Fynd AI Intern – Take Home Assessment

Task 1: Rating Prediction via Prompting

Harsh Maurya
AI Intern Applicant

1 Objective

The objective of Task 1 was to evaluate how different prompt engineering strategies affect a large language model's (LLM) ability to predict Yelp review star ratings (1–5) while producing structured JSON output. The focus of this task was on prompt design and evaluation rather than model training.

2 Dataset

The Yelp Reviews dataset was sourced from Kaggle. For efficiency and cost considerations, a random subset of 200 reviews was sampled from the dataset. Each review consisted of textual feedback along with an actual star rating provided by the user, which was used as ground truth for evaluation.

3 LLM Setup

All experiments were conducted using a free, open-source large language model accessed via OpenRouter. The same model and inference pipeline were used across all prompt versions to ensure fair comparison and consistency in evaluation.

4 Prompt Design and Iterations

Three different prompt versions were designed and tested.

4.1 Prompt Version 1 : Baseline

This prompt instructed the LLM to read a Yelp review and predict a star rating between 1 and 5, requesting the output in JSON format. No additional constraints or rating guidelines were provided. This prompt served as a baseline to observe the model's default behavior.

4.2 Prompt Version 2 : Criteria-Guided Prompt

The second prompt introduced explicit rating guidelines defining what each star level represents, along with stricter instructions to return only valid JSON. The goal was to reduce ambiguity and improve structured output reliability.

4.3 Prompt Version 3 : Step-Based Prompt

The third prompt encouraged internal step-by-step reasoning before producing the final answer while still enforcing a JSON-only output format. This approach aimed to improve reasoning quality and consistency.

5 Evaluation Metrics

Each prompt was evaluated using the following metrics:

- **Accuracy**: Proportion of reviews where the predicted rating exactly matched the actual rating.
- **JSON Validity Rate**: Percentage of responses that were valid and parseable JSON.
- **Accuracy (Valid JSON Only)**: Accuracy calculated only on outputs that produced valid JSON.

6 Results

Prompt Version	Accuracy	JSON Validity	Accuracy (Valid JSON Only)
V1 (Baseline)	0.01	0.015	0.67
V2 (Guided)	**0.055**	**0.085**	0.65
V3 (Step-based)	0.035	0.065	0.54

Table 1: Comparison of prompt performance across evaluation metrics

7 Discussion

The baseline prompt (V1) resulted in very low JSON validity, which significantly impacted overall accuracy despite reasonable performance when valid JSON outputs were produced.

Prompt Version 2 demonstrated the best overall performance by achieving the highest JSON validity and overall accuracy. The inclusion of explicit rating guidelines and stricter output constraints improved reliability and reduced ambiguity.

Prompt Version 3, which introduced step-based reasoning, did not further improve performance and slightly reduced consistency. This suggests that for smaller open-source models, excessive instruction may negatively affect structured output compliance.

8 Key Takeaways

- Prompt clarity and output constraints significantly impact structured output reliability.
- Explicit rating guidelines improve both accuracy and consistency.
- Increasing prompt complexity does not always yield better results.

- JSON validity is a critical metric when evaluating LLMs for automated systems.

9 Conclusion

Among the three approaches evaluated, the criteria-guided prompt (Prompt Version 2) offered the best balance between accuracy and structured output reliability. This experiment highlights the importance of prompt engineering and careful evaluation when using LLMs for classification tasks.