Understanding Transformer Model Terminology and Prompt Engineering

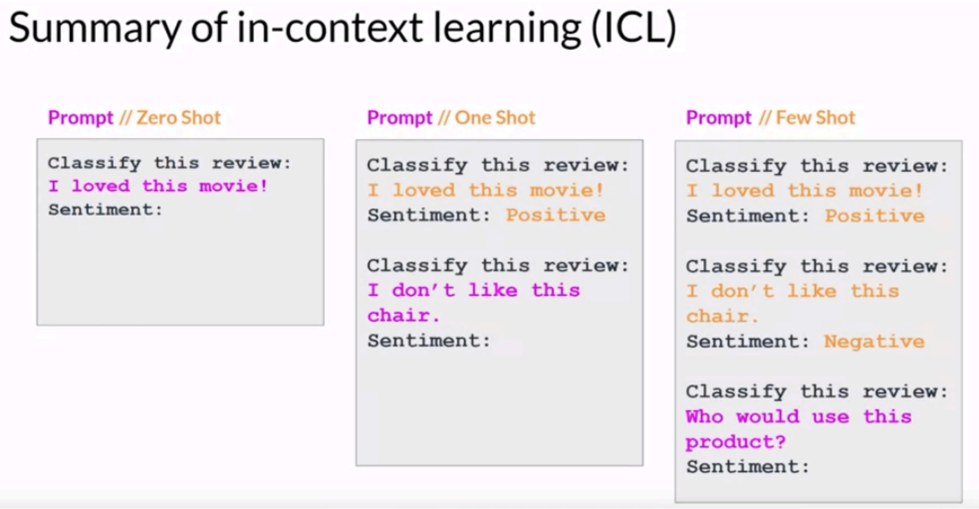
- Terminology Reminder:
    - Prompt:Text fed into the model for generation.
    - Inference: Process of generating text.
    - Completion: Output text generated by the model.
    - Context Window: Total available text for the prompt.

- Prompt Engineering and In-Context Learning:
    - Models may not always produce desired outcomes on the first try.
    - Prompt engineering involves refining prompt language to achieve desired results.
    - Including examples within the prompt can improve model performance.
    - In-Context Learning:-Incorporating examples or additional data in the prompt.

- Zero-Shot Inference:
    - Example: Classifying sentiment of a review without providing examples.
    - Larger models proficient in zero-shot inference.
    - Model grasps tasks and generates accurate responses without examples.

- One-Shot Inference:
    - Example: Providing a single example within the prompt.
    - Helps smaller models understand task and response format better.
    - Improves performance compared to zero-shot inference.

- Few-Shot Inference:
    - Example: Including multiple examples within the prompt.
    - Helps models learn from multiple instances of desired behaviour.
    - Improves performance further for smaller models.



Summary of in-context learning (ICL)

Prompt // Zero Shot

```
Classify this review:
I loved this movie!
Sentiment:
```

Prompt // One Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment:
```

Prompt // Few Shot

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this
chair.
Sentiment: Negative

Classify this review:
Who would use this
product?
Sentiment:
```

- Context Window Limitation:
  - There's a limit on the amount of in-context learning that can be passed into the model.
  - If the model struggles with multiple examples, consider fine-tuning instead.

- Model Scale and Task Performance:
  - Larger models perform better across multiple tasks due to increased parameters.
  - Smaller models proficient in tasks similar to the ones they were trained on.
  - Choose a model based on specific task requirements.

- Experimenting with Model Settings:
  - Once a suitable model is found, experiment with configuration settings.
  - Settings influence structure and style of completions generated by the model.