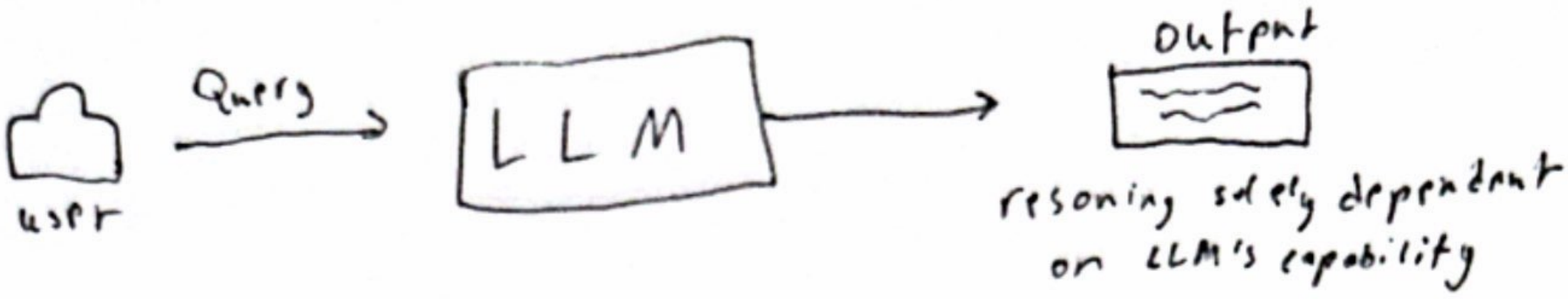
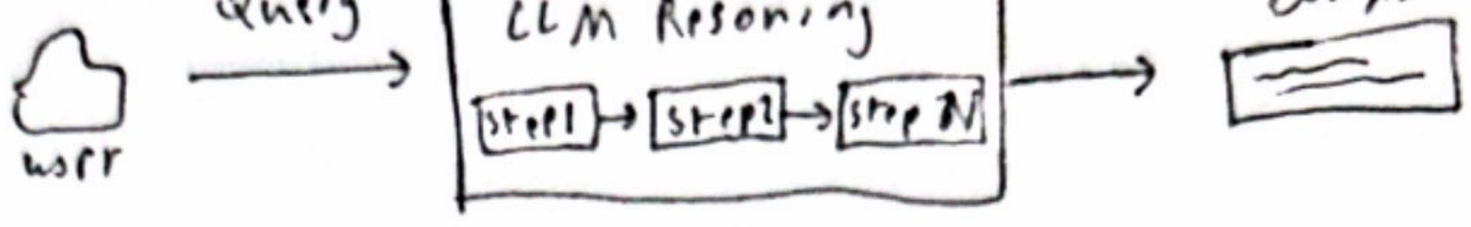
More Prompt Engineering Approaches (simple, CoT, ARQ)

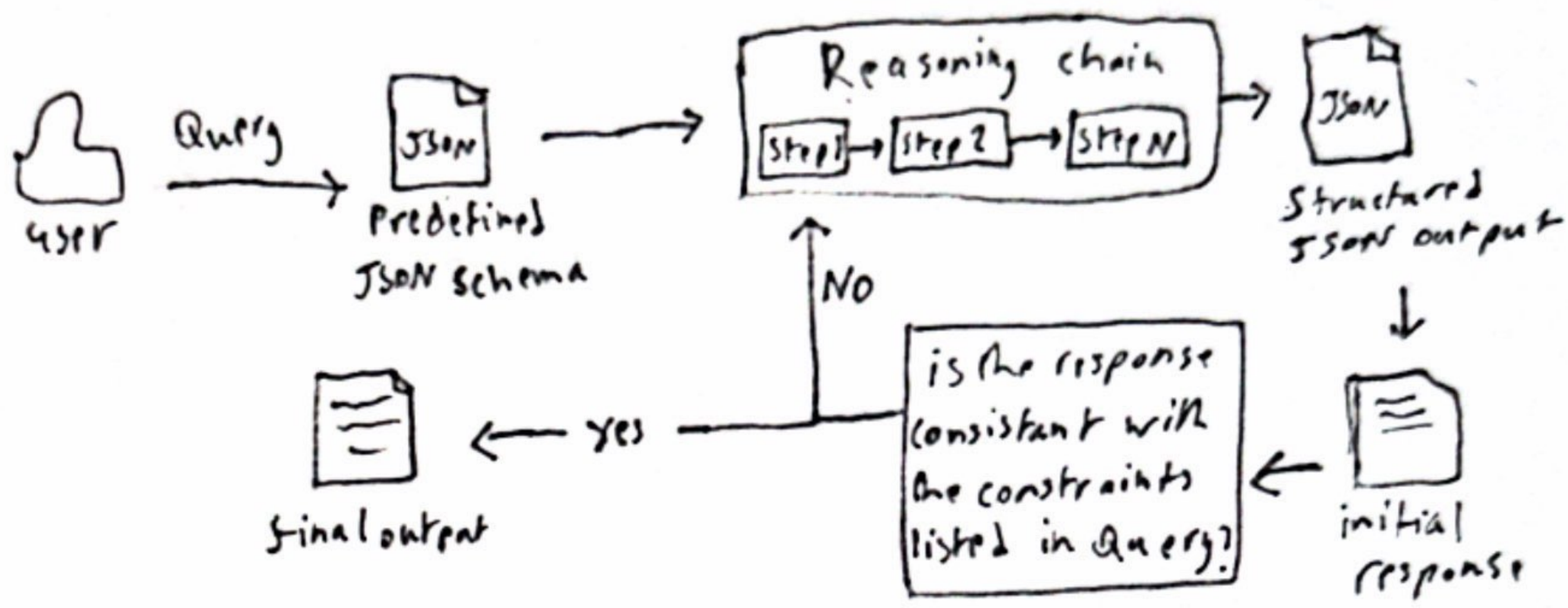
Direct prompting :



Chain of thought (CoT) :



Attentive Reasoning Queries (ARQ) :



Prevents Hallucinations more

Positional Embeddings

- in short: a positional embedding vector is a vector added to token embeddings to encode the position of each token in a sequence, so the model knows the order of the words

- Problem: Transformers Like GPT process all tokens in parallel (unlike RNN) they dont have a idea of sequence order. so feeding it a sentence Like: "the cat sat on the mat", without any extra info, they only see a bag of embeddings not which word came first.

- solution: Positional embedding.. a positional embedding is a vector that encodes the position of a token only (where in the sequence it is) position 0, 1, 2 etc, these are added to the token embedding (also vector) before feeding them into the transformer

So Ex: input: ["the", "cat," "sat", ...] $\xrightarrow[\text{learned from vocabulary}]{\text{Token embeddings}}$ E("the"), E("cat"), E("sat")...

$\xrightarrow[\text{Fixed or learned vectors}]{\text{positional embeddings}}$ P(0), P(1), P(2)... $\xrightarrow{\text{Model uses}}$ E("the")+P(0), E("cat")+P(1) ....

→ this way model knows which token came where

Two Main Types :
- Fixed (sinusodial): based on sin/cos functions of position (as in the Attention is all you need paper)
- Learned: Trainable vectors 112—trained the same way as word (token) embeddings during training (used in GPT)