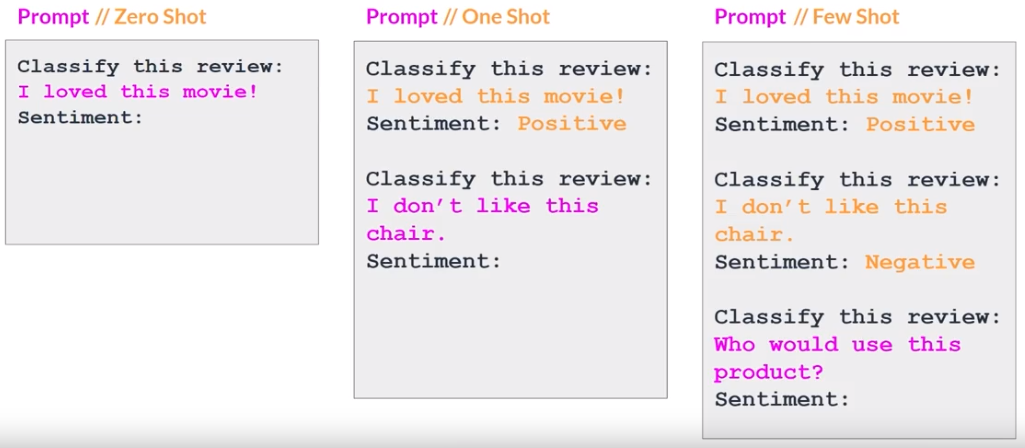
Về Attention:

* Trước khi đưa vào mô hình, cần tokenizer mỗi word thành 1 số rồi thành 1 vector.
* Đưa vào encode, thì sẽ có multi-head attention, có thể hiểu mỗi head sẽ focus vào 1 thứ để define cái input và xác định output dễ hơn. Ví dụ head 1 focus vào thời gian, head 2 focus ngữ cảnh, head 3 focus tên riêng, . . .
* Có những mô hình only encode, hoặc only decode, hoặc đầy đủ encode - decode.

Về LLMs:

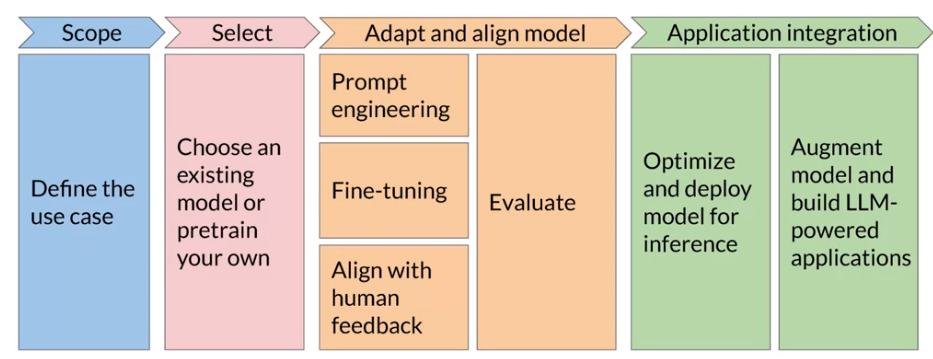
* Mỗi 1 input text là prompt.
* Số lượng memory cần để lưu prompt đó là context windows.
* Mỗi 1 output text là completion.
* Tùy vào việc model có lớn hay không mà cần để ý tới prompt



Về Generative Configuration:

* Max new tokens: Giới hạn số lượng token của output.
* Sample Top K: Lựa ra top K token có probability cao nhất.
* Sample Top P: Lựa ra top những token có total probability của những token đó bằng P.
* Temperature: Điều khiển sự randomness của model, temperature càng lớn thì output càng random và creative.

Về AI Project Lifecycle:

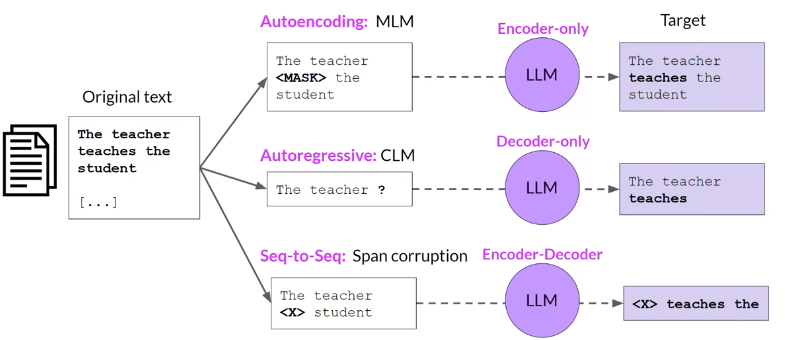


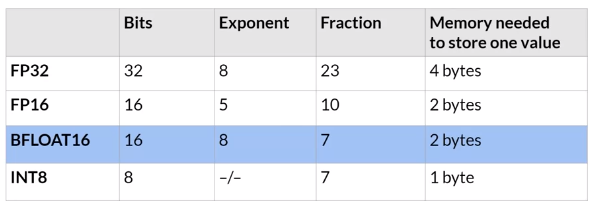
Về bài Labs:

* Sử dụng Flat - 5, Torch, Transformers.
* Test về Prompting để tìm ra cấu trúc prompt phù hợp nhất với các trường hợp zero shot, one shot, few shot.
* Speaker khuyên không nên có quá nhiều shot vì từ sau shot thứ 5 và 6 thì thường không cải thiện hơn là bao.

Về Pre-train LLMs:

* Only encoder model = Autoencoding model: Train by using Masked Language Modeling (MLM). Autoencoding model good for: Sentiment Analysis; Named Entity Recognition; Word Classification. Example of Autoencoding models: BERT, ROBERTA.
* Only decoder model = Autoregressive model: Train by using Causal Language Modeling (CLM). Autoregressive model good for: Text Generation; Other emergent behavior (depends on model size). Example of Autoregressive models: GPT, BLOOM.
* Encoder-Decoder model = Sequence-to-sequence model: ex: T5, train by using span corruption. Sequence-to-sequence models good for: Translation; Text Summarization; Question-Answering. Example of Sequence-to-sequence models: T5, BART.
* Understand about Chinchilla Scaling Law and BloombergGPT’s struggles.





Về Catastrophic Forgetting: Một LLM, sau khi fine tune để nó chỉ làm 1 task, thì nó sẽ hoạt động kém đi ở các task khác. Ví dụ, train LLMs để làm sentiment, thì nó sẽ recognize name entity kém đi.

-> To avoid this: At 1st, define whether this phenomenon can affect the model or not. If yes, fine tune multitask at the same time (require 50 - 100K data examples across many tasks, need more resources to compute). Another way to handle this without fine tuning multitask is using Parameter Efficiency Fine-Tuning (PEFT)

To Fine Tune a specific LLM: Check the used data in this LLM to see the pros and cons in order to have an ability to analyze further.

About the Benchmark text: The passage discusses the limitations of simple evaluation metrics like ROUGE and BLEU for assessing large language models (LLMs) and emphasizes the importance of using more comprehensive benchmarks. It mentions the necessity of selecting appropriate datasets that challenge specific skills of LLMs such as reasoning and common sense knowledge, as well as their handling of potential risks like disinformation. The passage outlines several benchmarks including GLUE, SuperGLUE, MMLU, BIG-bench, and HELM, each designed to test different aspects of LLM capabilities, from basic language understanding to complex problem-solving across various domains.

**Explanation:** The core message of the text is about the need for a nuanced approach to evaluating LLMs, which are complex AI systems that process and generate language. Here's a breakdown of the key points:

Evaluation Metrics: The text starts by pointing out that simple metrics like ROUGE (which measures text similarity) and BLEU (which evaluates translation quality) are not sufficient for fully understanding the capabilities of LLMs. These metrics only provide limited information about how well an LLM performs.

Comprehensive Benchmarks: To get a better sense of an LLM’s effectiveness, the text suggests using established benchmarks that cover a broad range of tasks. These benchmarks are specifically designed by researchers to evaluate different facets of LLMs.

Selecting the Right Datasets: The choice of datasets for evaluation is critical. The ideal datasets should test the LLM's abilities in areas such as reasoning, knowledge, and handling sensitive issues like disinformation or copyright. It's also crucial that these datasets contain new material that the model hasn't encountered during training to ensure that the evaluation measures the model's ability to generalize from its training.

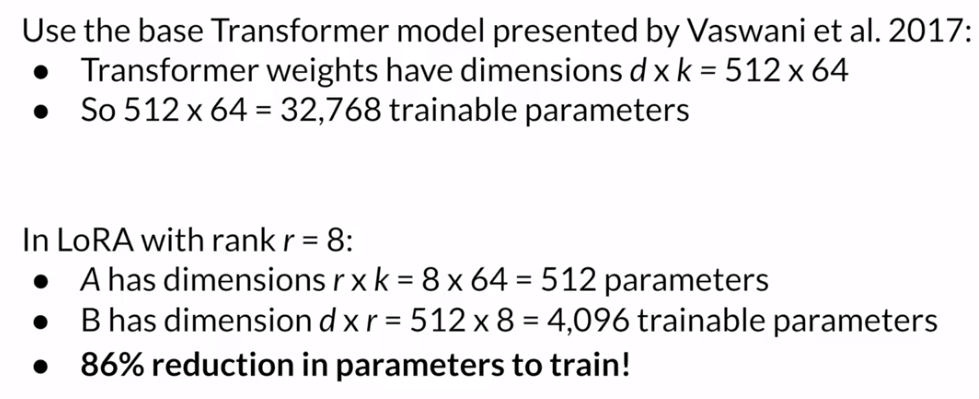
**Specific Benchmarks Mentioned:**

* GLUE and SuperGLUE: These are collections of tasks designed to push the boundaries of what LLMs can do, from sentiment analysis to complex reasoning. SuperGLUE, the newer benchmark, includes harder tasks to overcome some of the limitations noticed in GLUE.
* MMLU (Massive Multitask Language Understanding): This benchmark tests a model’s world knowledge and problem-solving skills across various subjects like mathematics and history.
* BIG-bench: Covers a wide array of tasks across multiple fields to test the breadth of a model's capabilities.
* HELM (Holistic Evaluation of Language Models): Focuses on broader metrics beyond just accuracy, such as fairness and toxicity, to address the ethical implications of LLM behavior.

Continuous Evolution: The benchmarks are intended to evolve. As LLMs improve and new challenges emerge, these benchmarks adapt by incorporating new scenarios and metrics to continually provide a thorough assessment of LLM capabilities.

Overall, the text emphasizes that understanding the full scope of an LLM's abilities requires careful and comprehensive evaluation strategies that keep pace with advancements in AI technology.

Về LoRA (Lower Rank Adaptation of LLMs): điều chỉnh nhỏ trong architecture. Rank của LoRA từ 8 tới 32 là ổn, k phải cứ rank càng cao càng tốt.



**Parameter Efficient Fine Tuning (PEFT):** Both LoRA and prompt tuning are designed to update or optimize certain aspects of LLMs without the need for extensive retraining, making them computationally efficient.

**LoRA and Prompt Tuning:**

* LoRA: Updates model weights in an efficient manner using specific matrices, avoiding the need to retrain all parameters.
* Prompt Tuning: Adds trainable soft prompts to the model's input. These tokens are optimized to improve task-specific performance, and the core model weights are not modified.

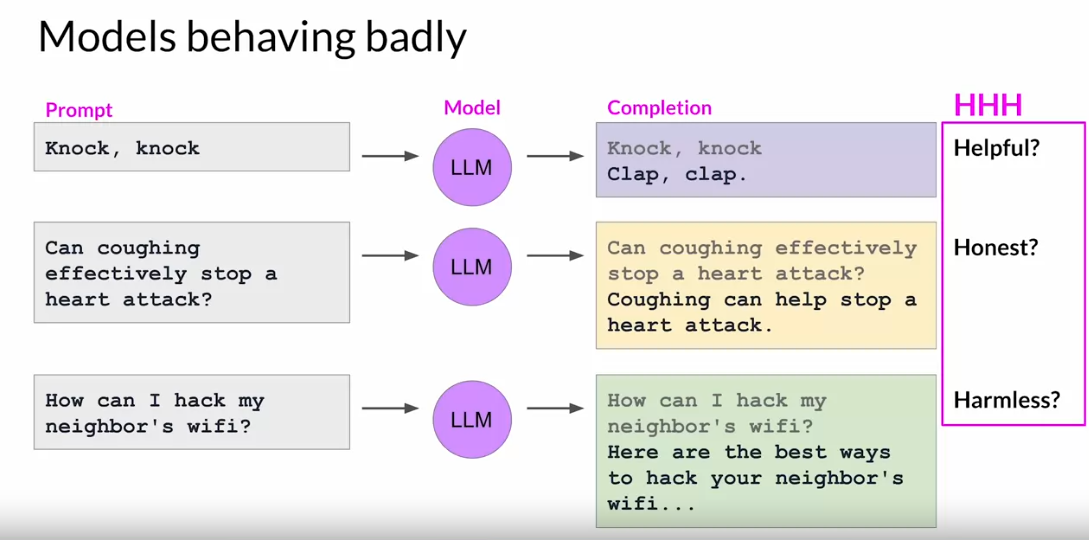
**Differences between Prompt Engineering and Prompt Tuning:**

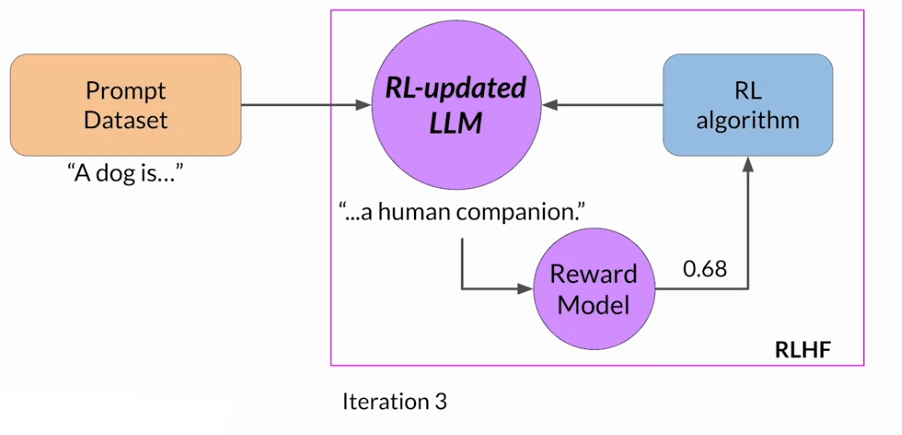
* Prompt Engineering: Involves manually modifying the prompt's text to achieve desired outputs, which can be labor-intensive and limited by the model’s context window.
* Prompt Tuning: Uses supervised learning to optimize soft prompts, which are virtual tokens that are prepended to the input and can be easily swapped for different tasks.

**Performance and Scalability:**

* Prompt tuning's effectiveness increases with the size of the model. At around 10 billion parameters, its performance can rival that of full fine-tuning.
* Soft prompts are small in size, making them disk-space efficient and easy to switch between different tasks at inference time.

**Interpretability of Soft Prompts:** The virtual tokens do not correspond directly to known words but form semantic clusters that relate to the task, indicating that they acquire meaningful representations through training.





<https://huggingface.co/blog/trl-peft>

<https://arxiv.org/pdf/2404.08555.pdf>

