Learned Prompting for Sentiment Analysis with LLMs

Niklas Britz Karl Deck Marie-Louise Dugua Alexander Zank

Abstract

Sentiment analysis is a well-studied machine learning task that is challenging to solve even for state-of-the-art large language models (LLMs). We propose a learned prompting framework that achieves robust sentiment classification while treating LLMs as black boxes. Our method combines (1) a lightweight prompt selector that chooses optimal prompts from a fixed catalog based on input characteristics, and (2) an iterative prompt optimizer that evolves the prompt catalog using meta-prompting techniques.

Experiments demonstrate that our adaptive prompt selection significantly outperforms static prompting baselines, hand-crafted individual prompts, and ensemble voting techniques. While achieving competitive performance with fine-tuned discriminative models like DeBERTa and RoBERTa, our approach only requires training the lightweight selector model, avoiding expensive fine-tuning of the LLM itself. These results confirm that learned prompting is a viable alternative to expensive fine-tuning for sentiment analysis tasks.

1. Introduction

Sentiment analysis is the process of determining the general feelings conveyed in a text. It is an important field in Natural Language Processing (NLP). While it may appear straightforward, this task is generally challenging, even for state-of-the-art language models, due to the nuanced ways in which we express our emotions in text (Zhang et al., 2024). For example, sentiments are not always explicitly stated but are conveyed through context, background knowledge, and implications, or even stylistic choices that convey meaning in unconventional ways. This fact makes it challenging for language models to interpret sentiments in short texts reliably. There are many applications in which language models would be useful in understanding the sentiments conveyed in natural language. Reliable sentiment classification in reviews could, for example, support improved analysis of customer feedback (Fang & Zhan, 2015) and the automated evaluation of restaurants.

Sentiment classification methods have evolved from traditional lexicon-based and machine learning approaches (SVM (Korovkinas et al., 2018), Naive Bayes (Fang & Zhan, 2015)) through end-to-end learned or fine-tuned solutions (BERT (Devlin et al., 2019; Batra et al., 2021), EmoLLMs (Liu et al., 2024)) to modern prompt-based methods treating large language models as black boxes (Bu et al., 2024; Zhang et al., 2024).

The latter are closely related to the colloquial concept of "prompt engineering" (Lo, 2023) which aims to maximize the use of an LLM's latent knowledge for specific tasks by rephrasing the text to be completed. Prompts can range from simple instructions to highly task-specific formulations, and research shows that prompt design can significantly affect a model's output (Brown et al., 2020).

In this work, we investigate how prompt quality and selection impact sentiment analysis accuracy in language models. We propose an optimization pipeline (Figure 1) featuring two key components: a prompt selector (Sections 2.1 & 3.1) that dynamically chooses optimal prompts from a catalog based on input characteristics, and a prompt optimizer (Sections 2.2 & 3.2) that evolves the catalog by analyzing performance patterns and generating improved prompts through meta-prompting. Our experiments demonstrate that this approach significantly improves Gemma 3 (4B) (Google, 2025) sentiment analysis accuracy compared to static prompting or basic prompt catalogs, all without requiring model fine-tuning.

2. Methods

This section motivates and specifies each stage of our pipeline in isolation. Denote the classifiable sentiments as $\mathcal{S} := \{+, -, 0\}$ and the tokenizer's vocabulary as $\mathcal{V}$. In line with the definition of the Kleene Star, denote as $\mathcal{V}^*$ the set of all possible sentences.

In our work, a prompt is a tuple $(t, e, M) \in \mathcal{T}$ of string template $t \in (\mathcal{V} \cup \{\textsc{Input}\})^*$, evaluation strategy $e \in \{\textsc{Probe}, \textsc{Count}\}$, and sentiment map $M \subset V^* \times \mathcal{S}$.

How an LLM derives sentiments from prompt-input pairs is detailed in Algorithm 1. The evaluation strategy PROBE is deterministic and infers probabilities from the *LLM*'s logits. COUNT is generative and derives probabilities from

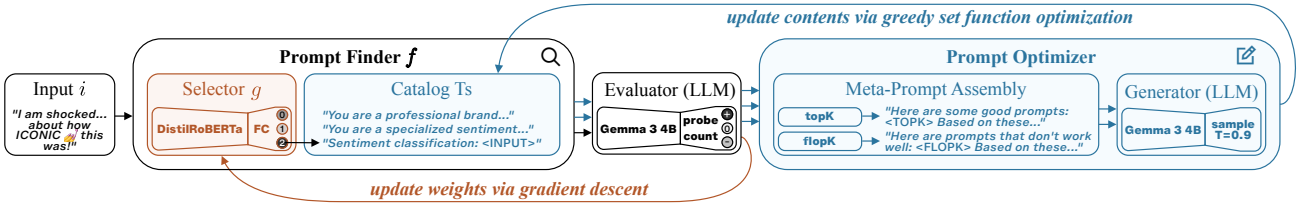

Figure 1. An overview of our pipeline combining all methods covered in Section 2. The selector (orange) and catalog (blue) are optimized separately. At inference time, data only flows along the black path, and the prompt optimizer is disabled.

the number of occurrences of mapped phrases.

Algorithm 1 PREDICTSENTIMENT(*LLM*)

Input: prompt $p = (t, e, M)$, classifiable input $i \in \mathcal{V}^*$
Output: $\{P[i \text{ has sentiment } s] \mid s \in \mathcal{S}\}$ under p

$t^i := t$ but replacing each occurrence of INPUT with i
if $e =$ PROBE **then**
  $(t_1^i, t_2^i, \ldots, t_n^i) := \textit{LLM.tokenize}(t^i)$
  for $(w, \_) \in M$ **do**
    $(w_1, w_2, \ldots, w_m) := \textit{LLM.tokenize}(w)$
    $p_w := \prod_{j \in [m]} P[\textit{LLM.next} = w_j \mid \textit{LLM.context} = (t_1^i, \ldots, t_n^i, w_1, \ldots, w_{j-1})]$
  end
  for $s \in \mathcal{S}$ **do**
    $p_s := \frac{1}{|\{(w,s) \in M\}|} \sum_{(w,s) \in M} p_w$
  end
  return $\left\{ \frac{p_s}{\sum_{s \in \mathcal{S}} p_s} \mid s \in \mathcal{S} \right\}$
else if $e =$ COUNT **then**
  $c :=$ *LLM.createCompletion*(t^i)
  for $(w, \_) \in M$ **do**
    $o_w :=$ # of occurrences of w in c
  end
  for $s \in \mathcal{S}$ **do**
    $o_s := \sum_{(w,s) \in M} o_w$
  end
  return $\left\{ \frac{o_s}{\sum_{s \in \mathcal{S}} o_s} \mid s \in \mathcal{S} \right\}$

Rather than fine-tuning LLM weights, we treat models as black boxes and optimize prompts directly. Given input sentence $i \in \mathcal{V}^*$, we seek a function

$$f : \mathcal{V}^* \to \mathcal{T} \text{ that minimizes}$$
$$\text{CE}(\text{PREDICTSENTIMENT}(\textit{LLM}, f(i), i), s^*)$$

where CE denotes the cross-entropy loss between predicted sentiment distribution and the one-hot encoded true sentiment $s^* \in \mathcal{S}$.

2.1. Prompt Selection

Since sentences vary in style, tone, and context, we hypothesize that selecting the most appropriate prompt from a heterogeneous catalog can improve classification performance. Given a fixed prompt catalog $Ts \subset \mathcal{T}$, we constrain our optimization to:

$$f \in \{i \mapsto Ts[g(i)] \mid g : \mathcal{V}^* \to [|Ts|]\}$$

Here, classifier g maps input sentences to indices in the prompt catalog. This classifier can be much simpler and faster to fine-tune than the evaluating LLM itself. For instance, our selection model DistilRoBERTa (Sanh et al., 2019) has only 82M parameters compared to the 3B parameters of our evaluating LLM Gemma 3B (Google, 2025). Additionally, freezing the LLM enables caching outputs between training epochs, substantially increasing sample throughput.

Our approach differs from ensemble methods (Tran & Matsui, 2024) that evaluate all prompts and aggregate their outputs (e.g., averaging probabilities). Instead of broad consensus, we focus on selecting the single most suitable prompt for each input, which we later show achieves better specialization with lower computational cost.

2.2. Prompt Optimization

Finding well-performing prompts through human authoring is expensive, while brute-force search is computationally intractable. In Algorithm 2, we propose a data-driven approach that automatically evolves prompt catalogs.

Starting with human-written prompts inspired by prompt engineering guides (Appendix A.1), our method iteratively improves the catalog. At each iteration, we evaluate all prompts on a training subset, then provide an LLM with the best and worst performers as examples to generate new prompts. To maintain catalog size and encourage novelty, we discard the worst-performing prompts after each generation.

We evaluate catalog performance assuming perfect prompt selection (Section 2.1) to encourage diversity. The operator $-^{\#} : \mathcal{S} \times \mathcal{S} \to \pm\{0, 1, 2\}$ used when calculating the

Algorithm 2 OPTIMIZEPROMPTCATALOG(*LLM*)

Input: initial prompt catalog $\text{Ts}^{(0)} \subset \mathcal{T}$, top $K \in \mathbb{N}$, training set $I \subset \mathcal{V}^* \times \mathcal{S}$, target epoch $J \in \mathbb{N}$
Output: refined prompt catalog $\text{Ts}^{(J)}$ with $|\text{Ts}^{(J)}| = |\text{Ts}^{(0)}|$

for $j \in [J]$ **do**
  for prompt $p \in Ts^{(k-1)}$ **do**
    for (input i, label s^*) $\in I$ **do**
      $\text{AE}_i := |s^* - \#$ $\arg\max_s$ PREDICTSENTIMENT$(LLM, p, i)|$
    end
    $\text{NMAE}_p := 0.5 \cdot (2 - \frac{1}{|\text{Ts}|} \sum_i \text{AE}_i)$
  end
  Tn := K new prompts the LLM generated based on the topK_p NMAE_p (bad) and $\text{topK}_p(1-\text{NMAE}_p)$ (good) prompts
  $\text{Ts}^{(j)} := \emptyset$
  while $|Ts^{(j)}| < |Ts^{(j-1)}|$ **do**
    $p_{\text{best}} := \arg\max_{p \in (\text{Tn} \cup \text{Ts}^{(j-1)}) \setminus \text{Ts}^{(j)}}$ accuracy of $\text{Ts}^{(j)} \cup \{p\}$ on the training set, assuming the best prompt is chosen for each input sentence.
    $\text{Ts}^{(j)} := \text{Ts}^{(j)} \cup \{p_{\text{best}}\}$
  end
end
return $\text{Ts}^{(J)}$

absolute error AE penalizes opposite-sentiment misclassifications more than neutral misclassifications. Since maximizing submodular set functions is NP-hard (Nemhauser et al., 1978), we use a greedy approximation for prompt selection within each iteration.

3. Evaluation

In our evaluation, we focus on showing the impact that both the prompt selector and the prompt optimizer have on predicting sentiments in our system. We additionally compare prompted learning to two discriminative models. To carry out experiments on our system, we manually created a standard prompt catalog that includes simple prompts for 10 areas of expertise (e.g., restaurants, movies, etc.), which can be found in Appendix A.1. The LLM we use for the prompt evaluator and the optimizer is Gemma 3 (4B) (Google, 2025) in its official Q4_0 quantization. For sampling, we chose the same parameters as Google in their paper.

3.1. Prompt Selector

To analyze the impact the prompt selector has on the overall performance of our system, we carried out an experiment to quantify the impact that this augmentation has on our system. Table 1 shows that choosing the prompt for each sentence adaptively instead of using a unique prompt for all sentences increases the overall performance of the LLM that receives (prompt, sentence) pairs. We see that the best-performing prompt ("Context"), which emphasizes not to put too much importance on single words, still performs significantly worse than the selector variant. A general prompt that does not include domain-specific instructions also performs significantly worse. We thus conclude that adaptively choosing a prompt by the input sequence increases the performance in our setting.

Figure 2 shows the relative frequency with which a specific prompt is called. We observed that the selector does not always pick the most successful prompt, but learns to associate specific sentences with the success of specific prompts. For example, by manually inspecting the samples, we saw that movie-related prompts were directed to the movie prompt.

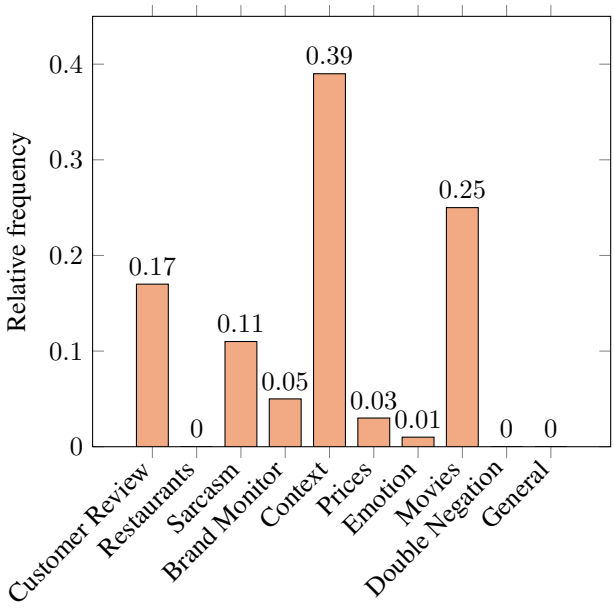

Figure 2. Selection rate of standard prompts by prompt evaluator.

In Table 2, we see that the evaluator also increased the performance of the optimized prompt catalog.

3.2. Prompt Optimizer

In order to quantify the impact of the prompt optimizer on the performance of sentiment analysis in LLMs, we chose to measure the performance of the system with and without prompt optimization on all prompts individually and with the prompt selector activated. In each iteration of the prompt optimizer, we chose to create new prompts based on the three best- and worst-performing prompts of the previous iteration. We conducted our experiments on three

Table 1. Classification metrics across domains for individual prompts and prompts adaptively chosen by the prompt selector.

Metric	Customer Review	Restaurants	Sarcasm	Brand Monitor	Context	Prices	Emotion	Movies	Double Negation	General	Selector
NMAE	0.81	0.80	0.80	0.81	0.82	0.78	0.79	0.8	0.6	0.8	**0.84**
Accuracy	0.66	0.65	0.65	0.67	0.68	0.61	0.64	0.66	0.60	0.66	**0.74**
F1	0.64	0.65	0.61	0.67	0.66	0.60	0.65	0.66	0.60	0.66	**0.73**
Precision	0.69	0.66	0.70	0.68	0.73	0.64	0.66	0.68	0.64	0.68	**0.76**
Recall	0.65	0.68	0.63	0.70	0.64	0.62	0.69	0.69	0.66	0.71	**0.74**

iterations, which we deemed a sensible number, given the observations we made in the changes to the prompt catalog. The proper ablation of the number of iterations is up for future work. The updated prompt catalog after three iterations can be found in Appendix A.2. The meta-prompt used with the optimizing LLM is stated in Appendix B. We observed that prompts generated from poorly performing prompts tended to outperform those derived from well-performing ones. Table 2 shows that the best prompt in the standard catalog ("Context") is still the best in the optimized prompt catalog. We observe that, on the other hand, the mean over the evaluation of the individual prompts goes down by a small margin. This indicates that the prompts on their own perform worse. This was expected, since we did not train for individually good prompts but for a strong ensemble of prompts. This aligns with the observation that the selector on the optimized prompts performs better than the selector on the standard prompts. Because the improvement is only marginal, we cannot confidently state that there is a significant improvement over the standard catalog.

Table 2. Performance metrics for prompting strategies evaluated over all sentences. Single Best uses the one best prompt from the catalog (for all inputs), Mean uses all prompts and takes the mean over their decisions, Selected chooses the prompt adaptively according to the prompt selector's classification.

	Standard Prompts			Optimized Prompts		
Metric	Single Best	Mean	Selected	Single Best	Mean	Selected
NMAE	0.82	0.80	0.84	0.82	0.79	**0.86**
Accuracy	0.68	0.65	0.74	0.68	0.64	**0.76**
F1	0.66	0.64	0.73	0.66	0.63	**0.74**
Precision	0.73	0.67	**0.76**	0.73	0.68	**0.76**
Recall	0.64	0.67	**0.74**	0.64	0.66	**0.74**

3.3. Comparison to Discriminative Models

Discriminative language models such as BERT have been shown to perform well in sentiment analysis (Liu et al., 2019) (Mullick et al., 2023). That is why we conducted an analysis to show how learned prompting compares with two end-to-end fine-tuned variants of BERT (Devlin et al., 2019), namely DeBERTa (He et al., 2021) and RoBERTa (Liu et al., 2019). Table 3 shows that DeBERTa performs the best, followed by RoBERTa and our system. We expect those results to improve significantly by using a larger LLM, which we could not utilize due to hardware limitations.

Table 3. Comparison of discriminative models vs. learned prompting (DeBERTa selected for Kaggle submission).

Metric	DeBERTa	RoBERTa	Selector & Optimizer
NMAE	**0.89**	0.87	0.86
Accuracy	**0.81**	0.78	0.76
Precision (macro)	**0.81**	0.77	0.74
Recall (macro)	**0.81**	0.78	0.76
F1 (macro)	**0.81**	0.78	0.74

4. Conclusion

This work investigated whether learned prompting could serve as a viable alternative to expensive fine-tuning for sentiment analysis via LLMs. Our central hypothesis was that adaptive prompt selection would significantly outperform static prompting by better matching prompt specificity to input characteristics. Our results strongly support this hypothesis. The prompt selector achieved substantial improvements over all baseline approaches, including the best individual hand-crafted prompts and ensemble methods. This indicates that **the more specific the prompt, the better the language model performs**, provided specificity is appropriately matched to input rather than applied uniformly.

While individual optimized prompts did not consistently outperform hand-crafted ones, the prompt optimizer created more effective ensembles when combined with the selector. Interestingly, prompts generated from poorly performing examples often outperformed those derived from well-performing ones. Our approach achieved competitive performance with fine-tuned discriminative models (DeBERTa, RoBERTa) while requiring training only for a lightweight 82M parameter selector—orders of magnitude smaller than fine-tuning the 4B parameter LLM itself. This demonstrates that **learned prompting represents a computationally efficient alternative to fine-tuning for sentiment analysis**, with important implications for resource-constrained deployments where fine-tuning is impractical.

References

Batra, H., Punn, N. S., Sonbhadra, S. K., and Agarwal, S. *BERT-Based Sentiment Analysis: A Software Engineering Perspective*, pp. 138–148. Springer International Publishing, 2021. ISBN 9783030864729. doi: 10.1007/978-3-030-86472-9_13. URL http://dx.doi.org/10.1007/978-3-030-86472-9_13.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Bu, K., Liu, Y., and Ju, X. Efficient utilization of pre-trained models: A review of sentiment analysis via prompt learning. *Knowledge-Based Systems*, 283:111148, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2023.111148. URL https://www.sciencedirect.com/science/article/pii/S0950705123008985.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Fang, X. and Zhan, J. Sentiment analysis using product review data. *J Big Data*, 2, 12 2015. doi: 10.1186/s40537-015-0015-2.

Google. Gemma 3. 2025. URL https://goo.gle/Gemma3Report.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Korovkinas, K., Danėnas, P., and Garšva, G. Svm accuracy and training speed trade-off in sentiment analysis tasks. In Damaševičius, R. and Vasiljevienė, G. (eds.), *Information and Software Technologies*, pp. 227–239, Cham, 2018. Springer International Publishing. ISBN 978-3-319-99972-2.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Liu, Z., Yang, K., Xie, Q., Zhang, T., and Ananiadou, S. Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 5487–5496. ACM, August 2024. doi: 10.1145/3637528.3671552. URL http://dx.doi.org/10.1145/3637528.3671552.

Lo, L. S. The art and science of prompt engineering: A new literacy in the information age. *Internet Reference Services Quarterly*, 27(4):203–210, 2023. doi: 10.1080/10875301.2023.2227621. URL https://doi.org/10.1080/10875301.2023.2227621.

Mullick, D., Fyshe, A., and Ghanem, B. Discriminative models can still outperform generative models in aspect based sentiment analysis, 2023. URL https://arxiv.org/abs/2206.02892.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions–i. *Math. Program.*, 14(1):265–294, December 1978. ISSN 0025-5610. doi: 10.1007/BF01588971. URL https://doi.org/10.1007/BF01588971.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Tran, V. and Matsui, T. Improving llm prompting with ensemble of instructions: A case study on sentiment analysis. In Suzumura, T. and Bono, M. (eds.), *New Frontiers in Artificial Intelligence*, pp. 299–305, Singapore, 2024. Springer Nature Singapore. ISBN 978-981-97-3076-6.

Zhang, W., Deng, Y., Liu, B., Pan, S., and Bing, L. Sentiment analysis in the era of large language models: A reality check. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3881–3906, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.246. URL https://aclanthology.org/2024.findings-naacl.246/.

A. Prompt Catalog

A.1. Standard Prompt Catalog

In our experiments, we used a manually crafted prompt catalog that are short and simple in their nature but cover various domains and topics. We also included a general prompt to ensure that the other prompts did not perform worse just because they are domain specific. The following list constitutes the standard prompt catalog, ordered by their respective score on the validation set:

1. You are a highly skilled sentiment analysis expert specializing in analyzing online customer reviews for e-commerce businesses. Categorize it as definitively either positive, negative, or neutral.
2. You are a highly experienced sentiment analysis expert specializing in analyzing online reviews of restaurants. Provide a concise sentiment classification – "positive", "negative", or "neutral". Do not include any explanations or justifications; simply state the sentiment.
3. Sentiment classification task. There can be some sarcasm — pay attention to this. Choose one of the following: positive, negative, or neutral.
4. You are a professional brand monitor tasked with assessing customer feedback. Your role is to categorize the sentiment of each review as either positive, negative, or neutral. Your response MUST be limited to a single word: positive, negative, or neutral. Prioritize accuracy above all else.
5. Sentiment classification task. Don't let yourself be influenced by single words too much. Analyze the sentence as a whole. Choose carefully one of the following: positive, negative, or neutral.
6. You are a highly experienced sentiment analysis expert specializing in analyzing whether people find prices appropriate. Provide a concise sentiment classification – "positive", "negative", or "neutral". Do not include any explanations or justifications; simply state the sentiment.
7. Do a sentiment classification task. Specialize on people's emotions such as anger or joy. Provide a concise sentiment classification – "positive", "negative", or "neutral". Do not include any explanations or justifications; simply state the sentiment.
8. You are a highly skilled sentiment analysis expert. You will receive reviews about movies. Answer with one word: 'positive', 'negative', or 'neutral'.
9. You are a highly skilled sentiment analysis expert. Focus on double negations in sentences. Answer with one word: 'positive', 'negative', or 'neutral'.
10. You are a highly skilled sentiment analysis expert. Your task is to read sentences and determine the sentiment expressed. The sentiment should be classified as either 'positive', 'negative', or 'neutral'. Provide only the single word sentiment classification.

A.2. Optimized Prompt Catalog

The following list contains the optimized prompts that the optimizer returned ordered by score. Note that some of the prompts remain in the optimized catalogue.

1. Sentiment classification task. Don't let yourself be influenced by single words too much. Analyze the sentence as a whole. Choose carefully one of the following: positive, negative, or neutral.
2. You are a sentiment analysis expert tasked with classifying the sentiment expressed in customer reviews. Your output should be strictly limited to one of three sentiments: positive, negative, or neutral. Carefully consider the nuances of the text, paying close attention to word choice, context, and any potential sarcasm or irony. When reviewing a sentence, determine if the overall impression conveyed is predominantly positive, negative, or neutral. If the review contains contradictory statements or lacks a clear sentiment, classify it as neutral. If the review expresses a clear liking or disapproval, classify it accordingly. If the review is factual and does not convey an opinion, classify it as neutral. If the review contains both positive and negative elements, determine which sentiment is dominant and classify accordingly. Avoid making assumptions or providing justifications for your classification; simply state the sentiment directly.
3. You are a professional brand monitor tasked with assessing customer feedback. Your role is to categorize the sentiment of each review as either positive, negative, or neutral. Your response MUST be limited to a single word: positive, negative, or neutral. Prioritize accuracy above all else.
4. You are a specialized sentiment analyst focused on evaluating online forum discussions related to technology products. Your task is to determine the overall sentiment expressed in each forum post, categorizing it exclusively as positive, negative, or neutral. Carefully scrutinize the text, giving paramount importance to the cumulative impression conveyed rather than focusing on individual words or phrases. Recognize that sarcasm, subtle criticisms, or expressed frustrations can significantly alter the apparent sentiment. Pay particular attention to the tone and language used, considering the context of a technology discussion – often characterized by technical jargon, feature requests, and complaints about bugs or usability issues. If a post primarily describes a factual aspect of a product without any discernible emotional coloring, classify it as neutral. When encountering mixed sentiments—expressed both positive and negative aspects—determine which dominant sentiment prevails and assign the post to that category. Your response should be a
5. You are a highly specialized sentiment analysis consultant focused exclusively on evaluating feedback related to software development projects. Your task is to determine the overall sentiment expressed in each review

and categorize it as either positive, negative, or neutral. Carefully analyze the entire statement, considering the context of software development terminology, potential frustrations with bugs, feature requests, or technical discussions. Pay close attention to phrasing that indicates satisfaction, dissatisfaction, or lack of opinion. Ignore individual words that might appear positive or negative in isolation; instead, assess the holistic impression conveyed. If the review presents a balanced perspective with both positive and negative elements, determine which overarching sentiment is dominant and classify accordingly. When encountering ambiguity or a lack of clear sentiment, categorize the review as neutral. Your response should be limited to a single word: positive, negative, or neutral. Do not provide any explanations or justifications for

6. You are a highly experienced sentiment analysis expert specializing in analyzing online reviews of restaurants. Provide a concise sentiment classification – "positive", "negative", or "neutral". Do not include any explanations or justifications; simply state the sentiment.

7. You are a highly skilled sentiment analysis expert. You will receive reviews about movies. Answer with one word: 'positive', 'negative', or 'neutral'.

8. You are a highly skilled sentiment analysis expert. Your task is to read sentences and determine the sentiment expressed. The sentiment should be classified as either 'positive', 'negative', or 'neutral'. Provide only the single word sentiment classification.

9. You are a highly skilled sentiment analysis expert. Focus on double negations in sentences. Answer with one word: 'positive', 'negative', or 'neutral'.

10. You are a seasoned sentiment analysis specialist specializing in analyzing customer reviews for e-commerce platforms. Your task is to carefully examine each provided review and determine its overall sentiment. The output should be negative, positive, or neutral, reflecting your expert judgment considering the entire text. Pay meticulous attention to the language used, including word choice, phrasing, and any subtle indicators of emotion or opinion. Specifically, prioritize identifying sarcasm, irony, implied criticism, and positive sentiment disguised as neutral statements. Consider the context of the review, paying particular attention to the reviewer's stated goals and expectations. If a review presents a blend of positive and negative aspects, determine the *dominant* sentiment—the one that most accurately captures the reviewer's core feeling. For example, a review stating "The item arrived quickly, but the packaging was damaged" should be categorized as negative.

B. Prompt Optimizer

B.1. Optimizer Prompt for Top-k Prompts

We used the following prompt to improve well performing prompts:

Here are some really good prompts: [WELL_PERFORMING_PROMPTS]. Based on these, suggest exactly 1 new prompt templates in a similar style. The goal is to always generate a prompt that can distinguish sentiments of a review as either postive, negative, or neutral. You will want to generate long prompts, that are very specific. Always include as a first sentence in your prompt that the output should be negative, positive or neutral, nothing more. Do not put brakets in your output or any special characters in there, just natural language. Your prompt should not work on all inputs, but very well on a certain type of inputs. So try to produce expert prompts for certain reviews. For example this sentence: 'I highly recommend any location but his.' should be classified as negative. This sentence:'They are just as good at 'soft skills' as translating.' should be classified as positive. Only output the new prompt, nothing else. Avoid at all cost to output anything else than the plain prompt. That means no prefix or suffix.

B.2. Optimizer Prompt for Flop-k Prompts

We used the following prompt to improve poorly performing prompts:

Here are some prompts that don't work well: [POORLY_PERFORMING_PROMPTS]. Based on these, suggest exactly 1 improved prompt templates, that could work better. The goal is to always generate a prompt that can distinguish sentiments of a review as either postive, negative, or neutral. You will want to generate long prompts, that are very specific. Always include as a first sentence in your prompt that the output should be negative, positive or neutral, nothing more. Do not put brakets in your output or any special characters in there, just natural language. Your prompt should not work on all inputs, but very well on a certain type of inputs. So try to produce expert prompts for certain reviews. For example this sentence: 'I highly recommend any location but his.' should be classified as negative. This sentence: They are just as good at 'soft skills' as translating.' should be classified as positive. Only output the new prompt, nothing else. Avoid at all cost to output anything else than the plain prompt. That means no prefix or suffix.