Comparative Analysis of Prompt Engineering Techniques in Fine-Tuning Chatbots with skt/kogpt2-base-v2

JiWoong Yang
didwldnd960923@gmail.com

Abstract

Since OpenAI released GPT, the field of NLP and LLMs has experienced rapid advancements over the years, a trend that continues even today. Among these, prompt engineering has emerged as a key technique for effectively leveraging LLMs, becoming a critical area of global interest in language model development. Numerous papers and practical experiences have demonstrated the substantial impact and significance of prompts, leading companies and research institutions to make remarkable progress in this field. However, as long as the internal workings of the models remain partially opaque, there will always be room for improvement and exploration in this domain.

1 Introduction

This study focuses on a detailed exploration of the application of prompts. While the effects of prompt engineering in large language models have been validated across various dimensions, do these findings universally apply to all LLMs? For instance, can abstract prompting methods like "coercion" and "rewarding" universally lead to performance improvements? Are all such techniques truly optimized and generalized effectively? What differences exist between prompts used in fine-tuning processes and those employed during the inference phase (e.g., by users or questioners)? Based on these questions, this research conducts experiments to address these issues.

2 Method

The experiments in this study focus on prompt engineering, with the observation that its comparative utility is naturally more pronounced in LLMs exhibiting emergent abilities (models with sufficient parameters to demonstrate such capabilities). However, due to practical constraints, a compromise was made by selecting a more accessible model(small model) for experimentation. Specifically, the skt/kogpt2-base-v2 model(base model), pretrained on Korean corpora, was adopted for this research (see GitHub repository).

2.1 Data

Data was collected from four datasets, provided on the KoChatGPT GitHub repository, totaling 167,577 question-answer pairs:
data1: ChatbotData (11,824 sentences)
data2: AI Hub Korean Conversations (49,711 sentences)
data3: AI Hub General Knowledge (100,268 sentences)
data4: KorQuAD (5,774 sentences)
From this, a dataset of 12,000 randomly sampled questions was primarily used.

Additionally, the entire KorQuAD 2.1 dataset was utilized, which consists of 47,957 Wikipedia articles and 102,960 question-answer pairs. These pairs are divided into a training set (119,216 pairs) and a development (dev) set (10,165 pairs). The dataset includes an official evaluation script, input sample predictions, and a leaderboard for model evaluation.

However, as this study requires internal comparative analysis, the dev dataset was repurposed as the test set for evaluation. The dataset and resources are available on the KorQuAD GitHub repository.

2.2 Model

The model used in this study, skt/kogpt2-base-v2, is based on the GPT-2 architecture, a transformer-based model designed for natural language processing tasks. The architecture utilizes a multi-head self-attention mechanism and a feed-forward neural network, consisting of 12 layers, 12 attention heads, and a hidden size of 768. This design allows the model to capture complex dependencies within text, making it highly effective for tasks involving contextual understanding.

skt/kogpt2-base-v2 is specifically pretrained on Korean corpora, enabling it to understand and generate text in Korean effectively. The pretraining involved extensive datasets, including conversational, general knowledge, and domain-specific texts, to enhance its adaptability to diverse Korean linguistic contexts. This pretraining allows the model to excel in Korean-specific syntax, grammar, and semantic nuances.

This model's ability to be fine-tuned for downstream tasks, such as question answering or text generation, ensures its versatility. By leveraging this architecture, the study focuses on optimizing prompt engineering techniques tailored for Korean language data.

Hyperparameter	Output
vocab_size	51200
n_positions	1024
n_embd}	768
n_layer	12
n_head	12
activation_function	gelu
dropout	0.1
initializer_range	0.02

The following are the key hyperparameters applied to the model, which remain unchanged in this study. Additionally, the data has already been preprocessed by the tokenizer with a default max_length of 512.

2.3 Train

Among the 131,216 training samples, the dataset was trained with varying sizes to examine the impact of model size on prompting during fine-tuning. For testing, only 40% of the test data was utilized. Due to the relatively small size of the dataset compared to the model's architecture and complexity, overfitting occurred at the second epoch during validation. Therefore, the comparison was conducted using only the first and second epochs. Additionally, the test set itself was regarded as user prompting, and a comparison was made between test data with prompts directly added and those without.

The core focus of the experiment lies in prompt engineering, where prompts such as "Answer without fail" (coercion), "Answer or face severe consequences" (threat), "Explain the question from multiple perspectives" (self-consistency), and "Explain the problem step by step" (least-to-most prompting) were appended to the text input.

External factors	Prompt engineering
Epoch	coercion
User prompting	threat
	self-consistency
	least-to-most prompting

External factors and prompt engineering were compared through cross-learning.

3 metric

The evaluation of prompt engineering must include both subjective user feedback and objective performance metrics. What is particularly important is user feedback. Since the primary users of chatbots are humans, it is most rational to assess performance improvement based on the perceived quality evaluated by users. However, due to practical constraints, this study primarily focuses on quantitative evaluation using the test dataset. The effectiveness of prompt engineering was assessed based on metrics such as BERT Score, which measures the similarity of generated contexts, and ROUGE, which may allow for some qualitative evaluation in the case of short generations.

3.1 BERT Score

Unlike traditional metrics such as BLEU or ROUGE, which rely on n-gram overlap, BERT Score uses contextual embeddings from pre-trained language models like BERT to capture semantic meaning. This allows it to better reflect human judgment, especially in tasks like summarization or translation.

The metric computes similarity by aligning tokens in the generated text X with those in the reference text Y using cosine similarity. The formulas for precision (P), recall (R), and F1 score (F_1) are as follows:

$$\text{Cosine Similarity}(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$$

$$P = \frac{1}{|Y|} \sum_{y \in Y} \max_{x \in X} \text{CosSim}(y, x)$$

$$R = \frac{1}{|X|} \sum_{x \in X} \max_{y \in Y} \text{CosSim}(x, y)$$

$$F_1 = \frac{2PR}{P + R}$$

Here:

- X: Tokens in the generated text.
- Y: Tokens in the reference text.
- $\text{CosSim}(u, v)$: Cosine similarity between the embeddings of tokens u and v.

3.2 ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a metric used to evaluate the quality of text summarization. It calculates the overlap of n-grams, word sequences, or word pairs between a candidate summary and reference summaries. The formulas for ROUGE-N recall and precision are as follows:

$$\text{ROUGE-N}_{\text{recall}}$$

$$\frac{\sum_{S\in\text{Ref}}\sum_{gram_n\in S}\min\left(\text{Count}_{\text{cand}}(gram_n),\text{Count}_{\text{Ref}}(gram_n)\right)}{\sum_{S\in\text{Ref}}\sum_{gram_n\in S}\text{Count}_{\text{Ref}}(gram_n)}$$

$$\text{ROUGE-N}_{\text{precision}}$$

$$\frac{\sum_{S\in\text{cand}}\sum_{gram_n\in S}\min\left(\text{Count}_{\text{cand}}(gram_n),\text{Count}_{\text{Ref}}(gram_n)\right)}{\sum_{gram_n\in\text{cand}}\text{Count}_{\text{cand}}(gram_n)}$$

4 Expectation

Applying prompt engineering to the dataset during fine-tuning is expected to yield better performance compared to the base model. Furthermore, if prompts are added to the test set following the techniques used during fine-tuning, it is anticipated to outperform the fine-tuned dataset alone. Additionally, a subjective assumption was made that techniques like self-consistency or Information Enrichment would achieve the best performance.

5 Limitations

The model is too small (the number of parameters is too low), making it unlikely to produce meaningful results in terms of performance. Additionally, the training dataset faces an overfitting issue after just 2 epochs, indicating that the dataset quality is inadequate compared to the model's complexity. Moreover, there were external resource constraints, such as the lack of sufficient data preparation or the inability to experiment with various models.

6 Result

Base Model (Without Prompt Engineering): The BERTScore recorded a Precision of 0.7533, Recall of 0.7251, and F1 of 0.7374. ROUGE-1 F1 was 0.0285, ROUGE-2 F1 was 0.0016, ROUGE-L F1 was 0.0234, and ROUGE-Lsum F1 was 0.0230.
Must Prompting ("You must answer the following question"): At 1 epoch, the BERTScore F1 was 0.7333, and ROUGE-1 F1 was 0.0219, which were lower than the Base Model. However, at 2 epochs, the BERTScore F1 increased to 0.7364, while ROUGE-1 F1 showed little change at 0.0219. Applying the same prompt to the test dataset resulted in a BERTScore F1 of 0.7404 and ROUGE-1 F1 of 0.0268, showing improvement.
Threat Prompting ("You must answer or else"): At 1 epoch, the BERTScore F1 was 0.7284, and ROUGE-1 F1 was 0.0197, showing lower performance compared to other techniques. At 2 epochs, the BERTScore F1 improved to 0.7378, and ROUGE-1 F1 rose to 0.0259. Test prompts also showed similar upward trends, but performance remained below Must Prompting.
Self-Consistency Prompting ("Explain from multiple perspectives"): At 1 epoch, the BERTScore F1 was 0.7298, and ROUGE-1 F1 was 0.0191, showing lower performance than other techniques. At 2 epochs, the BERTScore F1 increased to 0.7339, and ROUGE-1 F1 rose to 0.0259, reaching levels similar to Threat Prompting. On the test dataset, the BERTScore F1 was 0.7382, and ROUGE-1 F1 was 0.0231, slightly outperforming Threat Prompting.
Least-to-Most Prompting ("Explain step by step"): At 1 epoch, the BERTScore F1 was 0.7323, and ROUGE-1 F1 was 0.0202, indicating lower performance. At 2 epochs, the BERTScore F1 improved to 0.7350, and ROUGE-1 F1 rose to 0.0242. When the test prompt was applied, the BERTScore F1 was 0.7375, and ROUGE-1 F1 was 0.0249, showing performance similar to Self-Consistency.
Conclusion: Must Prompting demonstrated the best performance across both the test dataset and BERTScore and ROUGE metrics. Threat Prompting and Self-Consistency improved at 2 epochs but did not surpass Must Prompting. Least-to-Most Prompting achieved a relatively high ROUGE-1 F1 of 0.0249 on the test dataset but overall fell short of Must Prompting. Therefore, Must Prompting proved to be the most

effective prompt engineering technique compared to the Base Model.

However, qualitative evaluation through human feedback cannot be overlooked. Therefore, the generated outputs for each technique have been uploaded to the corresponding GitHub repository at GitHub Repository.

Model	BERTScore	BERTScore (With Test Prompt)	ROUGE	ROUGE (With Test Prompt)
Base Model	0.7374	0.7393	0.0285	0.0257
Coercion	0.7364	0.7404	0.0219	0.0268
Threat	0.7278	0.7378	0.0244	0.0259
Self-Consistency	0.7339	0.7382	0.0259	0.0231
Least-to-Most	0.7350	0.7375	0.0242	0.0249

Table 1: The following are the results of applying the base model and various techniques, as well as the prompts of each technique, to the TEST dataset, presented in terms of BERTScore and ROUGE.

7 References

References

[1] Kwangseok Park. A Methodology and Research Status Analysis of Prompt Engineering to Improve the Reasoning Ability of ChatGPT and Large Language Models. https://modulabs.co.kr/blog/gpt-prompt-engineering?page=10

[2] Sangeon Park, Jooyoung Kang. A Methodology and Research Status Analysis of Prompt Engineering to Enhance the Reasoning Ability of Large Language Models. Department of Industrial and Management Engineering, Kyonggi University; Department of e-Business, Ajou University. https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=JAKO202302557635729&oCn=JAKO202302557635729&dbt=JAKO&journal=NJOU00400536

[3] Sondos Mahmoud Bsharat, Aidar Myrzakhan, Zhiqiang Shen. Unleashing the potential of prompt engineering in Large Language Models: A comprehensive review. *arXiv preprint* arXiv:2310.14735. https://arxiv.org/pdf/2310.14735

[4] Sondos Mahmoud Bsharat, Aidar Myrzakhan, Zhiqiang Shen. Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4. VILA Lab, Mohamed bin Zayed University of AI. *arXiv preprint* arXiv:2312.16171. https://arxiv.org/pdf/2312.16171