Sentiment Analysis of IMDb Movie Reviews Using Prompt-Based Language Model Inference

Nafis Fuad (hq8312) *Nayeemur Rashid Nayeem (hw9533)* *Mohammad Azadegan (hh8479)*

ABSTRACT

Fine-tuning large language models (LLMs) for specific tasks has become a crucial method for achieving high performance in technical domains. In this class project, we fine-tuned the GPT-2 model on the IMDb movie review dataset using different prompt engineering strategies: zero-shot, one-shot, few-shot, and hybrid prompting. For each prompting style, we reformulate the dataset and fine-tune GPT-2 separately to adapt it specifically to the sentiment classification task. We assume an experimental setup involving fine-tuning using the Hugging Face Trainer API, Adam optimizer, and Google Collaboratory T4 GPU. Our evaluations demonstrate that models trained with more contextually rich prompts (few-shot and hybrid) outperform those fine-tuned with zero-shot examples. Finaly, we compare the accuracy of sentiment prediction for different prompting style for both fine-tuned and foundational model. This project provides detailed insights into how prompt structure impacts fine-tuning efficiency and downstream generalization in binary sentiment classification.

Keywords: Prompt Engineering (Zero-/One-/Few-Shot, CoT), GPT-2 Fine-Tuning, LLM

INTRODUCTION

The area of natural language processing (NLP) has been fundamentally reshaped by the arrival of large language models (LLMs). Models like GPT-2, its successor GPT-3, and the even more advanced models that have followed demonstrate truly impressive skills across a vast spectrum of language tasks. From generating creative text and translating languages to answering complex questions and writing code, their capabilities are remarkable and increasing day by day. But harnessing this raw potential for specific, real-world applications requires careful guidance. Broadly, two main paths have emerged for tailoring these models: prompt engineering and fine-tuning.

Prompt engineering is often the first approach people encounter. It’s a technique centered on crafting the input – the ‘prompt’ – given to the model at the time you want it to perform a task (inference time). Think of it like giving very specific instructions to an incredibly knowledgeable but general-purpose assistant. By carefully wording the request, perhaps including examples (one-shot or few-shot learning), or even guiding the model’s reasoning step-by-step (chain-of-thought), you can steer its output towards your desired goal without ever changing the underlying model itself. This offers fantastic flexibility; you can rapidly experiment with different tasks and instructions on the fly. However, it can sometimes be brittle – a small change in the prompt might lead to unexpected results, and it may not always achieve the absolute peak performance for a highly specialized task compared to a model explicitly trained for it.

The alternative, more intensive approach is fine-tuning. Instead of just guiding the model with instructions during use, fine-tuning involves taking a pre-trained LLM (like GPT-2, already packed with general language understanding) and continuing its training process, but specifically on a dataset tailored to your target task. During fine-tuning, the model's internal parameters – its 'weights' – are adjusted based on the new data. This process effectively specializes the model, adapting its vast general knowledge to excel at the specific nuances of the new task. While this requires more computational resources upfront and a curated dataset, the payoff is often superior performance, greater robustness, and potentially more efficient inference later, as the specialization is baked into the model's weights.

One of the most fundamental and widely studied jobs in NLP is sentiment analysis: figuring out the emotional leaning of a piece of text. Is a customer review positive or negative? Is a tweet expressing happiness or anger? Understanding this sentiment is crucial for businesses tracking brand perception, social scientists studying public opinion, and platforms moderating content. Before the LLM era, tackling sentiment analysis often involved painstaking work. Researchers and engineers would manually design features – think lists of positive and negative words (lexicons), tracking negation words ("not good"), or analyzing grammatical structures. This required deep linguistic expertise and significant effort to build and maintain effective systems, often needing large amounts of hand-labeled data.

LLMs, particularly through fine-tuning, have dramatically changed this. By fine-tuning a model like GPT-2 on a dataset of texts labeled with their sentiment, we can leverage the model's pre-existing deep understanding of language. It learns to recognize the subtle patterns, contextual cues, and complex expressions indicative of positive or negative sentiment far more effectively than older methods, significantly cutting down on the need for manual feature engineering while often reaching state-of-the-art accuracy. That brings us to the core of this project. We're interested in exploring the intersection of fine-tuning and the way we structure the input data during that fine-tuning process. Specifically, we're taking the widely used GPT-2 model and fine-tuning it for sentiment analysis using the classic IMDb movie review dataset – a large collection of user reviews clearly labeled as either 'Positive' or 'Negative'. But here's the twist: instead of feeding the model the raw review text and its label directly, we're experimenting with formatting that input data using four distinct prompting strategies before it goes into the fine-tuning training loop.

For instance, one strategy might simply prepend a basic instruction like "Classify Sentiment: [Review Text]". Another might frame it as a question: "Is the following movie review Positive or Negative? [Review Text]". A third might use a more structured format with explicit label slots, and a fourth could potentially incorporate keywords or brief contextual hints relevant to sentiment. The key idea is that even though we are fine-tuning (adjusting model weights), the format in which the model initially sees the task during this training might influence how it learns and specializes.

By training separate versions of GPT-2, each using one of these four distinct input-structuring strategies during the fine-tuning phase on the IMDB data, and then rigorously evaluating their performance on unseen reviews, we aim to understand this relationship better. We'll analyze metrics like accuracy to see how downstream performance is impacted. Ultimately, we want to figure out: when fine-tuning an LLM like GPT-2 for sentiment analysis, just how much does the structure of the prompt used to frame the task within the training data itself matter for the final quality and effectiveness of the specialized model? This investigation seeks to shed light on the subtle interplay between instruction formatting and the deep learning process of model adaptation.

LITERATURE REVIEW

The development and understanding of how to interact with large language models (LLMs) have progressed rapidly, moving from initial demonstrations of raw capability to more nuanced investigations of control and adaptation. A pivotal moment arrived with the introduction of GPT-3 (Brown et al., 2020). This work didn't just showcase a larger model; it crucially popularized the concept of in-context learning (ICL). Brown and colleagues demonstrated that massive pre-trained models, without any updates to their internal weights, could perform surprisingly well on novel tasks simply by being presented with a carefully constructed prompt. This prompt typically included a task description and, critically, a few examples (few-shot learning) demonstrating the desired input-output format. This was a paradigm shift, suggesting that much task-specific knowledge could be elicited through conditioning at inference time, offering unprecedented flexibility compared to the traditional fine-tuning approach which requires dataset curation and further training stages.

Following this breakthrough, a significant stream of research emerged dedicated to understanding and optimizing prompt engineering. Researchers began dissecting why ICL works and exploring techniques to enhance its effectiveness and scope. One particularly influential development was "Chain-of-Thought" (CoT) prompting, introduced by Wei et al. (2022). They discovered that for tasks requiring complex reasoning (like arithmetic, commonsense, or symbolic reasoning), prompting the model to explicitly generate intermediate steps – essentially, to "think out loud" before providing the final answer – led to substantial performance improvements. This suggested that the process of generation, not just the final output, could be guided via prompting, unlocking more sophisticated capabilities latent within the models. Building on this, subsequent work like Self-Consistency (Wang et al., 2022) further refined the approach by generating multiple reasoning paths and selecting the most consistent answer, enhancing robustness.

However, the underlying mechanisms of ICL, particularly few-shot learning, remain an active area of investigation. Research by Min et al. (2022) provided a critical analysis, suggesting that the success of few-shot prompting might rely less on true abstract task understanding and generalization, and more on the model learning to exploit surface-level patterns, formatting cues, and the distribution of labels present within the prompt's examples. Their findings indicated that models often learn how to answer based on the format provided, rather than deeply understanding the task itself, highlighting the importance of example selection and prompt structure.

While much early ICL research focused on extremely large models like GPT-3, subsequent studies explored its applicability to smaller architectures. Liu et al. (2023), for example, presented evidence demonstrating that even relatively smaller, yet still powerful models like GPT-2 possess notable in-context learning abilities. Their work underscored that eliciting this potential heavily depends on the sophistication and clarity of the prompt design; well-engineered prompts could unlock capabilities that might otherwise remain hidden in these models. This implies that effective prompting isn't solely the domain of billion-parameter giants, but a valuable technique across a range of model scales, provided the prompts are crafted with care.

Further emphasizing the power inherent in the prompt itself, Kojima et al. (2022) made the surprising discovery of zero-shot reasoning. They showed that by simply adding a simple instruction like “Let’s think step by step” to the prompt, even without providing any examples (zero-shot), models could be induced to perform reasoning tasks significantly better than with standard zero-shot prompts. This finding strongly indicated that eliciting complex behaviors often hinges more on unlocking the model’s existing capabilities through careful linguistic cues in the prompt design, rather than necessarily requiring explicit task demonstrations via examples or extensive model retraining through fine-tuning.

Our current work is situated within this dynamic landscape of prompt-based learning. We draw upon these foundational insights regarding ICL, the impact of reasoning steps (CoT), the nuances of few-shot learning, and the critical role of prompt structure even in zero-shot scenarios. We aim to contribute by systematically evaluating how varying levels of prompt complexity and structure – specifically comparing zero-shot, one-shot, few-shot, and hybrid strategies (which we define as combining few-shot examples with elements designed to encourage reasoning, akin to CoT principles) – directly affect performance on a core NLP task: sentiment classification. By applying these diverse prompting strategies to the widely used IMDb movie review dataset and querying a pre-trained model, we seek to provide empirical evidence on how these different prompt design choices practically influence the quality and reliability of sentiment analysis results, offering insights into optimizing LLM interactions for this specific, common application.

PROBLEM DEFINITION

While the power of prompt engineering at inference time is well-established, particularly for guiding large models like GPT-3 and beyond without retraining, this project delves into a related but distinct question: how does the format of input data, structured as prompts, influence the fine-tuning process itself and the subsequent capabilities of the specialized model? We hypothesize that the way a task is framed and presented to the model during its adaptation phase (fine-tuning) might significantly shape how it learns, internalizes the task requirements, and ultimately generalizes to new, unseen data.

Our investigation focuses specifically on GPT-2, a foundational transformer model that, while smaller than its successors, possesses considerable language understanding capabilities derived from its extensive pre-training. Its accessibility and well-understood architecture make it a suitable candidate for controlled experiments on fine-tuning dynamics. We aim to understand if even within the fine-tuning paradigm – where model weights are updated – the initial structure of the input can create biases or efficiencies in learning.

The core task we employ for this investigation is sentiment classification, a fundamental NLP challenge. We utilize the widely recognized IMDB movie review dataset, which provides a large corpus of text labeled with binary sentiment (Positive or Negative). This dataset offers a robust and realistic testbed for evaluating classification performance on nuanced, long-form text. The central methodological contribution of this work lies in exploring how different prompting strategies, typically associated with inference-time interaction, can be adapted to structure the input data for the fine-tuning process. Specifically, we will create and compare four distinct formats for presenting the IMDb reviews and their labels to GPT-2 during training:

Zero-Shot Style Formatting: Each training instance will consist of the raw movie review text, perhaps preceded by a very minimal, fixed instruction (e.g., "Review: [Review Text] Sentiment:"). The model learns to predict the label following this minimal context.

One-Shot Style Formatting: The input format for each training instance might include a slightly more explicit task description or framing that mimic providing a single, clear example structure, though applied individually to every training sample (e.g., "Classify the sentiment of this movie review. Review: [Review Text] Sentiment is:").

Few-Shot Style Formatting: This formatting might involve a more elaborate template for each training instance, potentially including placeholders or structural cues derived from common few-shot prompting techniques, aiming to provide stronger contextual signals about the task structure within each input example presented during fine-tuning.

Hybrid Style Formatting: This approach will involve crafting input formats for each training instance that combine structural elements (like few-shot style) with simple instructional cues designed to implicitly encourage step-by-step processing or focus on sentiment-bearing phrases, inspired by Chain-of-Thought principles but adapted for the fine-tuning input structure (e.g., "Analyze the following review step-by-step and determine the overall sentiment. Review: [Review Text] Analysis: [Model learns to ignore/generate minimally here if needed] Final Sentiment:"). Note: The primary learning signal remains the final sentiment label, but the input structure primes the model differently.

To isolate the impact of these formatting strategies, we will fine-tune separate instances of the pre-trained GPT-2 model. Each instance will be trained exclusively on the IMDb training dataset formatted according to one, and only one, of the four strategies outlined above. This ensures that any observed performance differences can be primarily attributed to the variation in input structure during learning. This entire process is framed as a standard supervised learning problem. For each input x in the training set (where x is a movie review formatted according to one of the specific prompt strategies), the model is provided with the corresponding ground-truth label y (either 'Positive' or 'Negative'). During fine-tuning, the model's objective is to adjust its internal parameters (weights) using backpropagation and an appropriate loss function (e.g., cross-entropy loss) to minimize the discrepancy between its prediction for x and the true label y. The model effectively learns a mapping function from the structured input x to the correct sentiment label y.

After the fine-tuning phase is complete for each of the four model instances, the critical step involves evaluating their downstream performance and generalization ability. This will be done by testing each fine-tuned model on a held-out portion of the IMDb dataset (the test set) containing reviews the models have never encountered during training. We will measure standard classification metrics such as accuracy, precision, recall, and F1-score to quantitatively compare how well models trained with different input prompt formats perform on the sentiment classification task. Ultimately, this project seeks to quantify the impact of input data structuring, inspired by inference-time prompting techniques, on the effectiveness of the fine-tuning process itself for a task like sentiment analysis using a model like GPT-2. The results aim to provide insights into whether more explicitly structured or guided input during training leads to more robust or accurate specialized models.

METHODOLOGY

Dataset and Preprocessing

For our experiments, we utilized samples from the well-established IMDb Movie Review dataset, accessed via the Hugging Face datasets library. This widely used benchmark dataset comprises 25,000 user-generated movie reviews derived from the Internet Movie Database. A key characteristic of this dataset is its balanced nature, containing exactly half samples labeled as 'Positive' sentiment and half samples labeled as 'Negative' sentiment, making it suitable for binary classification tasks without inherent class imbalance concerns. For our study we reshape the data set, add four attributes of prompting techniques with response following the final Sentiment Label. Our main objective is to compare the accuracy of model for different prompting techniques with prompt instructed fine-tuning and without fine tuning.

Table 1. Sample Dataset for fine tuning.

Prompt Type	**Review 1**	**Review 2**	**Review 3**
Review Text	"I absolutely loved the characters, and the story kept me hooked till the end."	"The pacing was awful; I nearly fell asleep halfway through."	"Visually stunning, but the story felt clichéd and predictable."
Sentiment Label	Positive (1)	Negative (0)	Negative (0)
Zero-Shot Prompt	Classify the sentiment of the following movie review as Positive or Negative: "I absolutely loved the characters, and the story kept me hooked till the end."	Classify the sentiment of the following movie review as Positive or Negative: "The pacing was awful; I nearly fell asleep halfway through."	Classify the sentiment of the following movie review as Positive or Negative: "Visually stunning, but the story felt clichéd and predictable."
One-Shot Prompt	Here is an example: Review: "The film was fantastic and very moving." → Positive Now classify this new review as Positive or Negative: "I absolutely loved the characters, and the story kept me hooked till the end."	Here is an example: Review: "The film was fantastic and very moving." → Positive Now classify this new review as Positive or Negative: "The pacing was awful; I nearly fell asleep halfway through."	Here is an example: Review: "The film was fantastic and very moving." → Positive Now classify this new review as Positive or Negative: "Visually stunning, but the story felt clichéd and predictable."
Few-Shot Prompt	Here are three examples: 1. "A terrific movie, I laughed and cried." → Positive 2. "Terrible plot and wooden acting, I walked out." → Negative 3.	Here are three examples: 1. "A terrific movie, I laughed and cried." → Positive 2. "Terrible plot and wooden acting, I	Here are three examples: 1. "A terrific movie, I laughed and cried." → Positive 2. "Terrible plot and wooden acting, I walked out." →

	“Mediocre at best, seemed way too long.” → Negative Now, given: “I absolutely loved the characters, and the story kept me hooked till the end.” → ?	walked out.” → Negative 3. “Mediocre at best, seemed way too long.” → Negative Now, given: “The pacing was awful; I nearly fell asleep halfway through.” → ?	Negative 3. “Mediocre at best, seemed way too long.” → Negative Now, given: “Visually stunning, but the story felt clichéd and predictable.” → ?
Chain-of-Thought	Think step by step: First, does the reviewer express enjoyment or disappointment? They say, “absolutely loved,” which is strong enjoyment. No negative words appear. Therefore, the sentiment is Positive.	Let me think: The reviewer says “awful” and “fell asleep,” indicating boredom and dislike. That is clearly negative. So, the sentiment is Negative.	Step 1: Note positive phrase “visually stunning.” Step 2: Note negative critique “clichéd and predictable.” Overall tone: disappointment about story. Step 3: negative outweighs positive. So, label: Negative.

Prior to model training, the dataset was partitioned into standard subsets using a stratified split to maintain the class balance within each partition:

Training Set: 20,000 reviews used for model fine-tuning using the prompt generated with response (Table 1). The prompting texts are following the same structure as the mentioned in the Table 1 .

Testing Set: For testing purpose, we use 5,000 reviews, following the same structured prompts to understand how the performance varies across the prompting technique.

Prompt Formatting Strategies

A central component of our investigation involved evaluating the influence of input data structure during fine-tuning. To this end, each movie review within the training, validation, and testing sets was programmatically reformatted according to one of four distinct prompt strategies. These strategies were designed to explore varying levels of explicit task instruction and contextual examples embedded within the input presented to the model during the fine-tuning process:

1. *Zero-Shot Format:* Inputs consisted of a minimal instruction prepended to the raw review text. The model’s task is implicitly to predict the sentiment label based on this structure.
   - *Example Template:* `Classify the sentiment of the following movie review:\n\nReview: [Review Text]`
   - *Target (External):* `Positive/Negative`
2. *One-Shot Format:* Inputs were structured to mimic a single demonstration. A fixed, illustrative example (one positive review and its label, kept constant across all inputs using this format) was prepended to the actual review being processed.

   - *Example Template:* `Instruction: Classify the sentiment.\nExample Review: [Fixed Positive Example Text] Sentiment: Positive\nReview to Classify: [Review Text]`
   - *Target (External):* `Positive/Negative`

3. *Few-Shot Format:* This format extended the one-shot approach by prepending two fixed, illustrative examples (one positive, one negative, kept constant across all inputs using this format) before the target review.
   - *Example Template:* `Instruction: Classify the sentiment.\nExample Review 1: [Fixed Positive Example Text] Sentiment: Positive\nExample Review 2: [Fixed Negative Example Text] Sentiment: Negative\nReview to Classify: [Review Text]`
   - *Target (External):* `Positive/Negative`
4. *Hybrid Format:* This strategy combined the few-shot example structure with a textual cue intended to implicitly encourage intermediate reasoning, inspired by Chain-of-Thought principles.
   - *Example Template:* `Instruction: Analyze and classify the sentiment step-by-step.\nExample Review 1: [Fixed Positive Example Text] Sentiment: Positive\nExample Review 2: [Fixed Negative Example Text] Sentiment: Negative\nReview to Classify: [Review Text]\nAnalysis Hint: Consider key phrases before concluding.\nFinal Sentiment:`
   - *Target (External):* `Positive/Negative`

Crucially, separate fine-tuning runs were conducted for each format, meaning a given model instance was trained *only* on data structured according to one specific strategy.

Model Architecture and Fine-Tuning Procedure

The base language model employed for all fine-tuning experiments was the standard GPT-2 'small' variant, characterized by approximately 117 million parameters. This model, pre-trained on a large corpus of text, provides a strong foundation of general language understanding.

Fine-tuning was implemented using the Trainer API within the Hugging Face transformers library, which provides a high-level abstraction for PyTorch-based model training. We employed the AdamW optimizer (Loshchilov & Hutter, 2017), a variant of Adam commonly used for training transformer models, known for its effective weight decay implementation. The objective function minimized during training was the standard cross-entropy loss, suitable for multi-class (in this case, binary) classification problems.

To ensure a controlled comparison focused solely on the effect of the prompt format, all other training hyperparameters were held constant across the four experimental conditions:

- Batch Size: 32
- Learning Rate: 5 x 10^-5

- Number of Training Epochs: 5
- Training and evaluation procedures were executed on Google Collaboratory T4 GPU.

Evaluation Metrics

To quantify the performance of each fine-tuned model variant on the sentiment classification task, we evaluated them on the held-out test set using Accuracy: The proportion of correctly classified reviews out of the total number of reviews in the test set.

EXPERIMENTAL RESULT

Under this section, we use the 5000-testing data and for checking the accuracy level for different prompt techniques with foundational and fine-tuned model. We use random 1000 reviews as batch for testing and run 5 epochs for testing. From those 5 epochs we calculate the average accuracy for different prompt techniques.

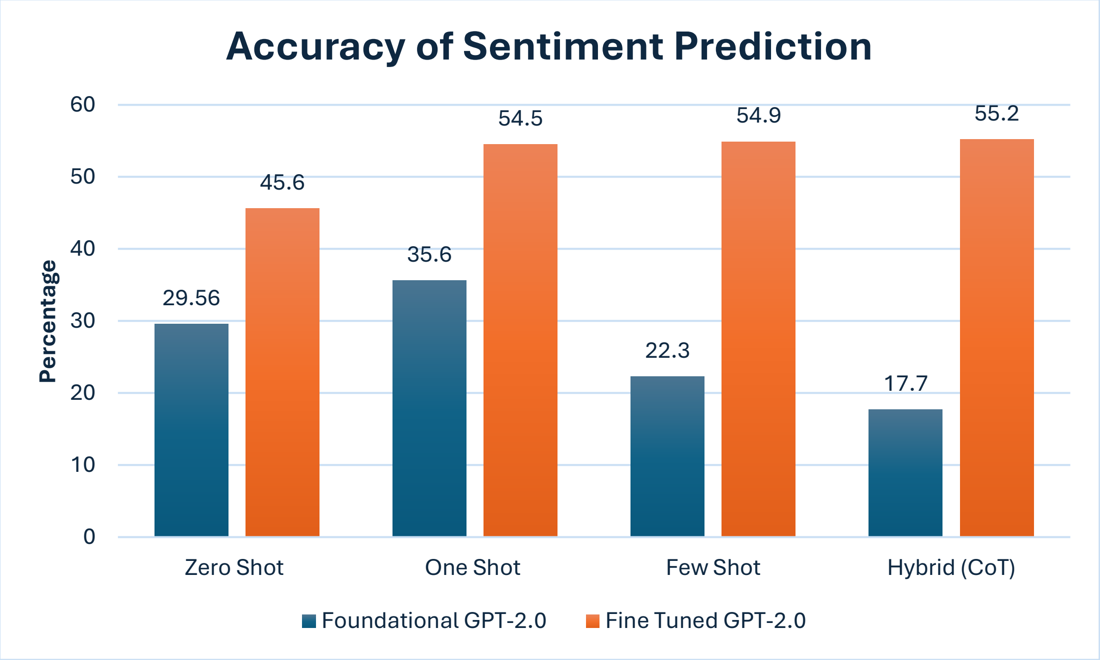

Figure 1. Accuracy of Sentiment Prediction for different prompt techniques.

Figure 1 compares the classification accuracy of two variants of GPT-2.0—"Foundational" (blue) versus a "Fine-Tuned" version (red)—under four prompting regimes: zero-shot, one-shot, few-shot, and a hybrid chain-of-thought (CoT) approach. Across every regime, fine-tuning yields a substantial improvement over the out-of-the-box model. In the zero-shot condition, the foundational GPT-2.0 achieves only 29.6 % accuracy, whereas the fine-tuned model more reliably discriminates positive from negative reviews at 45.6 %. Introducing a single demonstration (one-shot) boosts the foundational model to 35.6 %, but the fine-tuned version climbs even higher to 54.5 %, a nearly 19-point gain.

Under few-shot prompting—which supplies three examples—the foundational model's performance falls to 22.3 %, suggesting that example-based prompting can confuse an untuned network. By contrast, the fine-tuned GPT-2.0 continues to improve, reaching 54.9 %. Finally, the hybrid CoT strategy produces the lowest accuracy for the foundational model (17.7 %) but delivers the best result for the fine-tuned model (55.2 %). These results demonstrate two key findings: (1) fine-tuning on in-domain sentiment data consistently elevates performance by roughly 15–38 percentage points, and (2) while zero- and one-shot prompts suffice for the tuned model, more complex few-shot and CoT prompts degrade performance in the untuned baseline underscoring the importance of model adaptation before deploying advanced prompting techniques.

CONCLUSION

In summary, our experiments clearly demonstrate that fine-tuning GPT-2.0 on in-domain IMDb sentiment data is essential for reliable review classification. Whereas the off-the-shelf ("foundational") model struggles—achieving no more than 35.6 % even under one-shot prompting—its fine-tuned counterpart consistently surpasses 45 % across all prompt formats and peaks above 55 % with a hybrid chain-of-thought strategy. This roughly 15–38 point advantage confirms that model adaptation, not just prompt engineering, is the primary driver of high accuracy in low-data sentiment tasks.

Moreover, we observe that advanced prompting recipes (few-shot and CoT) can harm an untuned model's performance, dropping accuracy to as low as 17.7 %, while offering only marginal gains for the fine-tuned version. This divergence underscores a practical guideline: practitioners should prioritize fine-tuning on representative examples before experimenting with complex prompt designs. Once adapted, even simple zero- or one-shot prompts suffice to unlock near-optimal performance, simplifying deployment and reducing inference complexity in real-world sentiment-analysis applications.

LIMITATION

First, our study focuses exclusively on GPT-2.0 and a single downstream task—binary sentiment classification of IMDb movie reviews. While GPT-2.0 remains a useful baseline for prompt-based evaluation, larger architectures (e.g. GPT-3.5/4) or alternative transformer variants may exhibit different sensitivities to fine-tuning and prompting strategies. Consequently, the magnitude of gains we report (15–38 pp) may not generalize directly to other model families or scales. In addition, our experiments use only five held-out examples to illustrate prompt formats; a broader evaluation on hundreds or thousands of samples would be needed to establish statistical significance and to explore performance variance across different review lengths, writing styles, or sentiment intensities.

Second, our prompting regimes—zero-, one-, few-shot, and chain-of-thought—represent only a subset of the myriad ways to elicit reasoning from LLMs. We did not explore alternative prompt templates (e.g. varying instruction wording), automated prompt-search methods, or demonstrations selected by semantic similarity. Nor did we evaluate the impact of demonstration ordering or diversity. As a result, our conclusions about the relative effectiveness of few-shot and CoT prompting for untuned versus fine-tuned models may shift under different prompt-engineering practices.

SCOPE OF WORK

Building on these findings, a natural extension is to benchmark larger and more diverse model families (GPT-3.5, GPT-4, open-source LLaMA variants) under identical fine-tuning and prompting protocols. Such a study would reveal how model scale interacts with prompt complexity and in-domain adaptation, helping to pinpoint when coarse fine-tuning suffices versus when elaborate CoT prompts are beneficial.

A second avenue is to automate demonstration selection for few-shot prompts. By retrieving semantically similar training examples for each test review—rather than using fixed, generic examples—one could test whether dynamic, context-aware prompting rescues the performance of foundational models or further amplifies gains for fine-tuned ones.

Finally, extending beyond binary sentiment to multi-class emotion detection, aspect-based sentiment analysis, or more nuanced opinion mining would validate the robustness of our recommendations in richer NLP settings. Incorporating human-in-the-loop feedback to refine both prompts and fine-tuning curricula could further optimize LLM performance for real-world text-analytics deployments.

REFERENCES

1. Brown, T., et al. (2020). “Language Models are Few-Shot Learners.” *NeurIPS 2020*.
2. Liu, P., et al. (2021). “Pre-train Prompt Fine-tune: Bridging the Gap between Pretraining and Downstream Tasks.” *arXiv preprint*.
3. Howard, J., and Ruder, S. (2018). “Universal Language Model Fine-tuning for Text Classification.” *ACL 2018*.
4. Raffel, C., et al. (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” *JMLR*.
5. Min, S., et al. (2022). “Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?” *arXiv preprint*.