First, let’s define what Prompts are.

Prompts are guiding cues or instructions used to elicit specific responses from language models. They serve as a bridge in our communication with language models, helping us direct the model to generate responses that align with our specific needs.

**1. Enhance Response Accuracy.**

Prompts guide models to generate more accurate and relevant answers. With Prompts, the model’s responses are more targeted.

**2. Improve Context Understanding.**

Prompts help in directing the model to perform specific tasks like summarisation or translation by improving its understanding of the context

**3. Boost Creativity and Exploration.**

Prompts can stimulate creative responses and explore various hypothetical scenarios. By giving targeted prompts, the model can produce imaginative and innovative text, helping us explore new ideas and concepts.

I’ll introduce several prompts types we used in our experiment.

The first one is zero-shot prompting, when using zero-shot ,we directly throw task to the model without any example and context.

For few-shot prompting, we’ll give some examples for model to refer to help them understand.

As for direct prompting, we are asking model a question or giving a clear command to the model. The difference between direct prompting and zero-shot prompting we do not give specific subject to zero-shot prompting.

Then contextual prompting, we added some context or background information to guide model.

Persona prompting, we always set a specific role to model to help it adopt and understand the scenario.

Completion Prompting, give beginning text and let model to complete it.

RAG provides more context to the prompt, by drawing reference to the past data collected and stored in the vector store. The references drawn are specific to the user’s query.

In this example, beside the pre-defined zero-shot prompting, we also get more context from RAG.

We retrieved some relating templates from AlloyDB vector store based on the text embedding of the query and the database.

Here are two results generated by our models.

The left one is generated without RAG and the right one is with RAG.

We can see clearly that without RAG, the result exist a lot problems, like Hallucination , the model will add some info which does not appear in the context. And it does not mention the company info. And in the end, it added redundant info. If we want put this message into use, we must do a lot of revise and tailor.

Let’s look at the result of RAG, no hallucination since we provided plenty context for model to refer. And it constantly quoting company’s name within the response, adding more personalized touches to the content​.

So we see the benefits of RAG:​

It can reduce the risks of hallucination.

Tone used in the message is more customized and specific to company info/tone.