

Prompt Engineering Assignment

Name: Kritika

Submission Date: March 26, 2025

Email: kritikabansal386@gmail.com

Introduction

Artificial Intelligence has always fascinated me because of its power to solve real-world problems and automate complex tasks. As a B.Tech student specializing in Artificial Intelligence, I constantly look for opportunities to apply what I learn in practical ways that go beyond textbooks.

For this internship assignment, I was asked to complete three tasks focused on prompt engineering for a chatbot that assists startup founders. Rather than using an online tool like ChatGPT or Hugging Chat, I challenged myself to build my own AI chatbot completely offline using open-source technology.

I used the **Mistral-7B-Instruct** model in .gguf format with the **llama-cpp-python** library to run the model locally on my system. This gave me complete control over the chatbot pipeline and allowed me to deeply explore how prompts affect the behaviour and quality of large language models. This hands-on approach not only strengthened my understanding of prompt engineering, but also gave me valuable experience in working with local LLMs and deploying AI models in resource-efficient ways.

Task 1: Create 3 Unique Fundraising Prompts

Prompt:

Suggest 3 creative and unique prompt ideas for a chatbot that helps startup founders with fundraising advice and strategies.

Response (Generated by Mistral-7B-Instruct):

The Chatbot Fundraiser:

This chatbot would provide personalized fundraising advice and strategies based on the startup's industry, business model, and funding goals. It would also provide access to a network of investors and offer guidance on how to pitch to them effectively.

The Investor Finder:

This chatbot would help startup founders identify potential investors by analyzing their business plan, market opportunity, and funding requirements. It would also provide customized messaging templates for pitching to investors and track progress toward funding goals.

The Fundraising Roadmap:

This chatbot would provide a step-by-step guide for startup founders on how to raise funds. It would offer a personalized roadmap based on the startup's stage of development, funding needs, and target investors. The chatbot would also offer resources and advice on creating a compelling pitch, preparing for investor meetings, and closing funding deals.

Task 2: Improve a Prompt

Objective:

The task was to improve a generic prompt to make it more specific, result-oriented, and useful for an AI chatbot response.

Original Prompt:

"Tell me about digital marketing."

Improved Prompt (Generated by Mistral-7B-Instruct):

"Provide a detailed explanation of email marketing campaigns, focusing on effective subject lines, personalization, and segmentation."

Explanation:

The improved prompt is more targeted and directs the AI to focus on a specific area within digital marketing. This helps generate a more structured and actionable response, which is especially useful for users looking for strategic insights. It showcases how even a small change in prompt wording can lead to significantly better outputs from a language model.

Task 3: Draft a Welcome Message

Objective:

Write a short and friendly welcome message for a chatbot named "**Startup GPT**" that helps startup founders with fundraising strategies. The message should encourage users to engage and ask questions.

Prompt Used:

"Write a friendly and professional welcome message for a chatbot called 'Startup GPT' that helps founders with fundraising strategies. The message should encourage users to ask questions."

Response (Generated by Mistral-7B-Instruct):

Hello and welcome to Startup GPT, your friendly and professional fundraising companion! I'm here to help you navigate the exciting and sometimes challenging world of fundraising for your startup. Whether you're just starting out or looking to take your fundraising efforts to the next level, I'm here to provide you with valuable insights, strategies, and tips to help you succeed. Don't hesitate to ask me any questions you may have about fundraising, venture capital, angel investors, crowdfunding, and more. I'm here to listen and provide personalized advice tailored to your unique needs and goals. So, what can I help you with today? Remember, the only stupid question is the one you didn't ask. Let's get started!

Explanation:

This message sets a friendly, helpful tone while clearly stating the chatbot's purpose. It also lowers barriers for engagement by encouraging users to ask questions, making it an effective onboarding message for a startup-oriented AI assistant.

Screenshots

The following screenshots show the terminal-based responses from my local chatbot powered by the Mistral-7B-Instruct model. These demonstrate the successful completion of all three prompt engineering tasks using an offline AI setup.

- Mistral Model Loaded in Terminal:

```
File Edit Selection View Go Run Terminal Help <- > AI_Internship_Assignment
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\OneDrive\Desktop\AI_ Internship Assignment> python mistral_chatbot.py
>>
llama_model_loader: loaded meta data with 20 key-value pairs and 291 tensors from mistral-7b-instruct-v0.1.Q2_K.ggf (version GGF V2)
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output
llama_model_loader: - kv 0: general.name str
llama_model_loader: - kv 1: general.quantization_version u32
llama_model_loader: - kv 2: llama.context_length u32
llama_model_loader: - kv 3: llama.embedding_length u32
llama_model_loader: - kv 4: llama.block_count u32
llama_model_loader: - kv 5: llama.feed_forward_length u32
llama_model_loader: - kv 6: llama.rope_dimension_count u32
llama_model_loader: - kv 7: llama.attention_head_count u32
llama_model_loader: - kv 8: llama.attention_head_count_kv u32
llama_model_loader: - kv 9: llama.attention_head_count_qk u32
llama_model_loader: - kv 10: llama.rope_freq_base f32
llama_model_loader: - kv 11: general.file_type u32
llama_model_loader: - kv 12: tokenizer_ggml.model.str
llama_model_loader: - kv 13: tokenizer_ggml.tokens arr[str,32000]
llama_model_loader: - kv 14: tokenizer_ggml.scores arr[f32,32000]
llama_model_loader: - kv 15: tokenizer_ggml.token_type arr[i32,32000]
llama_model_loader: - kv 16: tokenizer_ggml.bos_token_id u32
llama_model_loader: - kv 17: tokenizer_ggml.eos_token_id u32
llama_model_loader: - kv 18: tokenizer_ggml.unknown_token_id u32
llama_model_loader: - kv 19: general.quantization_version u32
llama_model_loader: - type f32: 65 tensors
llama_model_loader: - type q2_k: 65 tensors
llama_model_loader: - type qk_k: 160 tensors
llama_model_loader: - type qk_v: 1 tensors
print_info: file format = GGF V2
print_info: file type = Q2_K - Medium
print_info: file size = 2.87 GB (3.41 BW)
init_tokenizer: initializing tokenizer for type 1
load: control token: </s> is not marked as EOS
load: control token: <|> is not marked as EOS
load: special eos_id is not in special_eos_ids - the tokenizer config may be incorrect
load: special tokens cache size = 3
load: token to piece cache size = 0.1637 MB
print_info: arch = llama
print_info: vocab_only = 0
print_info: n_ctx_train = 32768
print_info: n_embd = 4096
In 32, Col 1 | Spaces: 4 | UTF-8 | CR LF | Python | 3.10.11

File Edit Selection View Go Run Terminal Help <- > AI_Internship_Assignment
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python + v x
print_info: n_embd = 4096
print_info: n_layer = 32
print_info: n_head = 32
print_info: n_head_kv = 8
print_info: n_rot = 128
print_info: n_sse = 0
print_info: n_embd_head_k = 128
print_info: n_embd_head_v = 128
print_info: n_gpt = 4
print_info: n_embd_k_gpt = 3024
print_info: n_embd_v_gpt = 3024
print_info: f_norm_eps = 0.0e+00
print_info: f_norm_rms_eps = 1.0e-05
print_info: f_clamp_kqv = 0.0e+00
print_info: f_max_alibi_bias = 0.0e+00
print_info: f_layr_norm_scale = 0.0e+00
print_info: f_attn_norm_scale = 0.0e+00
print_info: n_ff = 14336
print_info: n_expert = 0
print_info: n_expert_used = 0
print_info: causal attn = 1
print_info: pooling type = 0
print_info: rope type = linear
print_info: rope scaling = linear
print_info: rope scale train = 10000.0
print_info: freq scale train = 1
print_info: n_ctx_orig_yarn = 32768
print_info: rope_finetuned = unknown
print_info: ssa_d_conv = 0
print_info: ssa_d_inner = 0
print_info: ssa_d_state = 0
print_info: ssa_dt_rank = 0
print_info: ssa_dt_b_c_rms = 0
print_info: model_type = 78
print_info: model_params = 7.24 B
print_info: general.name = mistralai_mistral-7b-instruct-v0.1
print_info: vocab.type = SPM
print_info: n_vocab = 32000
print_info: n_merges = 0
print_info: BOS token = 1 '<s>'
print_info: EOS token = 2 '</s>'*
In 32, Col 1 | Spaces: 4 | UTF-8 | CR LF | Python | 3.10.11 | x

File Edit Selection View Go Run Terminal Help <- > AI_Internship_Assignment
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
python + v x
print_info: EOS token = 2 '</s>'
print_info: UNK token = 0 'unk'
print_info: LF token = 13 '<0xA>'
print_info: EOS token = 2 '</s>'
print_info: max token length = 48
load_tensors: loading model tensors, this can take a while... (mmap = true)
load_tensors: layer 0 assigned to device CPU
load_tensors: layer 1 assigned to device CPU
load_tensors: layer 2 assigned to device CPU
load_tensors: layer 3 assigned to device CPU
load_tensors: layer 4 assigned to device CPU
load_tensors: layer 5 assigned to device CPU
load_tensors: layer 6 assigned to device CPU
load_tensors: layer 7 assigned to device CPU
load_tensors: layer 8 assigned to device CPU
load_tensors: layer 9 assigned to device CPU
load_tensors: layer 10 assigned to device CPU
load_tensors: layer 11 assigned to device CPU
load_tensors: layer 12 assigned to device CPU
load_tensors: layer 13 assigned to device CPU
load_tensors: layer 14 assigned to device CPU
load_tensors: layer 15 assigned to device CPU
load_tensors: layer 16 assigned to device CPU
load_tensors: layer 17 assigned to device CPU
load_tensors: layer 18 assigned to device CPU
load_tensors: layer 19 assigned to device CPU
load_tensors: layer 20 assigned to device CPU
load_tensors: layer 21 assigned to device CPU
load_tensors: layer 22 assigned to device CPU
load_tensors: layer 23 assigned to device CPU
load_tensors: layer 24 assigned to device CPU
load_tensors: layer 25 assigned to device CPU
load_tensors: layer 26 assigned to device CPU
load_tensors: layer 27 assigned to device CPU
load_tensors: layer 28 assigned to device CPU
load_tensors: layer 29 assigned to device CPU
load_tensors: layer 30 assigned to device CPU
load_tensors: layer 31 assigned to device CPU
load_tensors: layer 32 assigned to device CPU
load_tensors: tensor "token_embd.weight" (Q2_K) (and 299 others) cannot be used with preferred buffer type CPU_AARCH64, using CPU instead
load_tensors: CPU Mapped model buffer size = 2939.57 MiB
In 32, Col 1 | Spaces: 4 | UTF-8 | CR LF | Python | 3.10.11 | x
```

- **Task 1 Prompt and Output:**

💡 You: Suggest 3 creative and unique prompt ideas for a chatbot that helps startup founders with fundraising advice and strategies.

Generating response...

llama.generate: 4 prefix+match hit, remaining 29 prompt tokens to eval
llama.perf_context_print: load time = 11084.07 ms
llama.perf_context_print: prompt eval time = 12645.66 ms / 29 tokens (436.06 ms per token, 2.29 tokens per second)
llama.perf_context_print: eval time = 70568.65 ms / 251 runs (281.15 ms per token, 3.56 tokens per second)
llama.perf_context_print: total time = 83539.13 ms / 280 tokens

⌚ Mistral: 1. "Chat with a fundraising expert": Create a chatbot that can simulate a conversation with an experienced fundraising expert who can provide advice and strategies based on the user's specific needs and goals. The chatbot could also provide personalized recommendations based on the user's industry, target market, and funding stage.

2. "Fundraising trivia game": Create a chatbot that asks users a series of fundraising trivia questions. The chatbot could provide hints and tips on how to answer each question correctly, and at the end of the game, provide a summary of the user's knowledge and areas for improvement. The chatbot could also offer additional resources and articles to help users learn more about fundraising.

3. "Fundraising challenge": create a chatbot that presents users with a fundraising challenge, such as raising a certain amount of money in a certain time frame. The chatbot could provide guidance on how to create a successful fundraising campaign, including how to identify potential donors, craft a compelling pitch, and promote the campaign. The chatbot could also provide feedback on the user's progress and offer additional resources if the user needs more support.

• **Task 2 Improve a Prompt:**

You: Improve the prompt: "Tell me about digital marketing." Make it more specific, result-oriented, and useful for AI.

Generating response...

```
llama.perf_context_print:      load time = 11316.41 ms
llama.perf_context_print: prompt eval time = 11316.42 ms /  34 tokens ( 332.78 ms per token,   3.00 tokens per second)
llama.perf_context_print: eval time = 31504.72 ms /  52 runs ( 605.86 ms per token,   1.65 tokens per second)
llama.perf_context_print:      total time = 42890.74 ms /  86 tokens

⌚ Mistral: Could you provide a detailed and specific explanation of digital marketing strategies that have been proven to increase website traffic and conversion rates? Additionally, please include examples of successful digital marketing campaigns and provide insights into how AI can be used to improve and optimize digital marketing efforts.

💡 You:
● Generating response...

llama.generate: 4 prefix-match hit, remaining 5 prompt tokens to eval
llama.perf_context_print:      load time = 11316.41 ms
llama.perf_context_print: prompt eval time = 2469.79 ms /  5 tokens ( 493.96 ms per token,   2.02 tokens per second)
llama.perf_context_print: eval time = 19174.37 ms /  35 runs ( 547.84 ms per token,   1.83 tokens per second)
llama.perf_context_print:      total time = 21692.49 ms /  40 tokens

⌚ Mistral: It appears that you have entered a title for this section. However, there is no text associated with the title. Please add the text you want to appear in the section.
```

- **Task 3 Welcome Message:**

You: Write a short, friendly, and professional welcome message for a chatbot called 'Startup GPT' that helps startup founders with fundraising strategies. The message should encourage users to ask questions.
Generating response ...

Generating response...

```
llama.generate: 4 prefix-match hit, remaining 44 prompt tokens to eval
llama.perf.context.print:    load time = 11316.41 ms
llama.perf.context.print: prompt eval time = 12051.70 ms /   44 tokens ( 273.90 ms per token,      3.65 tokens per second)
llama.perf.context.print: eval time = 26930.41 ms /   74 runs ( 363.92 ms per token,      2.75 tokens per second)
llama.perf.context.print: total time = 39069.12 ms / 118 tokens
Mistral: Hello and welcome to Startup GPT, your friendly AI fundraising companion! I'm here to help you navigate the exciting world of startup fundraising strategies. Feel free to ask any questions you may have about securing funding for your startup. Remember, every startup journey is unique, so let's tailor your approach together! Let's get started!
```

• **Appendix: Source Code (Local GPT Chatbot):**

```
from llama_cpp import Llama
```

```
llm = Llama(
    model_path="mistral-7b-instruct-v0.1.Q2_K.gguf",
    n_ctx=2048,
    n_threads=4
)
print("🤖 Mistral GPT Chatbot Ready!")
print("Type 'exit' to quit.\n")
```

while True:

```
prompt = input(" 🧑 You: ")
```

```
if prompt.lower() == "exit":  
    break
```

```
print("Generating response...")
```

```
print("Generating response...\n")  
full_prompt = f"[INST]{prompt.strip()}[/INST]"
```

output = lm(full_prompt, max_tokens)

```
reply = output['choices'][0]['text']
```