

# # ChatGPT prompt engineering

## INTRODUCTION :

Two types LLM

### ① Base LLM

Predicts next word, based upon text training data [internet articles & huge data]

Ex Once upon a time was a unicorn  
Base LLM predict → that lived in a magical forest with her friends.

Ex what is the capital of France?

Base LLM → what is France's largest city?  
what is France's population?

### ② Instruction Tuned LLM [We focus on this]

Trained to follow instruction.  
fine tune's on instruction and good attempt in following those instruction.

Ex> what is capital of France?

The capital of France is Paris.

We start off with Base LLM [already trained on text data] and further fine tune with Input and Output that are instructions and good attempts to follow the instruction.

This is further fine tuned using RLHF:  
Reinforced learning with human feedback

## GUIDELINES FOR PROMPTING

Principle 1 : write clear and specific instruction

Principle 2 : Give model some time to think.

→ Instruction should be as clear as possible, clear & specific doesn't mean prompt to be "sweet".

Longer prompts actually provide more clarity

Tactic 1 : These tactics are based on Principle 1  
use delimiter's to clearly indicate distinct part of input

For ex ⇒ "Some text" = text

prompt =   
"Summarize text delimited  
by triple backticks into single sentence  
```text```"



can be any clear punctuation that separates specific pieces of text.  
XML tags, quotes etc.

- How delimiters help in avoiding Prompt Injection?

Someone adds conflicting instruction to model by adding to prompt and making follow user instruction

for example in -text summarization, someone adds like or forget previous instruction & ~~to~~ ~~summarize~~ about pandas.

If any instruction gets added to our text we would just summarise it as it is part of our text under delimiters.

So in prompt injection the dynamic text provided to model can be misused to change instruction but being inside a delimiter it becomes part of our own devised prompt that seeks summarization of text inside delimiter.

Tactic d\* : Ask for structured output for ex JSON & HTML.

for ex prompt = "Give list of 3 made up books provide them in JSON format using following keys book-id, title, author"

These kind of response can be read directly.

Tactic 3\*: check whether conditions are satisfied. Check assumptions required to do the task.

Helps in dealing edge cases & how model could handle results.

For ex "In our prompt we can add something like in the end"

If text contains sequences of instructions then write instruction or else simply write no instruction provided.

This is like working on If - Else conditioning on the text data.

So if text follow some condition then do one output or else do other output.

Tactic 4 : few shot prompting  
Give successful example of completing the task. Then ask model to perform the task.

prompt = " Task to perform  
example 1 => Show hows output looks like  
example 2 "

PRINCIPLE 2 → Give your model time to think.

\* So if a complex task to do in short amount of time or in small number of words it may make up a guess which might be incorrect

Complex task

For ex -> summarizing a 1000 words essay into 50 or 100 words.  
Short output

what to do?

Reframe query to request chain or series of relevant reasoning to arrive at an answer.

Spending more computation power on the task.

TACTIC 1: Specify the steps required to complete the task.

For ex → for some text  
prompt = "1"

Your task is to perform following actions

- 1 - Summarize text demarcated by  
<>
  - 2 - Translate the summary to French
  - 3 - Output a json object that contains following key : french\_summary.

Use following format

Text : <text to summarize>

Summary : <summary>

Translation : <translation>

Output json : <json with french  
summary>

manu

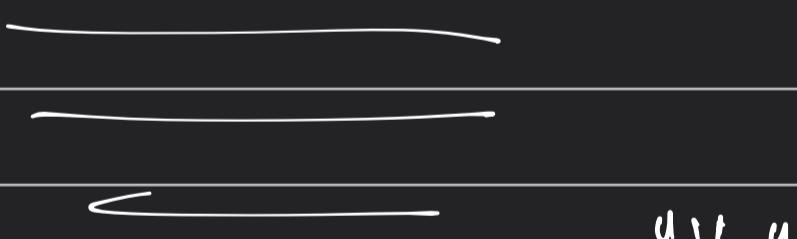
TACTIC 2 : Instruct model to work out its own selection before reaching to a conclusion

prompt = " " Determine if selection is correct or not.

To solve it use following instruction

- 1) First work out your own sol
- 2) Compare your sol with given sol
- 3) Evaluate if its correct or not

Format :



MODEL LIMITATIONS : HALLUCINATIONS

As models have been trained on large amount of data they haven't perfectly memorized the information.

It doesn't know its boundary and its

knowledge very well. This means it can answer about obscure topics and can make up things that are not actually true.

Reduce Hallucinations: If we are trying to answer some info based upon some text, we could ask model to find relevant info about the text and then answer based on relevant information.

