap by categorising seven distinct failure modes and empirically demonstrating that popular Few-Shot prompting can *amplify* certain technical errors by 38×. FLARE labels are orthogonal to accuracy metrics, they complement existing evaluations and provide actionable diagnostics for researchers in HCI, psychology, AI ethics, and NLP alike.

3 Methodology

3.1 Research Design

We used a mixed-methods approach combining quantitative error counts with qualitative pattern analysis. Our dataset comprised 900 election-related misinformation texts classified using Van der Linden's Six Degrees of Manipulation framework [16]: Discrediting, Emotion, Polarization, Impersonation, Conspiracy, and Trolling.

3.2 Data Collection

Each text was classified by GPT-4 Turbo (deployment: gpt-4-phase1) under the three prompting paradigms described above. To capture output variability, we performed six independent classification runs per text with temperature=1.0, yielding 5,400 total classification attempts (900 texts × 6 runs for each prompt).

Figure 1 shows the Zero-Shot prompting template used in our study. The model was instructed to classify the text passages into one of six manipulation categories. When the model could not confidently assign a manipulation category, it returned *"Inconclusive"*—a catch-all label that masks the underlying reason for classification failure. The **Few-Shot** prompt appends two labelled examples per category, while the **In-Context** prompt further supplies formal definitions, guiding questions, and one worked example per label.

Zero-Shot Prompt Template

Classify the following text according to the 6 Degrees of Manipulation framework. Choose from: Emotion, Impersonation, Polarization, Trolling, Conspiracy, Discrediting.

Definitions: *Emotion* - emotive language to provoke reactions; *Impersonation* - false credible sources; *Polarization* - encourages division; *Trolling* - provokes without constructive intent; *Conspiracy* - secretive claims without evidence; *Discrediting* - undermines credibility without proof

Format: ¡Category¿: ¡Brief explanation¿

Figure 1: Zero-Shot prompt used to elicit manipulation category classification using the Six Degrees of Manipulation framework.

We extracted all instances where the aggregated final classification was "Inconclusive" (n=533) for detailed analysis across all three paradigms.

3.3 Framework Development

Following established qualitative research methods [4, 10, 2], we developed FLARE through iterative analysis of 533 classification failures. Figure 2 illustrates the complete FLARE framework and its application process. Our approach combined deductive reasoning (separating technical from semantic failures) with inductive pattern recognition (allowing categories to emerge from the data).

An output was marked *Inconclusive* if none of the six runs produced a valid `<Label>: <Explanation>` response. For Few-Shot prompting, the same two examples per category were reused across runs. Error categories were assigned via open coding, with researchers reviewing failures and reaching consensus.

The development process began with a manual review to isolate the obvious parsing errors. We then applied the open coding qualitative data analysis method [4] to the remaining failures. This analysis involved identifying recurring themes in the failure data through successive review passes and then systematically classifying (i.e., coding) the failure instances into those themes. Each instance was assigned to a single category that reflected its dominant failure mode.

The resulting framework was validated across all three prompting paradigms and accompanied by precise definitions and representative examples to ensure reproducibility.

4 The FLARE Framework

Our analysis revealed seven distinct failure types that the FLARE framework identifies in "Inconclusive"

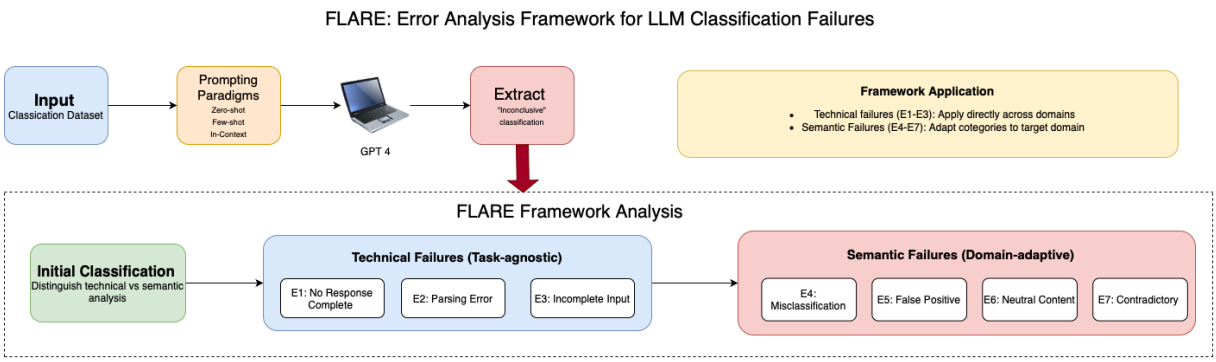

Figure 2: The FLARE error analysis framework for diagnosing LLM classification failures. The framework distinguishes between technical failures (E1-E3) that are universally applicable and semantic failures (E4-E7) that require domain-specific instantiation.

classifications:

E1: No Response. The model returns only "No response" across all attempts, indicating complete task failure. These cases typically involve inputs that trigger safety filters or exceed model capabilities.

Example:"BLM-antifa thugs are running around and beating people on the streets in DC. Video by"
Output: ['No response', 'No response', 'No response', 'No response', 'No response', 'No response']

E2: Parsing Error. Output formatting breaks down, with models failing to follow the required `<Label>: <Explanation>` format. Classifications are attempted but outputs include prefixes like "Classification:", inconsistent formatting, or responses like "None" or "Not Applicable" that fall outside the six allowed categories.

Example: "Tennessee is certainly leading the country in innovation..."
Outputs: ['Classification: None', 'This text does not contain any manipulation...', 'Not Applicable']
Issue: Invalid labels outside the six-category framework

E3: Incomplete Input. The model correctly identifies truncated or insufficient input but defaults to "Inconclusive" rather than explicitly flagging the input issue or requesting complete content.

Example: "[NAME] you too!"
Response: "This text does not contain enough information... Therefore, there is no label that can be accurately applied"

E4: Misclassification. Clear instances of manipulation categories the model fails to recognize, marking them as *"Inconclusive"* despite obvious indicators and even correct explanations from some annotators.

Example: "11,000 [NAME] residents get incorrect voter registration forms... This will be the most corrupt Election!"
Result: 5/6 annotators correctly identified "Discrediting" but final classification was *"Inconclusive"*.

E5: Not Applicable/False Positive. Neutral content that falls outside the classification scheme but which the model attempts to force into manipulation categories, revealing task overfitting.

Example: "We don't allow filming inside of the [NAME] unless there is a specific reason"
Issue: Non-political policy statement marked *"Inconclusive"* rather than noted as out-of-scope.

E6: Neutral Content Misrecognition. Legitimate political discourse incorrectly flagged as potentially manipulative, indicating the model cannot distinguish between criticism and manipulation.

Example: "Women and Minorities in STEM... supports research and Extension projects..."
Issue: Straightforward funding announcement labeled "Impersonation" by some annotators

E7: Contradictory Explanations. The model provides inconsistent reasoning, with different annotators assigning incompatible categories to the same input.

Example: "Tks to Margaret 4 joining me in DC to share successes..."
Disagreement: Split between "Emotion" (gratitude) and "Trolling" (informal style)

5 Results

5.1 Error Distribution Across Paradigms

Table 1 presents the distribution of FLARE-identified error types across the three prompting paradigms. The results reveal striking differences in both error frequency and type.

Few-Shot prompting exhibited a catastrophic 52.9% error rate, compared to 1.4% for Zero-Shot and 4.9% for In-Context Learning. Most remarkably, 337 of 476 Few-Shot errors (70.8%) were parsing failures (E2), suggesting that the inclusion of examples without sufficient structural guidance overwhelms the model's output generation capabilities.

Table 1: FLARE error analysis across prompting strategies

Error Type	ZS	FS	ICL	Total
E1: No response	13	13	13	39
E2: Parsing error	0	337	0	337
E3: Incomplete input	0	7	2	9
E4: Misclassification	0	34	10	44
E5: False positive	0	29	3	32
E6: Neutral content	0	40	8	48
E7: Contradictory	0	16	8	24
Total	13	476	44	533
Error Rate	1.4%	52.9%	4.9%	–

5.2 Semantic vs. Technical Failures

Our analysis reveals a critical distinction between semantic failures (E4-E7) and technical failures (E1-E3). While semantic failures might benefit from improved training data or refined prompts, technical failures require architectural or prompt engineering solutions. The dominance of technical failures in Few-Shot prompting (74.8% of errors) challenges the assumption that providing examples inherently improves model performance.

5.3 Cross-Paradigm Patterns

Certain error types appeared consistently across paradigms. All three approaches produced exactly 13 E1 (No Response) errors on the same inputs, suggesting these represent hard limits of the model rather than prompt-specific issues. Conversely, E2 (Parsing Error) appeared exclusively in Few-Shot prompting, indicating a specific interaction between example-based prompts and output generation.

6 Discussion

6.1 Implications for Prompt Engineering

Our findings challenge the assumption that Few-Shot prompting reliably improves performance. The 38-fold error increase—driven by parsing—shows FS prompts add complexity models struggle to handle. Even when labels were correct, outputs breaking the required `<Label>: <Explanation>` format (e.g., `Classification: Conspiracy`) were counted as errors, since such deviations disrupt pipelines. Zero-Shot rarely produced such errors because its format was simpler, whereas Few-Shot examples added prefixes and extra text that diverted the model from the strict format. These results highlight risks where reliability outweighs marginal gains.

6.2 The Hidden Cost of *"Inconclusive"*

By disaggregating *"Inconclusive"* into seven distinct failure types, the FLARE framework reveals that most failures are preventable through targeted interventions. Technical failures (E1-E3) require different solutions than semantic failures (E4-E7). For instance, the 337 parsing errors in Few-Shot prompting could potentially be eliminated through better output format specification or post-processing, while the 34 misclassifications might require model fine-tuning or improved examples.

6.3 Generalizability of FLARE

While demonstrated on misinformation detection, FLARE's structure suggests broad applicability as an error analysis method. Technical failures (E1-E3) are task-agnostic—parsing errors and non-responses occur across all classification tasks. Semantic failures (E4-E7) require domain adaptation but provide a template: replace "manipulation categories" with domain-specific classes. Researchers can adopt FLARE by (1) applying E1-E3 directly, (2) instantiating E4-E7 for their domain, and (3) extending with domain-specific categories as needed.

7 Limitations and Future Work

This study evaluates FLARE on a single model—GPT-4 Turbo—and one domain—election misinformation. Replicating the analysis with other models, tasks, and languages will be essential to confirm its generality. We also did not evaluate Chain-of-Thought prompting or structured-output interfaces, which may mitigate parsing failures. Automating the FLARE labelling process is another priority, so the framework can scale beyond manual annotation.

At present, FLARE assigns exactly one error label per instance; in practice, a failure can exhibit several problems at once. Future work should investigate hierarchical or multi-label variants of the taxonomy. We also plan to apply FLARE to higher-stakes settings such as medical-triage advice and safety-critical HCI scenarios, where understanding hidden failure modes is especially urgent.

8 Conclusion

We presented FLARE, an error analysis framework that transforms opaque *"Inconclusive"* classifications into actionable error diagnoses. Through systematic analysis of 533 failures, we demonstrated that Few-Shot prompting can increase error rates by 38-fold, with 70.8% of failures attributable to parsing errors rather than semantic challenges.

These findings have immediate practical implications for LLM deployment. Rather than assuming Few-Shot prompting improves performance, practitioners should evaluate error rates and types alongside accuracy metrics. The FLARE framework provides a method for such evaluation, enabling targeted debugging and informed deployment decisions.

References

[1] R. Bommasani, J. Hudson, E. Adeli, and et al. On the opportunities and risks of foundation models.

In *Proceedings of the 1st Workshop on Foundation Models*, 2021.

[2] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

[3] T. B. Brown, B. Mann, N. Ryder, and et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, 2020.

[4] J. M. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE, Thousand Oaks, CA, 4 edition, 2015.

[5] R. El-Yosef, H. Palangi, and F. Ahmed. Large language models in industrial text classification: A survey. *IEEE Intelligent Systems*, 39(2):56–68, 2024.

[6] R. Huidrom and A. Belz. A survey of error annotation schemes for human and machine generated text. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 383–398, 2022.

[7] D. Kumar, Y. AbuHashem, and Z. Durumeric. Watch your language: Investigating content moderation with large language models. *arXiv preprint arXiv:2309.14517*, 2024.

[8] M. Leippold. Sentiment spin: Attacking financial sentiment with gpt-3. *Finance Research Letters*, 55:103957, 2023.

[9] P. Liang, R. Bommasani, D. Tsipras, and et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

[10] M. B. Miles, A. M. Huberman, and J. Saldaña. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE, Thousand Oaks, CA, 3 edition, 2014.

[11] A. G. Møller, A. Pera, J. Dalsgaard, and L. Aiello. The parrot dilemma: Human-labeled vs. LLM-augmented data in classification tasks. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 179–192, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics.

[12] A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig. Stress test evaluation for natural language inference. In *Proceedings of COLING 2018*, pages 2340–2353, 2018.

[13] OpenAI. Api usage and adoption statistics. https://openai.com/blog/api-stats-2024, 2024. Accessed July 2025.

[14] N. Pangakis and S. Wolken. Knowledge distillation in automated annotation: Supervised text classification with llm-generated training labels. In *Proceedings of the 6th Workshop on NLP and Computational Social Science*, 2024.

[15] S. Schulhoff, M. Ilie, N. Balepur, and et al. The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2025.

[16] S. van der Linden. *Foolproof: Why Misinformation Infects Our Minds and How to Build Immunity*. W. W. Norton & Company, New York, NY, 2023.

[17] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2153–2162, 2019.

[18] H. Zhang, Y. Pei, S. Liang, and S. H. Tan. Understanding and detecting annotation-induced faults of static analyzers. *Proc. ACM Softw. Eng.*, 1(FSE), July 2024.

[19] T. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 2021.

[20] S. Zhou, Z. Xu, M. Zhang, C. Xu, et al. Large language models for disease diagnosis: A scoping review. *npj Artificial Intelligence*, 1(9), 2025.

[21] C. Zhu, B. Xu, Q. Wang, Y. Zhang, and Z. Mao. On the calibration of large language models and alignment. *Findings of EMNLP*, 2023.

[22] C. Zhu, B. Xu, Q. Wang, Y. Zhang, and Z. Mao. On the calibration of large language models and alignment. *Findings of EMNLP*, 2023.