

## Summary on Day 22:

I have completed my project where jayapriya assigned to me.so today my mentor told me to read the prompt engineering documents.

When we gave the prompt in the AI tool sometimes it does not give the exact output that what we needed,so we need prompt engineering. This documents helps us to understand what are all the challenges we will face while crafting prompts.

### Work of LLM:

It is a prediction engine. If we gave the text as input then it predicts what the following token should be, based on the data it was trained.

Before doing prompt engineering, we need to **choose the module** like chatgpt,Gemini,claude etc.

### LLM output configuration:

After choosing the models we will need to give model configuration. LLM contains various configuration.

First configuration is the **output length**.it generates the number of tokens when we gave the input. When the output length is higher then the LLM stops predicting the tokens when it reaches the limit.

Second configuration is **sampling controls**. LLM do not predict a single token.it predicts the probability of what the next token would be using the sampling controls. They are

- **Temperature**
- **Top k**
- **Top p**

**Temperature:** if we gave a single word in LLM, it predicts the probability of what the next token. So **lower** the temperature in prompts that expects more deterministic response. **higher** the temperature leads to unexpected results.

**Top k and top p:** it **restricts** the predicted probability of next token. Similar to temperature it also have some controls.

### Setting values to temperature, top k, top p:

- If temperature is 0,top k and top p becomes irrelevant.

- If temperature is extremely high, then temperature becomes irrelevant.
- If top-k is 1(higher), then temperature and top p becomes irrelevant.
- If top-p is 0, then temperature and top k becomes irrelevant.

### **Prompting techniques:**

**Zero shot:** it is the simplest prompt.it is like general prompting.

**One shot:** in one shot the prompt which provides a single example which helps the models to understand what you are asking for.

**Few shot:** it is similar to one shot with many examples.

### **Types of prompt:**

1. **System prompt**
2. **Role prompt**
3. **Contextual prompt**

**System prompt:** that generates the output when it meets the specific requirements.

**Role prompt:** once the role is assigned to the module, you can then give prompt to that role.

**Contextual prompt:** by providing contextual prompt, AI interactions are seamless and efficient.

**Step-back prompt:** is a technique for improving the performance by prompting. In general we first give the prompt and after it displays the answer then we again give prompt relevant to the specific task.

**Chain of thought (cot):**is a technique for improving reasoning. In general we give prompt and to improve reasoning we add the prompts as step-by-step.

**Tree of thought:** it is similar to chain of thought but with multiple reasoning paths simultaneously. It is used in complex problems.

**reAct (reason and act):** used to solve complex problems combined with external tool such as interacting with external API's to retrieve information.

**Automatic prompt engineering (APE):** writing prompts can be complex.

Write prompt to write prompts. This can be achieved using APE.

Today I have read these concepts and implemented some of the concepts in google AI studio and cleared some of my doubts with Richy.

Afternoon we had a session with Koushik sir. There he discussed about the previous problems which he given us to solve.

**Scope of variable:** it is the code declared inside the function cannot be accessed outside the function.

**Execution stack:** it is nothing but placing one over another.

For example: `int a=10;`

`Intb=20;`

`Add()`

`Print(a+b);`

First it stores the declared variables in the stack. then it stores the add function. Then `print(a+b)` is stored.

Then he given some examples to solve. Each one came with different outputs. Then he explained the problem detaily.

In this class I understood the concept of execution stack like how it stores the inputs which we given in the program step by step.