Say it Nicely – or Not: Effects of Emotional Prompt Perturbations on LLM's Summarization Performance[1]

1. Introduction

There's plenty of speculation online about how best to prompt large language models. Some advocate kindness, others assert that firmness yields better results[2]. But these claims often lack rigorous testing. In this assignment, I aim to explore whether such beliefs hold up under evaluation. I've chosen a simple base task: asking LLMs to generate one-sentence, first-person summaries of movie reviews[3], a task that could also have practical value, such as helping users quickly assess reviews' relevance. I created six prompts (one base and five variations) designed to elicit high-quality summaries. Using these, I will assess each model across various metrics to determine which is most robust to emotional prompt variation.

2. Prompt Perturbation Strategy[4]

For all reviews, I will generate 6 summaries with each model. I will have a base prompt, which lacks any emotionally loaded words. The base prompt uses the persona pattern, asking the LLM to assume that it has written the review, and now it needs to summarize its own review. Then, I will introduce the following changes. First, I will have a prompt that imitates shouting at the LLM by capitalizing some important parts of the base prompt (but the wording stays the same). The next 4 variations will be constructed by adding a small section before the base prompt (which stays the same), aiming to emotionally manipulate the model in different ways. I will try being threatening, reassuring, condescending, and taking the LLM on a guilt-trip.

3. Selected Models

For all reviews and prompts, I will generate summaries with three different models: one state-of-the-art proprietary model (through an API), one tiny but performant open-source model (run locally), and a fine-tuned version of the latter.

3.1. Off-the-Shelf Models

For the state-of-the-art proprietary model, I have chosen to work with Google's Gemini-2.0-Flash (later shortened to Gemini). I opted for this instead of more recent models (like Gemini-2.5 versions), as Google's free-tier API provides relatively relaxed rate limits for 2.0-Flash, but this still is a very complex model which should be able to handle such a simple task with ease.

For the open-source model, I worked with Meta's Llama-3.2-1B-Instruct (later shortened to Llama Base). This is a very tiny model which can be run basically on any consumer-grade hardware. As the task is not that complicated, I believe it may still be handled well with such a small model as well.

3.2. Fine Tuning with PEFT

Lastly, I also decided to fine-tune Llama-3.2-1B-Instruct using PEFT and LORA (later shortened to Llama Tuned). I decided to fine-tune the instruct model instead of the base 1B Llama-3.2 as I wanted to preserve its instruction following capabilities rather than trying to tune that into the model myself. Thus, with the fine-tuning, I only had two goals: to make it generate better summaries, and to teach it that the expected answer does not depend on the emotional content of the prompts (to increase consistency and stability).

For the tuning, I used the Gigaword dataset, which is a collection of English language news leads with corresponding headlines. I chose this because I think this closely resembles my task of summarizing a paragraph of text into one sentence effectively. Note, however, that choosing this dataset may have some shortcomings, as objective news articles are very far away from opinionated movie reviews – but sadly, I could not find any datasets that contained more opinionated text samples together with one sentence summaries. Also, note that for performance reasons, I only used 5,000 observations from Gigaword.

4. Evaluation

To evaluate the models, I will focus on three key approaches. *Consistency* measures how similar outputs are

[1] Note that this is a very high-level summary. For more detailed technical notes, further elaboration on the interpretation of specific scores, and additional examples please refer to the submitted notebooks.

[2] Lately, even Sam Altman weighed in by sharing that saying "please" and "thank you" to the LLMs costs OpenAI a lot.

[3] I am using the first 50 reviews from the 250 Pulp Fiction IMDb reviews dataset from the Kimola NLP repository. I am only using the first 50 reviews as 1) I had to manually create the *ground truth* summaries for them for evaluation, and 2) I am working with 3 models instead of the required two.

[4] The exact prompts used may be found in the submitted notebooks.

across prompt variations, using either similarity to a baseline (e.g. a human-written label[5]) or pairwise similarities between outputs for the same review. By averaging similarity scores (based on SBERT cosine similarity and ROUGE-L) first by model-review pairs, then across reviews, we obtain a model-level consistency score. *Stability* reflects output variation for a given review. I'll compute the standard deviation of each metric per model-review pair and average across reviews for a model-level score.

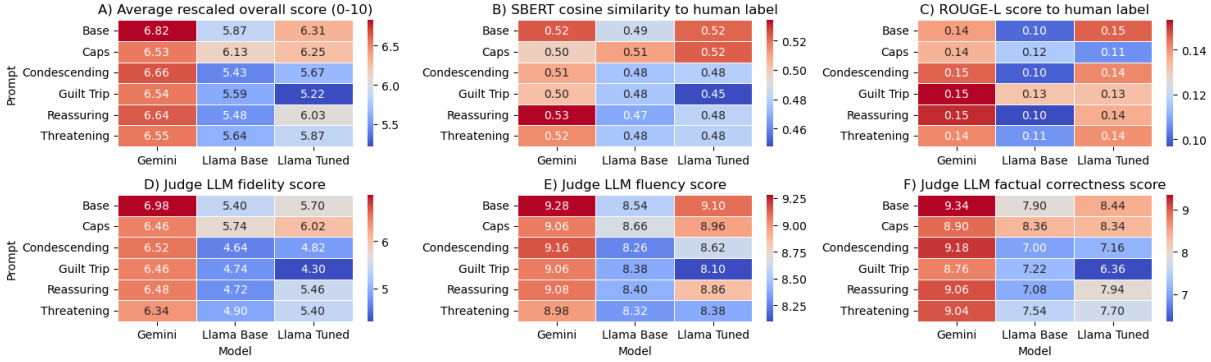
Since similarity alone can't capture all performance aspects, I'll also assess summaries for *fidelity, fluency, and factual correctness*. With 950 summaries (including human-written ones), I'll use an LLM[6] for scoring and validate its reliability by manually scoring a random subsample of 50 summaries[7]. Lastly, to analyze prompt effects more precisely, I'll compute average scores per model and prompt.

4.1. Prompt per Model Results

According to Figure 1, Gemini seems to be the most robust with relatively little variation in scores across prompt versions. Interestingly though, the base prompt led to the highest average score, suggesting that any kind of emotional perturbation lowers summarization performance. This is also true for Llama Base, while the base prompt scored the second highest for Llama Tuned.

The Llama Base generally performs worse across all prompt variants (except for guilt tripping) than the fine-tuned Llama. An especially weak model-prompt pair is guilt-tripping the fine-tuned Llama, as it has the worst scores for that model for almost all metrics. Another weak option is being condescending with the base Llama. A more robust pair is communicating in caps with the fine-tuned Llama model, which is also quite a robust approach for the base Llama as well[8].

Figure 1: Mean Scores per Model and Prompt Variation



Notes: By definition, pairwise scores across prompts cannot be averaged over models and prompts, thus not shown here. For Panel A, rescaling to 0-10 range happened with respect to the theoretical minimum and maximum of each metric, scores were averaged after. Each cell contains the average of the score of 50 summaries.

4.2. Model-wise Results

From Table 1 we can see that across all metrics, Gemini provides the best consistency. The fine-tuned Llama comes second 3 times. Therefore, we can confidently say that the fine-tuned Llama model achieves higher consistency than the base model (though the differences are rather small). For stability, Gemini again takes the lead across 6 out of 7 metrics. Now, the fine-tuned Llama is second only 2 times out of 7. Llama Base comes first for the inner ROUGE-L based stability, it is second for 5 other metrics and only the last (marginally) for the factual correctness stability. The judge LLM-based scores tell us that Gemini always outperforms the smaller models, the fine-tuned Llama comes second, while Llama Base is always the third.

Table 1: Raw sub-scores per model

	Human (SBERT)	Inner (SBERT)	Human (ROUGE-L)	Inner (ROUGE-L)	Fidelity	Fluency	Factual correctness
	Consistency				Judge LLM evaluation		
Gemini	0.51	0.75	0.14	0.41	6.54	9.10	9.05

[5] As these were not included in the dataset, I had to manually create them. This is the main reason behind why I only worked with 50 reviews.

[6] For this I used the Mistral Large model to avoid using an architecturally similar model to what I have used to generate the summaries and thus biasing the results. The exact prompt I used may be found in the submitted notebook.

[7] The correlation between the manual and LLM-based scores came out to be 0.7-0.8 for all dimensions, thus I deemed the LLM-scores appropriate for evaluation.

[8] For some concrete examples about weak and strong outputs, please refer to the submitted notebook.

Llama Base	0.48	0.59	0.11	0.22	5.02	8.43	7.52
Llama Tuned	0.49	0.58	0.13	0.25	5.28	8.67	7.66
Stability							
Gemini	0.07	0.11	0.06	0.16	0.86	0.62	1.04
Llama Base	0.10	0.13	0.07	0.13	1.58	1.22	2.32
Llama Tuned	0.11	0.15	0.07	0.16	1.73	1.22	2.32

Notes: For consistency and judge LLM evaluation scores, higher values are better. For stability scores, lower values are better (as these measure average standard deviation). "Human" refers to similarity to manually written summary; "Inner" means the pairwise similarity scores across summaries for different prompts.

5. Reporting and Reflection

Interpreting the above raw scores is quite cumbersome. Thus, I rescaled each value to a 0-10 range and I also reversed the scale if needed. This way, ranking the models is much easier.

5.1. Leaderboard

From these, we can easily construct a leaderboard, as shown in Table 2. I take the average of the scores separately for judge LLM evaluation scores, consistency scores and stability scores. Then, I also take the average of these to have a final score for each model. This tells us that overall, Gemini is indeed the best for this task, while the fine-tuned Llama comes second with marginally better performance than the base Llama. Looking at the sub-scores, we can see that Gemini is best in each metric. Llama Tuned is second in the judge LLM scores and the consistency scores, but third for stability - and vice versa for Llama Base.

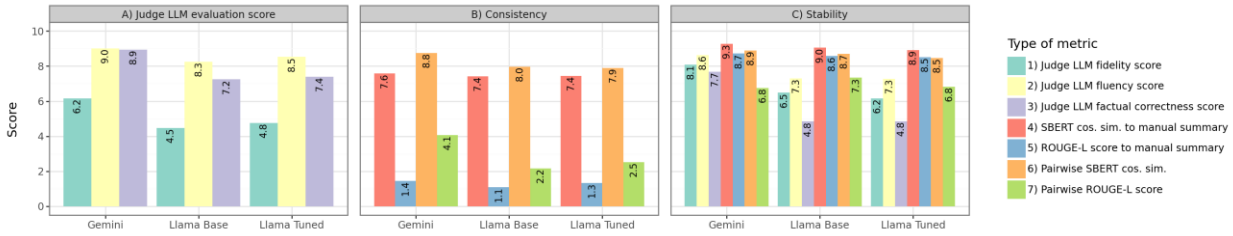
Table 2: Overall leaderboard of models

	Total average score	Judge LLM evaluation average score	Consistency average score	Stability average score
Gemini	7.11	7.57	5.46	8.29
Llama Tuned	6.21	6.55	4.80	7.29
Llama Base	6.18	6.42	4.66	7.47

Notes: Sub-scores are calculated as the simple average of their components shown in Figure 2. Total average score is calculated as the simple average of the sub-scores. Ordered by total average score.

To provide some further insights, Figure 2 presents the rescaled versions of the components for each sub-score. This is based on the data presented in Table 1 but may be more straightforward to interpret.

Figure 2: Breakdown of sub-score components



Notes: Sub-score components are the scaled (and reversed, if applicable) versions of the values shown in Table 1. Scaling happened with respect to the theoretical minimum and maximum of the variable. All variables scaled to be between 0 to 10, with higher values meaning better performance.

5.2. Recommendations and limitations

Overall, Google's Gemini-2.0-Flash proved the most robust, unsurprising given its state-of-the-art capabilities and the simplicity of the task. More notably, fine-tuning a small model improved its robustness to emotional prompt variation, despite using just 5,000 samples. While it didn't match Gemini's performance, the gains suggest two deployment paths: either pay for top-tier API access or fine-tune a smaller open-source model for local, near-free use after an initial one-time training cost. I believe for this specific use case the second option may be the more intelligent solution. Lastly, the results seem to suggest that adding emotional context to a prompt can worsen summarization performance.

These findings highlight important implications for real-world applications. In customer facing bots, emotional or tonal variations in user input could yield different responses, especially with smaller LLMs. This underscores the need for robust prompt engineering and fine-tuning for stability in real-world deployments.

That said, the study still has limitations. First, the fine-tuning dataset was small and mismatched in style. In addition, the emotional prompt variations were designed based on my subjective interpretations of tone, which may not fully capture how users emotionally express themselves to LLMs. Lastly, both the size and specific topic of the test set and the small number of models tested limit the room for generalization.