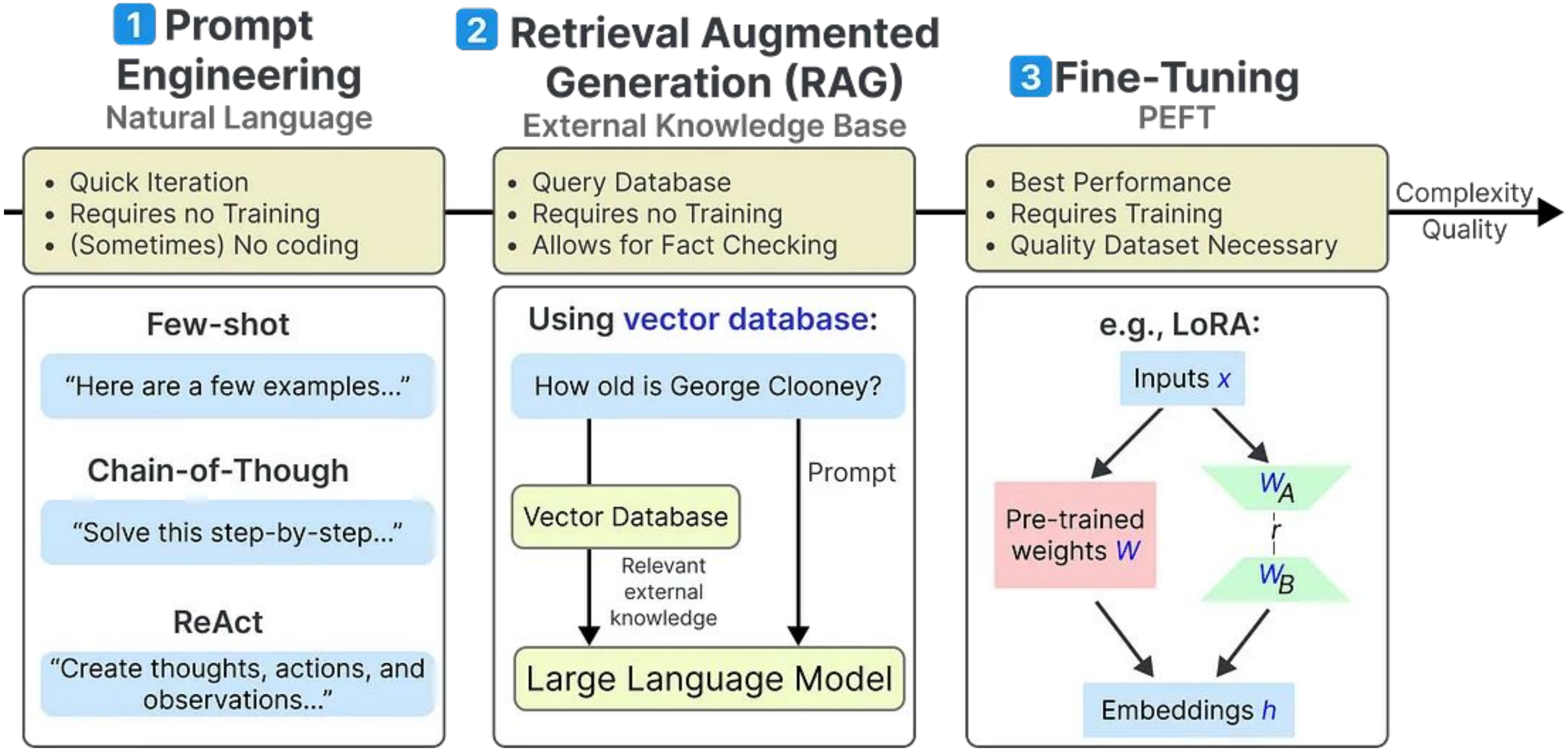
Improving LLM Performance

@jai-llm
3-Oct-2023

Prompting, RAG, Fine Tuning Improve LLM Performance



1 Prompt Engineering

Natural Language

- Quick Iteration
- Requires no Training
- (Sometimes) No coding

2 Retrieval Augmented Generation (RAG)

External Knowledge Base

- Query Database
- Requires no Training
- Allows for Fact Checking

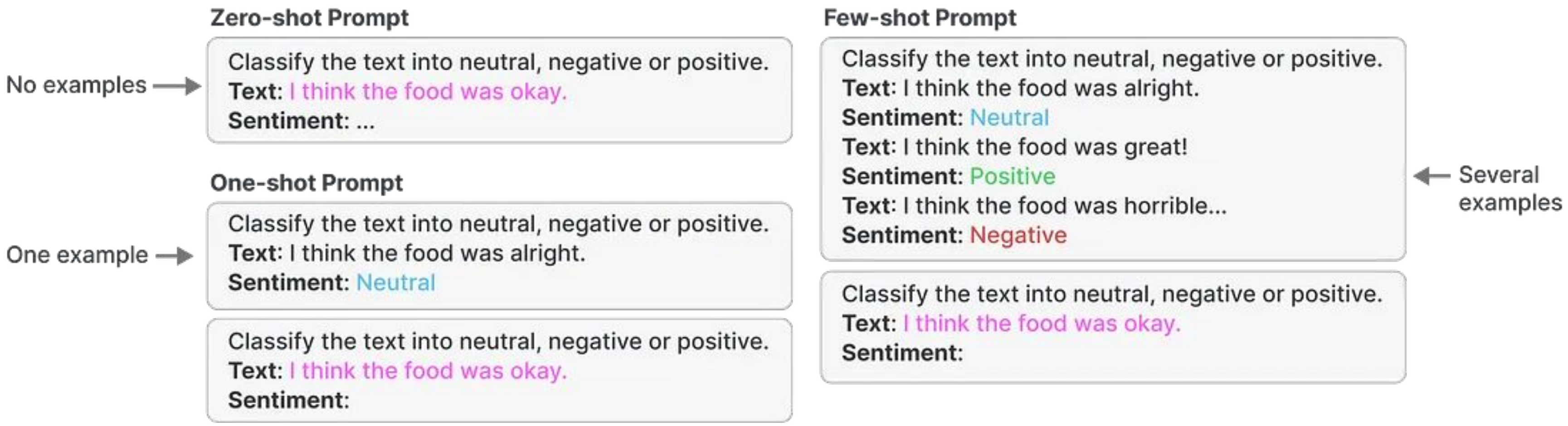
3 Fine-Tuning

PEFT

- Best Performance
- Requires Training
- Quality Dataset Necessary

Source: Medium

1. Prompt Engineering Flavors: Zero-Shot vs Few-Shot



Zero-shot Prompt

Classify the text into neutral, negative or positive.
Text: I think the food was okay.
Sentiment: ...

One-shot Prompt

Classify the text into neutral, negative or positive.
Text: I think the food was alright.
Sentiment: Neutral

Classify the text into neutral, negative or positive.
Text: I think the food was okay.
Sentiment:

Few-shot Prompt

Classify the text into neutral, negative or positive.
Text: I think the food was alright.
Sentiment: Neutral
Text: I think the food was great!
Sentiment: Positive
Text: I think the food was horrible...
Sentiment: Negative

Classify the text into neutral, negative or positive.
Text: I think the food was okay.
Sentiment:

Source: Medium

1. Zero Shot Prompting Example

```
prompt = """
<s>[INST] <<SYS>>

You are a helpful assistant.

<</SYS>>

Classify the text into neutral,
negative or positive.
Text: I think the food was okay.
[/INST]
"""
print(generator(prompt)[0]["generated_text"])
```

```
"""
Positive. The word "okay" is a
mildly positive word,
indicating that the food was
satisfactory or acceptable.
"""
```

Source: Medium

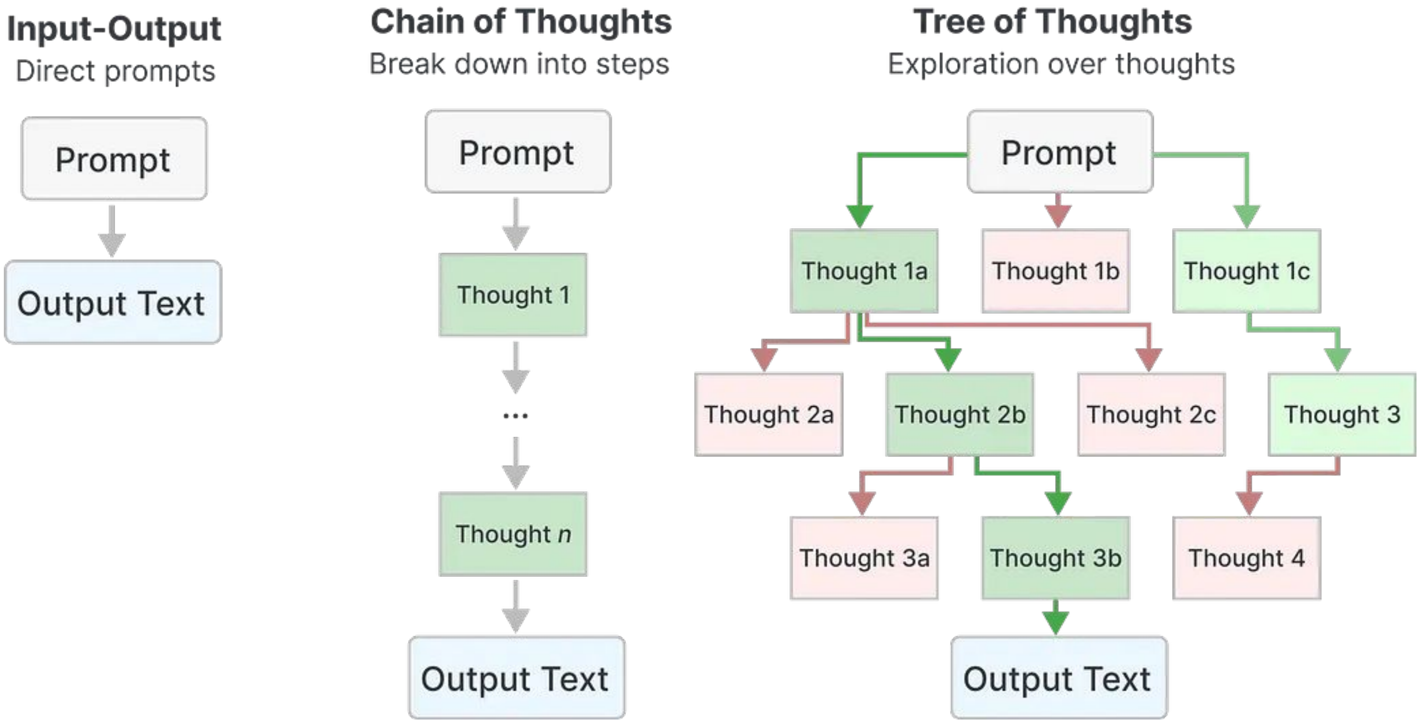
1. Few Shot Prompting Example

```
prompt = """
<s>[INST] <<SYS>>

You are a helpful assistant.

<</SYS>>

Classify the text into neutral, negative or
positive.
Text: I think the food was alright.
Sentiment:
[/INST]

Neutral</s><s>

[INST]
Classify the text into neutral, negative or
positive.
Text: I think the food was okay.
Sentiment:
[/INST]
"""
print(generator(prompt)[0]["generated_text"])
```

```
"""
Neutral
"""
```

Source: Medium

1. Prompt Engineering Flavors: Thought Based



Source: Medium

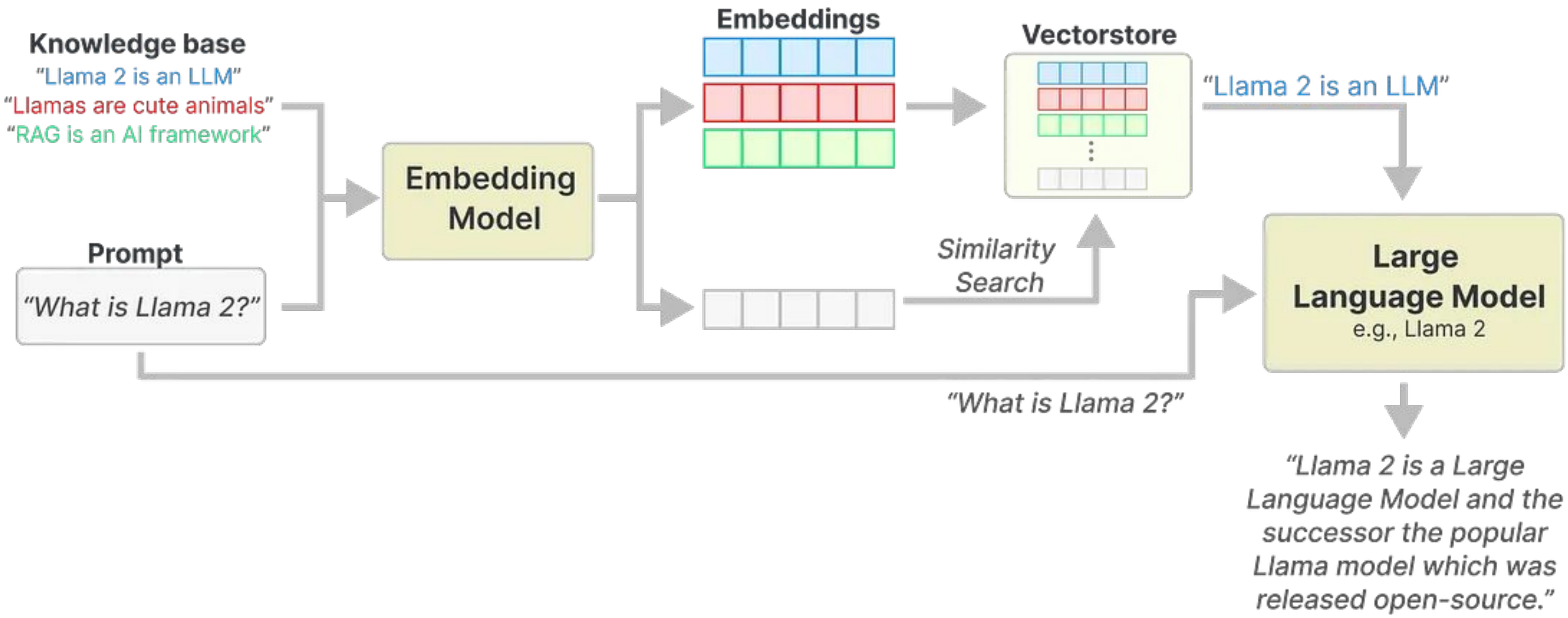
1. Thought Based Prompting Example

```
prompt = """
<s>[INST] <<SYS>>

You are a helpful assistant.

<</SYS>>

Do the odd numbers in this group
add up to an even number? 3, 5, 15,
32.
Solve by breaking the problem into
steps.
Identify the odd numbers, add them,
and indicate whether the result is
odd or even.[/INST]
"""
print(generator(prompt)[0]["generat
ed_text"])
```

```
"""
Hello! I'd be happy to help you with that.
Let's break down the problem into steps:

Step 1: Identify the odd numbers in the
group.

The odd numbers in the group are:

3, 5, 15

Step 2: Add the odd numbers.

3 + 5 + 15 = 23

Step 3: Is the result odd or even?

23 is an odd number.

Therefore, the sum of the odd numbers in the
group is an odd number.
"""
```

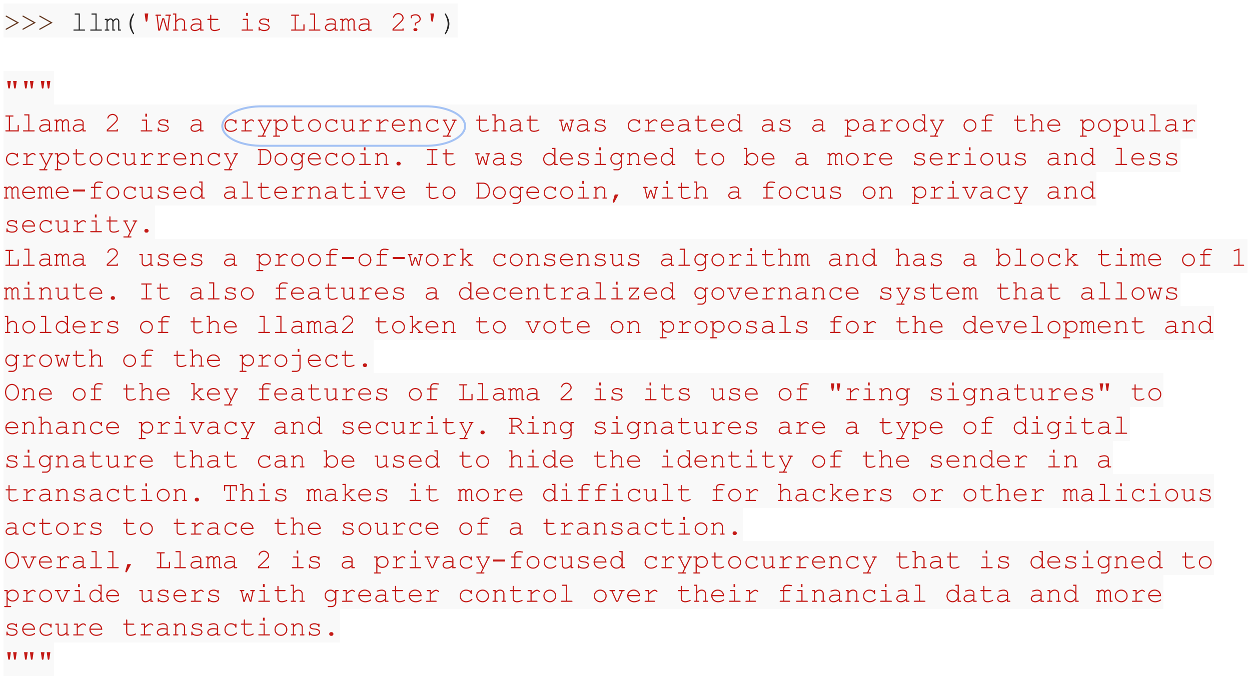
Source: Medium

2. RAG Improves Response using Context from Vector DB



Source: Medium

2. Base LLM Does Not Know Llama2 is an LLM



```
>>> llm('What is Llama 2?')

"""
Llama 2 is a cryptocurrency that was created as a parody of the popular
cryptocurrency Dogecoin. It was designed to be a more serious and less
meme-focused alternative to Dogecoin, with a focus on privacy and
security.
Llama 2 uses a proof-of-work consensus algorithm and has a block time of 1
minute. It also features a decentralized governance system that allows
holders of the llama2 token to vote on proposals for the development and
growth of the project.
One of the key features of Llama 2 is its use of "ring signatures" to
enhance privacy and security. Ring signatures are a type of digital
signature that can be used to hide the identity of the sender in a
transaction. This makes it more difficult for hackers or other malicious
actors to trace the source of a transaction.
Overall, Llama 2 is a privacy-focused cryptocurrency that is designed to
provide users with greater control over their financial data and more
secure transactions.
"""
```

Source: Medium

2. LLM + RAG Gives Better Response

```
knowledge base = [
"On July 18, 2023, in partnership
with Microsoft, Meta announced
LLaMA-2, the next generation of
LLaMA." ,
"Llama 2, a collection of
pretrained and fine-tuned large
language models (LLMs) ",
"The fine-tuned LLMs, called Llama
2-Chat, are optimized for dialogue
use cases.",
"Meta trained and released LLaMA-2
in three model sizes: 7, 13, and 70
billion parameters.",
"The model architecture remains
largely unchanged from that of
LLaMA-1 models, but 40% more data
was used to train the foundational
models.",
"The accompanying preprint also
mentions a model with 34B
parameters that might be released
in the future upon satisfying
safety targets."
]
```

```
>>> rag('What is Llama 2?')

"""
Llama 2 is a collection of
pretrained and fine-tuned large
language models
(LLMs) announced by Meta in
partnership with Microsoft on July
18, 2023.
"""
```

Source: Medium

2. Key Benefits of RAG are Trust and Safety

LLM Problems

1. **Static -** LLMs know what is in their training data so events happening post-training are not known to the LLM.
2. **Lack Domain Knowledge** - Trained to perform generalized tasks and does not know the specific context of your problem.
3. **Black Boxes -** It is not easy to know the sources the LLM considered when it arrived at its response. This makes it hard to trust LLMs.
4. **Training is Costly** - Building LLM and fine-tuning them requires specialized skills and is costly in terms of compute time.

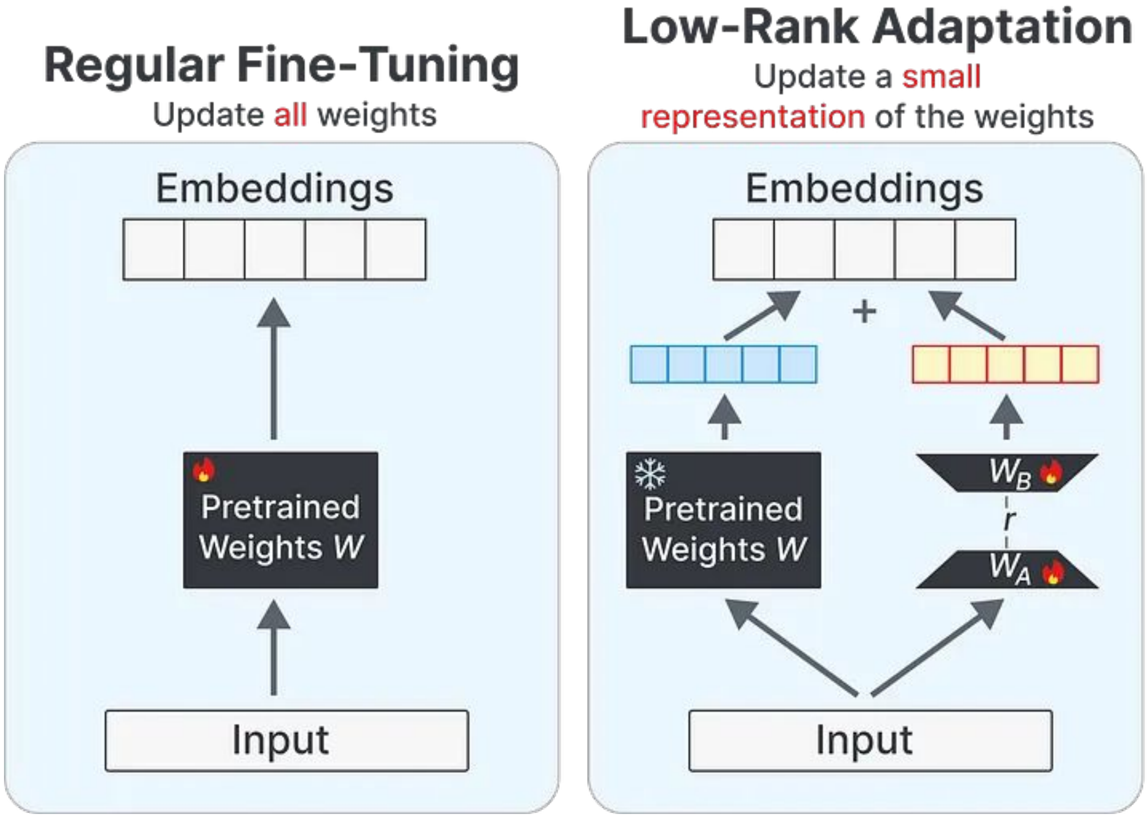
RAG Benefits

1. **Safety** - Reducing "hallucinations" by providing better context.

2. **Trust** - Increasing trust by providing the response sources (White-boxing LLMs).

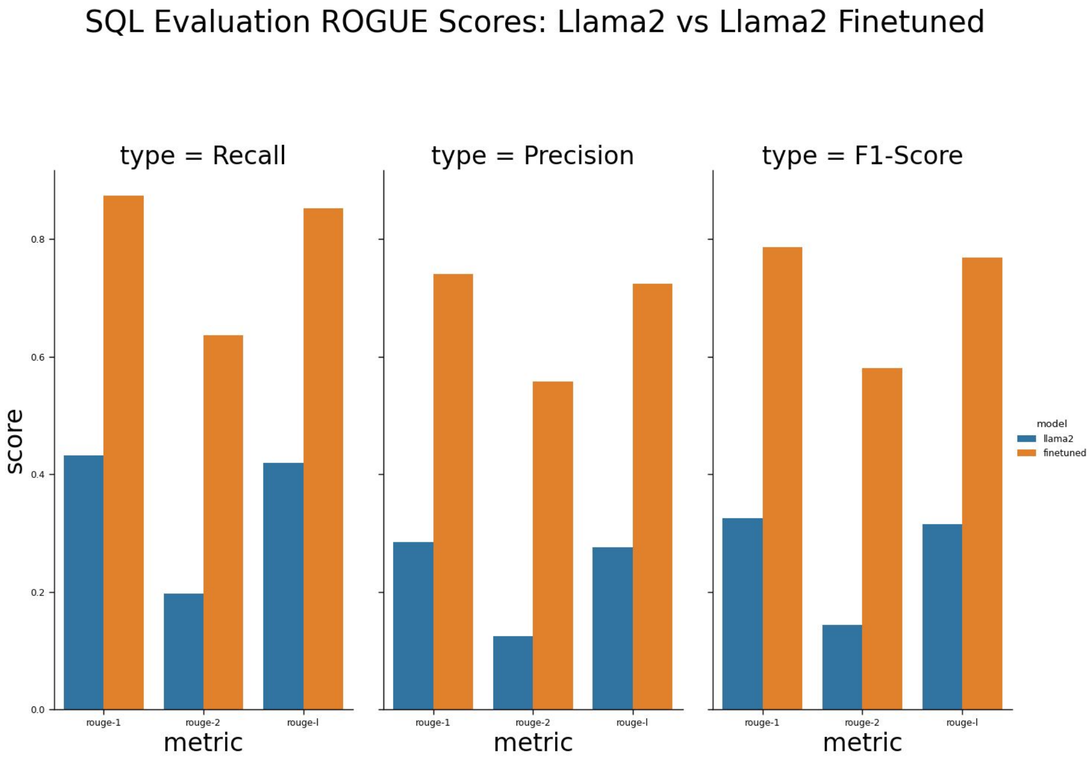
Source: Jai Github

3. PEFT+LoRA Finetune Save Time and Resources



Source: Medium

3. PEFT+LoRA: 2-3x Better ROGUE Score on Text2SQL



Source: [Jai Github]()

Takeaways

- LLM performance tuning involves prompting, RAG, and Finetuning.
- Prompt Engineering has 3 flavors:
  - **Zero Shot**.
  - **Few Shot**.
  - **Thought Based**: Chain of Thoughts and Tree of Thoughts.
- RAG can improve safety and trust with LLM but needs a larger context window to be effective.
- Finetuning is most effective but requires expertise and is resource intensive.
- Our RAG and Fine-tuning findings:
  - **Llama2 RAG** - Improves responses by providing context. Ex: Trees vs Decision Trees.
  - **Llama2 Finetuning** - 2-3x improvement in ROGUE scores on Text2SQL tasks.