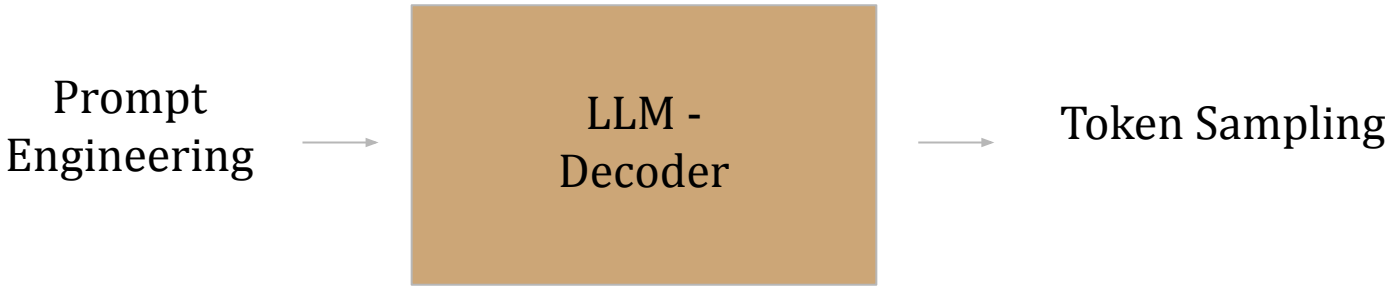
The LLM Journey

Vinija Jain
vinija.ai

Overview



Transformer Architecture

New Task: Input -> Output

Fine-tuning

PEFT

Introduction to Large Language Models

- What are LLMs?
  - Generative AI Models
  - Next Token Prediction
  - Can be fine-tuned for downstream tasks
  - Translation, Q&A, Code completion tasks, Text Classification
- Benefits:
  - Maintain context over longer text spans
  - Multilingual capabilities
  - Interactive and capable of dialogue/conversation
- Key LLMs: GPT-4, Claude, Llama 2
- HuggingFace Open LLM Leaderboard

Real World Problems LLMs Aim to Solve

1. Enhancing Communication
    a. Translate languages, generate and summarize content, and improve accessibility for non-native speakers or people with disabilities.
2. Improving Customer Service:
    a. Handling inquiries and providing information around the clock
3. Education and Learning
    a. LLMs can be used in personalized education tools, providing tutoring, answering student questions, and assisting in learning new languages.
4. Content Creation and Analysis
    a. Assist in writing, editing, and content generation for various fields like journalism, engineering, and science
5. Healthcare Support
    a. Aid in information dissemination, patient engagement, and even in interpreting medical literature or patient histories.
6. Data Analysis
    a. By analyzing large volumes of text, LLMs can extract insights, trends, and useful information for business strategy and decision-making.

Architecture: Transformer Decoder

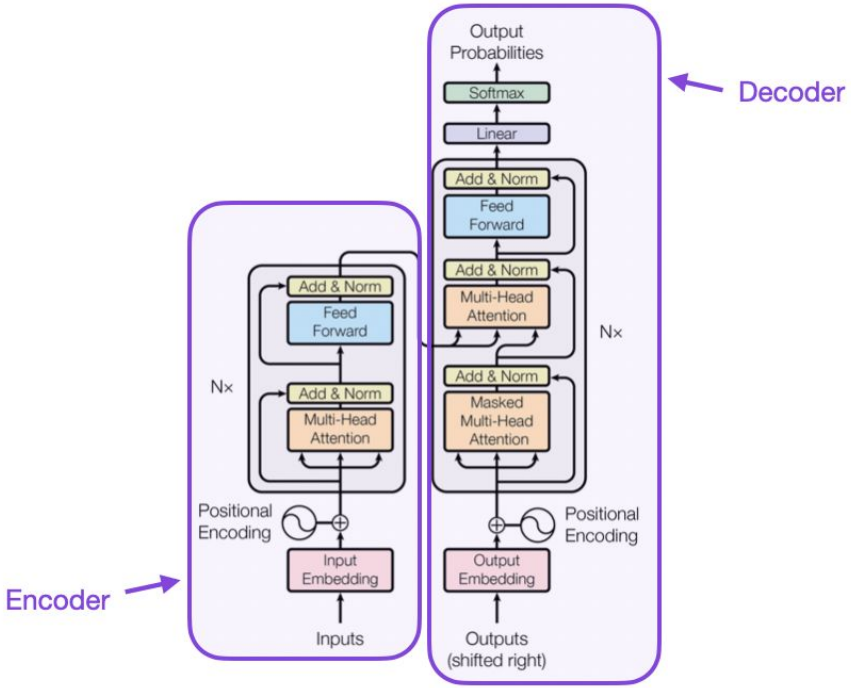


Figure 1: The Transformer - model architecture.

Source:
Attention Is All You Need by Vaswani et al.

Detailed information:
https://vinija.ai/nlp/transformers

Implement on a new task

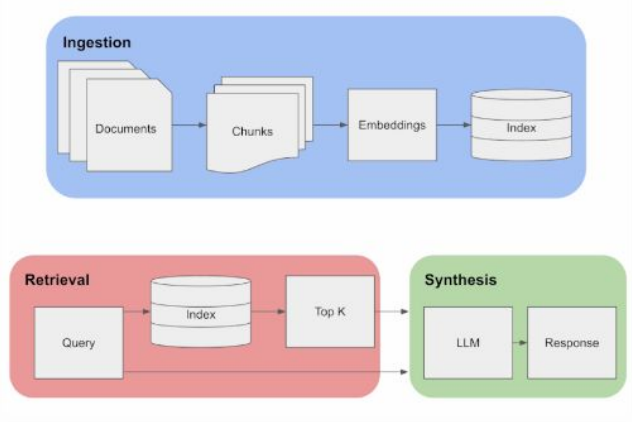
- Task: summarization of trending topics in the news
- Encoder (BERT) vs Decoder (GPT)?
  - Extractive:
    - Factually correct, not coherent in sentence structure, can be repetitive/ redundant and thus not convey meaning. It's like a highlighter.
  - Abstractive:
    - It's like a pen. Sentences will be fully formed and coherent and provide adequate summaries, but can have errors in them.

Input: Prompt Engineering

- Zero-shot
- Few-shot
- Chain of Thought
  - “Let’s think step by step”
- Chain of Verification:
  - Model drafts an initial response
  - Then plans verification questions to fact-check its draft
  - Answers those questions independently so the answers are not biased by other responses
  - Generates its final verified response

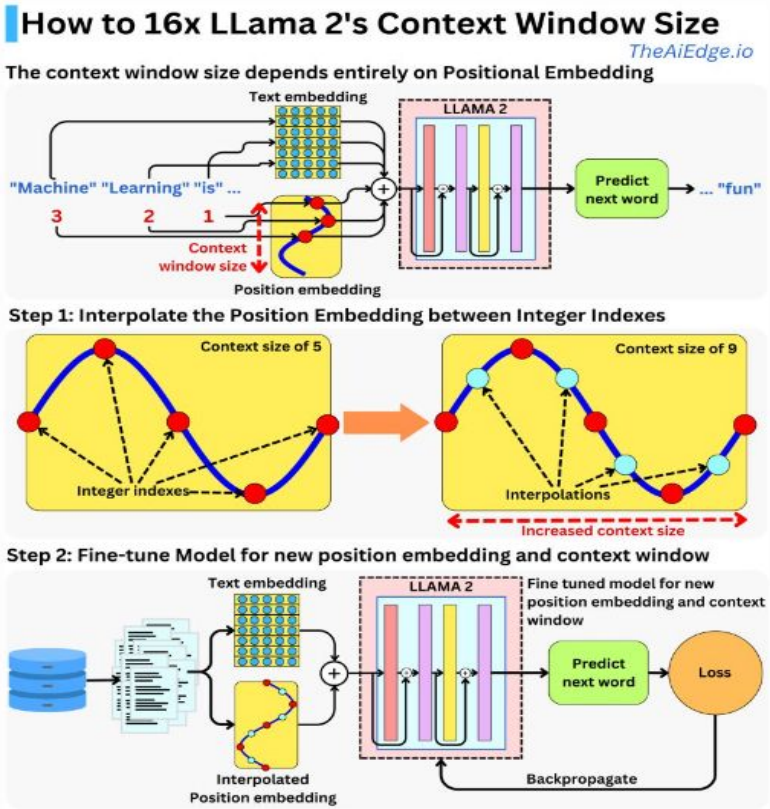
- **20+ more techniques can be found here:** https://vinija.ai/nlp/prompt-engineering/

- Retrieval Augmented Generation (RAG)



Context Length Extension

- Positional Interpolation
  - Instead of using integers for positions, this method utilizes non-integers values between the whole numbers.
  - LLama 2 can process text inputs that are much larger than its designed capacity or its original window size w/o reducing performance
- LongLora
  - Extends context length during finetuning by enhancing LoRA and introducing Shift Short Attention
  - S2-Attn approximates the full attention using short sparse attention within groups of tokens. It splits the sequence into groups, computes attention in each group, and shifts the groups in half the heads to allow information flow.



Source: TheAiEdge.io

Fine-tuning

- Full fine-tuning:
  - Further training the pre-trained model on a specific dataset or task
- Problems
  - Catastrophic Forgetting
  - Large amount of data
  - Overfitting
  - Compute

Parameter Efficient Fine-tuning (PEFT)

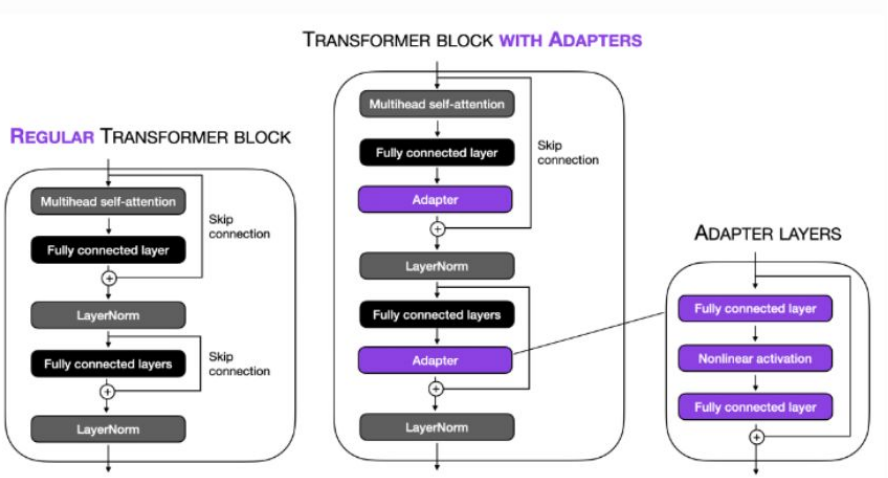
Prompt Modification

- Soft Prompt Tuning:
  - Introduces trainable parameters that are added to the models input embeddings
- Hard Prompt Tuning:
  - Simply alters the input prompt to the LLM fully
- Prefix Tuning:
  - Adds a trainable prefix parameters
  - Instead of adding a soft prompt to the model input, it prepends trainable parameters to the hidden states of all transformer blocks.
  - During fine-tuning, the LLM's original parameters are kept frozen while the prefix parameters are updated.

Parameter Efficient Fine Tuning

Adapters

- Adapter-based tuning simply inserts new modules called “adapter modules” between the layers of the pre-trained network.
- Image source: PEFT blog



Parameter Efficient Fine Tuning

Reparameterization

- **Low Rank Adaptation (LoRA)**
  - Adds low-rank matrices to existing weights to fine-tune large models
- **Quantized Low Rank Adaptation (QLoRA)**
  - The original model's weights that are quantized to 4-bit precision. The newly added LoRA weights are not quantized; they remain at a higher precision and are fine-tuned during the training process.
- **Quantization-Aware Low-Rank Adaptation (QALoRA)**
  - Quantizes both the new LoRA weights and the original model's weights.
  - LoRA weights are first merged with the full-precision model weights
  - Then jointly quantized to the targeted bit-width.

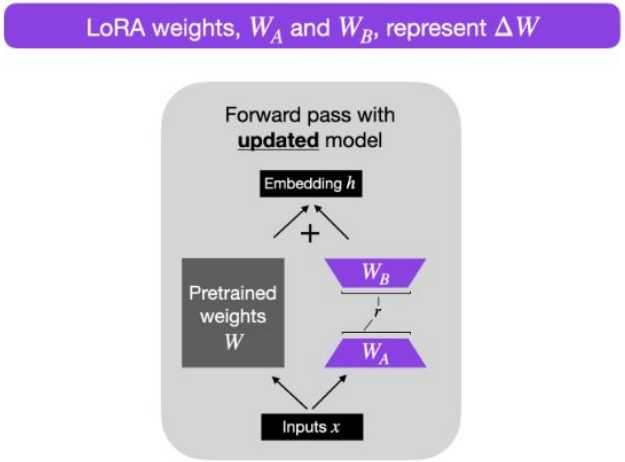


Image Source: https://sebastianraschka.com/blog/2023/llm-finetuning-lora.html

Token Sampling: Output

- Greedy Decoding
  - tokens with highest probability
- Exhaustive Search Decoding
  - Exhaustive search explores every possible combination of output sequences to find the best one.
- Beam Search
  - Explores multiple possibilities and retains the most likely one
- Top-K
  - Sample from a short list of top k tokens
- Top-P (Nucleus Sampling)
  - Dynamically sets the size of the short list based on a threshold
- Temperature
  - Not a sampling method but rather a hyperparameter on the softmax
  - Controls the distributions randomness
  - Higher temperature = more randomness in token selection

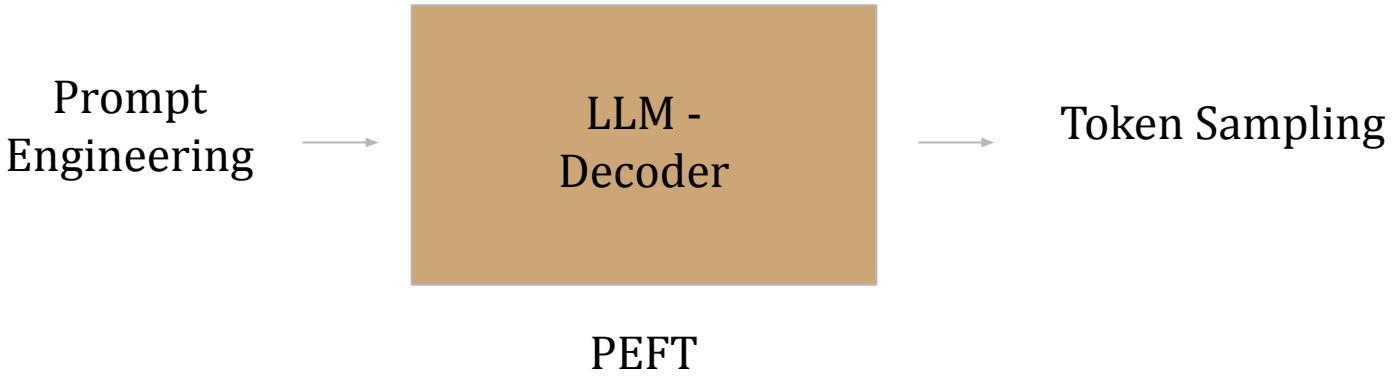
Hallucination Mitigation

1. Retrieval Augmented Generation
2. Human Moderators
3. Feedback Loops
4. Reinforcement Learning from Human Feedback
    a. Humans rank different generated response based on alignment with human expectations

Reversal Curse: LLMs trained on “A is B” fail to learn “B is A”

- Failure in logical deduction and generalization in LLMs, as they don't naturally learn the symmetry in relationships from their training data.
- Solution:
  - Augmented Training Data with Reversed Statements:
  - Along with the original statement, the training data is augmented with its reversed counterpart:
  - "Paris is the capital of France." and “France’s capital is Paris”
- Training Process:
  - The language model is then trained on this augmented dataset, which includes both the original statements and their reversed versions.
- Expected Outcome:
  - After being trained on such data, the model is expected to better handle reversible relationships.
- This method of data augmentation directly addresses the Reversal Curse by teaching the model that the order of elements in certain types of statements can be reversed without changing their meaning. This approach enhances the model's logical reasoning and generalization capabilities.

Final Thoughts



More Information: www.vinija.ai

- Word Vectors
- NLP Embeddings
- NLP Tasks
- Neural Networks
- Regularization
- Sampling
- Language Models
- AI Text Detection Techniques
- Tokenizer
- Conversational AI
- Machine Translation
- Word Sense Disambiguation
- Attention
- Knowledge Graphs
- NLP Architectures
- Transformer
- Preprocessing
- Fine-tuning
- Generative AI
- Metrics