

Transformers and LLMs (DRAFT)

Mark A. Austin

University of Maryland

austin@umd.edu

ENCE 688P, Spring Semester 2026

May 31, 2026

Overview

- 1 Definition of Transformers and Large Language Models (LLMs)
- 2 Computational Foundations
- 3 Transformers
- 4 Large Language Models
- 5 Advanced LLM Techniques
 - Technique 1: LLM-as-a-Judge
 - Technique 2: Retrieval Augmented Generation (RAG)
- 6 Transformer Applications
- 7 LLM Applications

Definition of Transformers and LLMs

Definition of Transformers

Transformer

Transformer models (2017) are neural networks that use a **technique called self-attention** to take into account the **context** of **elements in a sequence**, not just the elements themselves.

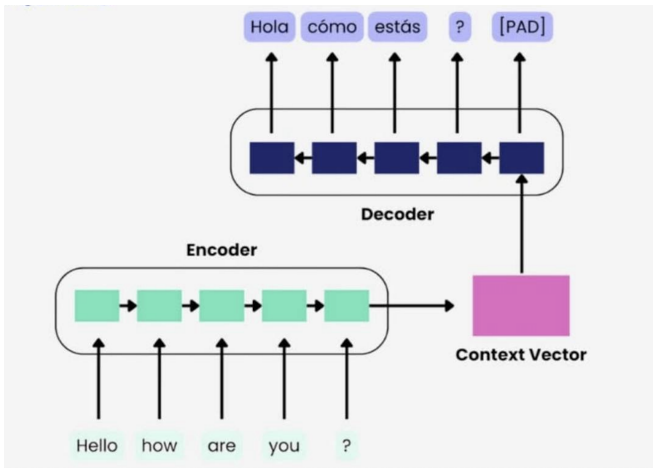
Benefits:

This ability makes transformers ideal for:

- Analyzing sentences/paragraphs of text, and
- Translating languages and understanding complex sentences.
- Computer vision (image classification, object detection, vision transformers).
- Speech recognition, transcription, voice assistants.
- Recommendation systems.

Definition of Transformers

Transformer Neural Networks:



Types of Transformer Model

Encoder-Only: (e.g., BERT).

- Ideal for understanding text, classification, embeddings.

Decoder-Only: (e.g., GPT).

- Task support includes text generation, conversation, creative writing.

Encoder-Decoder: (e.g., T5).

- Focus on translation, summarization and sequence tasks.

Transformer Mechanisms

Self-Attention

In **self-attention**, each word looks at every other word to understand its context and importance.

Multi-Head Attention

Multi-head attention mechanisms learn different relationships (syntax, meaning, position).

Transformer Models

HOW IT WORKS (STEP BY STEP)



1 Input Embedding + Positional Encoding
 Words are converted into vectors and positional information is added to understand order.



2 Self-Attention
 Each word looks at every other word to understand its context and importance.



3 Multi-Head Attention
 Multiple attention heads learn different relationships (syntax, meaning, position).

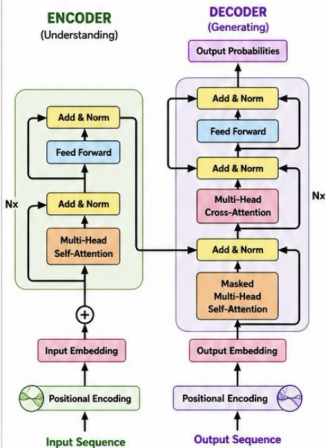


4 Feedforward Network
 Refines the information with fully connected layers and non-linearity.



5 Output
 The model produces predictions or generates new sequences.

TRANSFORMER ARCHITECTURE



KEY COMPONENTS



Self-Attention
 Calculates how much focus each word should pay to others.



Multi-Head Attention
 Allows the model to capture different types of relationships.



Feedforward Network
 Adds non-linearity and enhances the representations.



Residual Connection
 Helps training deep networks by avoiding gradient vanishing.



Layer Normalization
 Stabilizes and speeds up the training process.



Positional Encoding
 Adds position info since transformers process in parallel.

Transformer Models

TRANSFORMERS vs TRADITIONAL MODELS		
FEATURE	RNN / LSTM	TRANSFORMER
Processing	Sequential	Parallel
Speed	Slow	Fast
Long Dependencies	Weak	Strong
Scalability	Limited	High
Context Awareness	Limited	Global
Training Efficiency	Low	High

TYPES OF TRANSFORMER MODELS		
	ENCODER-ONLY (e.g., BERT)	Great for understanding text, classification, embeddings.
	DECODER-ONLY (e.g., GPT)	Great for text generation, conversation, creative writing.
	ENCODER-DECODER (e.g., T5)	Great for translation, summarization, and sequence tasks.

REAL-WORLD APPLICATIONS

 NLP Translation, Chatbots, Summarization, Sentiment Analysis	 COMPUTER VISION Image Classification, Object Detection, Vision Transformers	 SPEECH Speech Recognition, Transcription, Voice Assistants	 RECOMMENDATION SYSTEMS Personalized Recommendations, User Behavior Modeling	 CODE & DATA Code Generation, Bug Detection, Data Analysis
--	---	--	---	---

WHY TRANSFORMERS REVOLUTIONIZED AI

 PARALLEL PROCESSING Trains faster and handles more data efficiently.	 BETTER CONTEXT UNDERSTANDING Captures long-range dependencies with ease.	 SCALABILITY Performs exceptionally well with large datasets.	 TRANSFER LEARNING Pre-trained models can be fine-tuned for many tasks.	 VERSATILITY Works across text, images, audio, code and more.
--	--	--	--	--

BERT Models

BERT vs GPT		
Feature	BERT (Encoder)	GPT (Decoder)
Architecture	Encoder-only	Decoder-only
Training Direction	Bidirectional	Left-to-right
Best For	Understanding	Generation
Applications	Classification, QA, NER, Sentiment Analysis	Text Generation, Summarization, Chatbots
Context Handling	Strong Comprehension	Strong Generation

BERT VARIANTS	
🌟 RoBERTa	Improved training strategy, larger datasets
🌟 DistilBERT	Smaller, faster and lighter version
🌟 ALBERT	Parameter reduction, more efficient
🌟 BioBERT	Designed for biomedical NLP tasks
🌟 TinyBERT	Lightweight for mobile & deployment

BERT ADVANTAGES
✔ Deep context understanding
✔ High accuracy on NLP benchmarks
✔ Transfer learning makes it powerful
✔ Flexible fine-tuning for many tasks
✔ Better semantic understanding

LIMITATIONS
✘ Computationally expensive
✘ High memory usage
✘ Slow inference
✘ Limited input sequence length

```

PYTHON EXAMPLE (Hugging Face)

from transformers import BertTokenizer, BertModel
import torch

# Load tokenizer and model
tokenizer = BertTokenizer.from_pretrained(
    'bert-base-uncased')
model = BertModel.from_pretrained(
    'bert-base-uncased')

# Input text
text = "BERT is transforming NLP."
# Tokenize input
inputs = tokenizer(text, return_tensors='pt')
# Generate embeddings
outputs = model(**inputs)
print(outputs.last_hidden_state.shape)

```

This code loads a pre-trained BERT model and generates contextual embeddings for the text.

IMPACT ON SEO & DIGITAL MARKETING



BERT helped Google understand search queries better, leading to smarter search results.

SEO Shift:

- From Keyword Stuffing → To User Intent
- From Exact Match → To Semantic Relevance
- From Short Content → To Quality & Context-Rich Content

THE FUTURE OF BERT & NLP



- More Efficient Models
- Multimodal Understanding
- Real-time & Context-aware AI
- Domain-Specific Adaptations
- Foundation for Advanced LLMs (GPT, TS, PaLM, Gemini, LLaMA)

BERT revolutionized NLP by enabling machines to understand language like never before. It remains a foundation of modern AI and continues shaping the future of intelligent applications.



Definition of Large Language Model (LLM)













Large Language Model (LLM)

A LLM is an advanced computational model designed to understand, generate human language, and perform a wide range of tasks. LLMs use **deep learning techniques** to **learn statistical relationships** between **words** and **phrases**, enabling them to perform tasks like **language translation**, **text summarization**, and **question answering**.

Examples of LLM Frameworks:

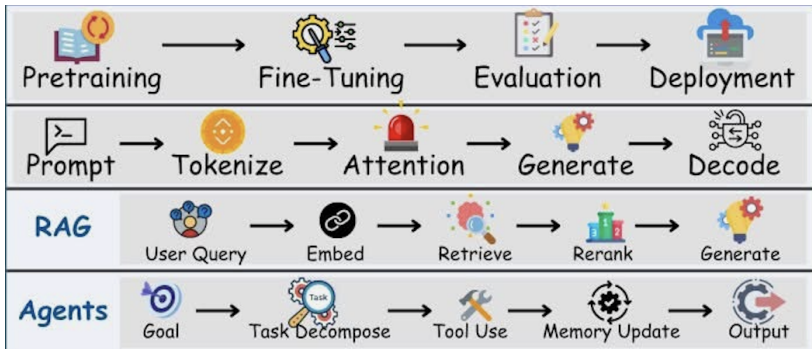
- OpenAI: GPT-4, GPT 5-5.
- Google: BERT and PaLM,
- Meta: LLaMA.9.

Types of LLM (by Function)

Type	Example	Purpose
Autoregressive	 GPT  LLaMA	Text generation (next-token prediction)
Masked Language	 BERT	Understanding context (classification, NER)
Multimodal	 Gemini  Flamingo	Text + Image/Audio/Video processing
Instruction-Tuned	 Open Chat  Alpaca	Better alignment with user commands
MoE (Sparse)	 Mixtral	Activates select expert layers per input
Agentic Models	 AutoGPT  CrewAI	Combine tools, reasoning, and planning
Small Language Models (SLMs)	 Phi  TinyLLaMA	Efficient inference on limited compute

Source: LLM cheatsheet.

LLM Workflows



Source: LLM cheatsheet.

Top Open-Source LLMs (2025)



LLaMA 3 (Meta) -
General-purpose, state-
of-the-art



**DeepSeek / DeepSeek-
Coder** - Code generation
and reasoning



Mistral / Mixtral -
Sparse MoE, strong
open competitor



**OpenChat / Zephyr /
NeuralBeagle** - Instruction-
tuned, RLHF-aligned



Gemma (Google) -
Lightweight, powerful,
fine-tuning friendly

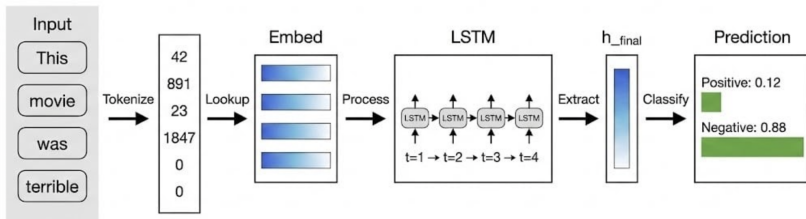


Phi-3 (Microsoft) -
Efficient small model for
edge inference

Source: LLM cheatsheet.

LLM Applications

Text Processing Pipeline:



LLM Applications

Image-to-Text Transformation:

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



LLM Weaknesses

Weaknesses:

- No Persistent Memory.
- No Reasoning Ability.
- Limited Planning.

Tokenization

Tokenization

Type	Pros	Cons	Illustration
Word	<ul style="list-style-type: none"> • Easy to interpret • Short sequence 	<ul style="list-style-type: none"> • Large vocabulary size • Word variations not handled 	<code>teddy bear</code>
Subword	<ul style="list-style-type: none"> • Word roots leveraged • Intuitive embeddings 	<ul style="list-style-type: none"> • Increased sequence length • Tokenization more complex 	<code>ted ##dy bear</code>
Character Byte	<ul style="list-style-type: none"> • No out-of-vocabulary concerns • Small vocabulary size 	<ul style="list-style-type: none"> • Much longer sequence length • Patterns hard to interpret because too low-level 	<code>t e d d y b e a r</code>

Embeddings

Embedding

An **embedding** is a numerical representation of an element (e.g., token, sentence) that is characterized by vector $x \in \mathbb{R}^n$.

Document Embeddings

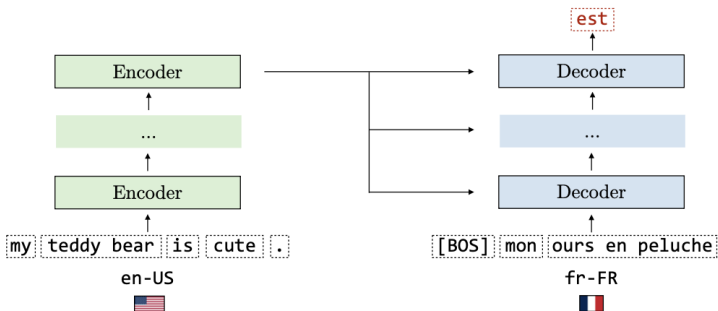
Embedding Operations

Dimension Reduction:

Transformer Definition

Transformer

A **transformer** model uses **self-attention mechanisms** and a composition of encoders and decoders to generate token-aware, position-aware, and context-aware embeddings.

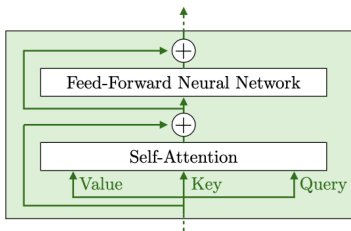


Transformer Definition

Encoders compute meaningful embeddings of the input.
Decoders then predict the next token in the sequence.

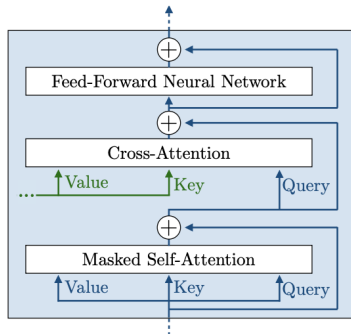
Encoder

Encoded embeddings encapsulate meaning of input



Decoder

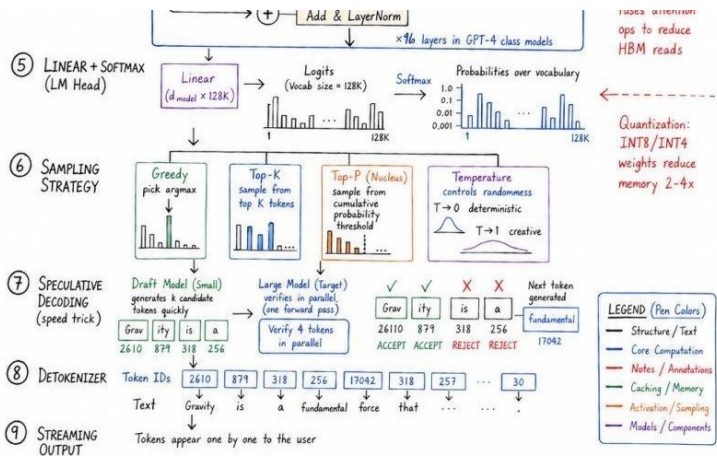
Decoded embeddings encapsulate meaning of both input and output predicted so far



Step-by-Step Procedure

- Read input.
- Tokenizer (process input tokens in parallel – compute-bound).
- Embedding layer.
- Transformer Block (multi-head self attention; KV cache).
- Linear and softmax transformations (estimate logits and probabilities over vocabulary).
- Sampling strategy (e.g., greedy, top-K tokens, sample from cumulative probability threshold)..
- Speculative decoding (speed trick).
- Detokenizer (generate tokens one-by-one).
- Stream output (to end user).

Step-by-Step Procedure



Attention (part02)

Multi-Head Attention

Transformer Architecture

Encoder/Decoder Components

Variants (Encoder Only)

Bidirectional Encoder Repr from Transformers (BERT)

Variants (Decoder Only)

Generative Pre-trained Transformer (GPT)

Definition

Prompt Engineering

Prompt Engineering

Prompt Engineering is the art and science of designing, refining and optimizing inputs (prompts) to guide generative AI models toward producing accurate, relevant and high-quality outputs.

Importance:

- Prompt engineering influences the quality, relevance and accuracy of Generative AI outputs.
- A well-crafted prompt ensures that AI comprehends the user's intents, a prerequisite for producing meaningful responses.

Prompt Engineering

How it Works:

- Prompt engineering acts as a bridge between human intent and AI understanding, transforming vague requests into structured commands.

Common Use Cases:

- Automated code generation; generate text summary; data processing and extraction; logical reasoning.

Examples:

- Write a python program to compute the factorial of a number.
- Translate the following python code to Java.
- Debug the Java code and explain why it throws an exception.

Prompt Engineering

Prompting Techniques:

- **Zero-shot prompting:** Provides the model with a direct instruction or question without any additional context or examples.
- **Few-shot prompting:** Provide the model with one or more examples of the desired input-output pairs before presenting the actual prompt.
- **Chain-of-thought prompting:** Encourages the model to break down complex reasoning into a series of intermediate steps, leading to a more comprehensive and detailed output.

Prompting)

Standard and Chain-of-Thought Prompting:

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Prompting

Question

Tracy used a piece of wire 4 feet long to support tomato plants in the garden. The wire was cut into pieces 6 inches long. How many pieces did she obtain?

62B Model Output

She cut the wire into 6 inch pieces. This means she got $4 * 6 = 24$ pieces. The answer is 24.
(Incorrect: semantic understanding error)

540B Model Output

The wire was 4 feet long. This means it was $4 * 12 = 48$ inches long. It was cut into pieces 6 inches long. This means she obtained $48 / 6 = 8$ pieces. The answer is 8.
(Correct)

Question

Tom's ship can travel at 10 miles per hour. He is sailing from 1 to 4 PM. He then travels back at a rate of 6 mph. How long does it take him to get back?

62B Model Output

He travels from 1 to 4 PM. That means he travels for 3 hours. He travels back at 6 mph. That means he travels for $3 * 6 = 18$ miles. The answer is 18. (Incorrect: semantic understanding error)

540B Model Output

He travels at 10 mph from 1 to 4 PM. This means he travels 3 hours. 3 hours at 10 mph means he travels $3 * 10 = 30$ miles. He then travels back at 6 mph. This means he travels 6 miles per hour. He has to travel 30 miles, so it takes him $30 / 6 = 5$ hours. The answer is 5.
(Correct)

Prompting

Question

Stephen placed an online order for groceries. His final bill came to \$40.00. Because this was through a delivery vendor, they tacked on a 25% fee to his final total and charged him \$3.00 in delivery fees. Stephen also added a \$4.00 tip. After the extra fees, what was the final price of Stephen's groceries?

62B Model Output

The final price of Stephen's groceries was \$40.00. The delivery vendor added 25% to the final total, so the final total was 125% of \$40.00. This means the final total was $125\% * \$40.00 = \50.00 . The delivery vendor also charged \$3.00 in delivery fees. So the final price of Stephen's groceries was $\$50.00 + \$3.00 = \$53.00$. The answer is \$53.00. **(Incorrect: one step missing error)**

540B Model Output

Stephen's bill was \$40.00. The delivery vendor added 25% to the bill. This means the bill was increased by 25% of \$40.00. This is $0.25 * \$40.00 = \10.00 . The delivery fee was \$3.00. The tip was \$4.00. So the total bill was $\$40.00 + \$10.00 + \$3.00 + 540b \text{ model } \$4.00 = \$57.00$. The answer is 57. **(Correct)**

Question

There are four schools competing at a basketball tournament. Each school has sent a girls' basketball team and a boys' basketball team and each team has 5 players each. Each school has also sent a coach for each team. In total, how many people have all of the schools sent?

62B Model Output

There are 4 schools. Each school has sent a girls' basketball team and a boys' basketball team. Each team has 5 players each. That means there are $4 * 2 * 5 = 40$ players. Each team has a coach. That means there are $40 + 4 = 44$ coaches. The answer is 44. **(Incorrect: one step missing error)**

540B Model Output

Each school has sent 2 teams, each with 5 players. This means each school has sent $2 * 5 = 10$ players. Each school has also sent 2 coaches. This means each school has sent $10 + 2 = 12$ people. There are 4 schools, so in total all of the schools have sent $4 * 12 = 48$ people. The answer is 48. **(Correct)**

Strategies for Writing Better Prompts

Set clear goals and objectives:

- **Tactic:** Use action verbs to specify the desired action (e.g., write a bulleted list item that summarizes the key findings of the attached research paper).
- **Tactic:** Define the desired length and format of the output (e.g., Compose a 200-word abstract summarizing the key findings of the research paper.)
- **Tactic:** Specify the target audience (e.g., Write a 400-word essay targeting young teens concerned with global climate warming.)

Source: Wikipedia entry for prompt engineering

Strategies for Writing Better Prompts

Provide context and background information:

- **Tactic:** **Include relevant facts and data.** (e.g., Given that temperatures in the Tasman sea have risen 1 degree Celcius over the past decade, describe the likelihood of long term impacts caused by global warming.)
- **Tactic:** **Reference specific sources or documents.** (e.g., Based on the attached document, generate a forecast for the companies profitability over the next 18 months.)
- **Tactic:** **Define key concepts and terms.** (e.g., Explain the key ideas in quantum computing using terms suitable for a non-technical audience.)

Strategies for Writing Better Prompts

Use Few-Shot Prompting:

- **Tactic:** Provide a few examples of input-output pairs. (e.g., input: cat, output: A small hairy animal with whiskers, input: kiwi, output: A small flightless bird found in New Zealand.)
- **Tactic:** Demonstrate the desired style or tone. (e.g., (humorous) the dignitary delivered a speech that was so dull it could cure insomnia, (formal) the dignitary delivered a speech that was informative and entertaining.)
- **Tactic:** Show the desired level of detail. (e.g., (brief) The movie was about a young boy who has lunch with an alien, (detailed) The science fiction movie follows the adventures of Casper, who meets and lunch with a friendly alien.)

Strategies for Writing Better Prompts

Be Specific:

- **Tactic:** Use precise language and avoid ambiguity. (e.g., Write a persuasive essay arguing for regulated approaches to sustainability.)
- **Tactic:** Whenever possible, quantify the requests. (e.g., Write a poem with 14 lines.)
- **Tactic:** Break down complex tasks into simpler tasks. (e.g., Instead of saying develop a meeting plan, say: 1. Identify the target audience, 2 Develop key marketing messages, 3 Select appropriate marketing channels.)

Strategies for Writing Better Prompts

Iterate and Experiment:

- **Tactic:** Try a variety of phrasings and keywords. (e.g., Rephrase the prompt using a variety of synonyms or alternate sentence structures.)
- **Tactic:** Adjust the level of detail and specificity. (e.g., Add or remove information to fine-tune the output.)
- **Tactic:** Experiment with different response lengths. (e.g., Experiment with short/long output lengths to find the desired balance.)

Strategies for Writing Better Prompts

Leverage CoT Prompting:

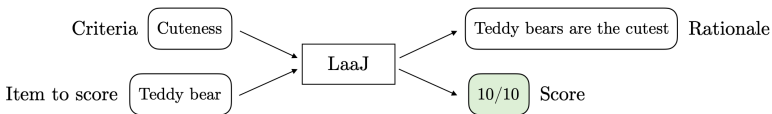
- **Tactic:** Encourage multi-step reasoning. (e.g., Solve the problem step-by-step.)
- **Tactic:** Ask the model to explain its reasoning strategy. (e.g., Explain the thought process in determining the sentiment review of a movie, for example, the acting was superb, but the plot too predictable.)
- **Tactic:** Guide the model through a logical sequence of thought. (e.g., To classify the e-mail as spam, consider the following: 1. Is the sender known?, 2. Does the subject line contain suspicious words?, 3. Is the e-mail offering something too good to be true?)

Advanced LLM Techniques

Technique 1: LLM-as-a-Judge

LLM-as-a-Judge

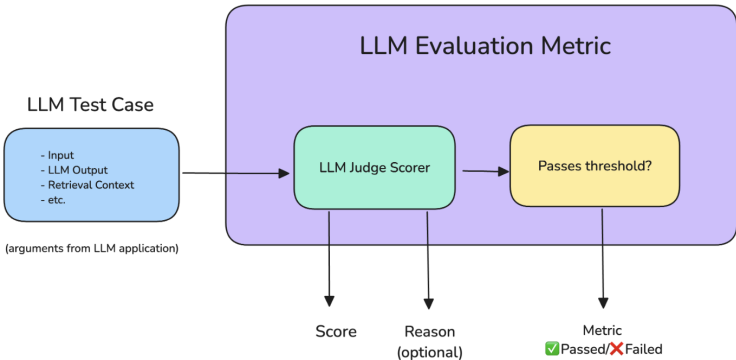
LLM-as-a-Judge is an **evaluation framework** where a high-performing language model (e.g., GPT 5.5) assesses the **outputs of other LLMs** or AI Agents based on specific criteria.



The approach aims to **replace manual scoring** by humans with **automated scoring** against **user-defined evaluation criteria**.

Technique 1: LLM-as-a-Judge

Evaluation Metrics/Criteria:



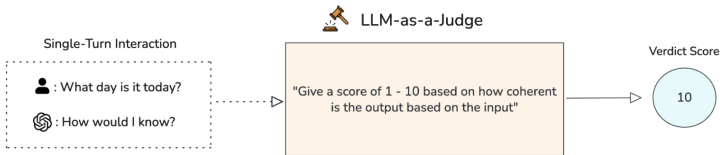
Technique 1: LLM-as-a-Judge

Evaluation Criteria:

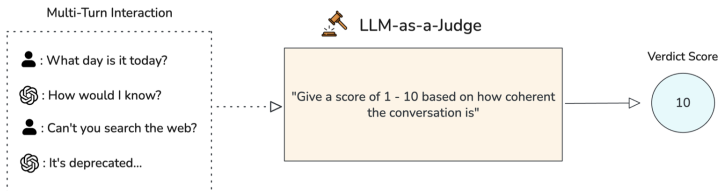
- **Answer Relevancy:** Did the response actually address the user's request?
- **Helpfulness:** Was the answer genuinely useful? Or just pretty sentences, no substance?
- **Faithfulness:** Did it stick to the facts? Or did it wander into hallucinations?
- **Bias:** Was the answer balanced, or did some skew creep in?
- **Correctness:** Was the answer accurate start to finish?

Technique 1: LLM-as-a-Judge

Single-Turn LLM-as-a-Judge

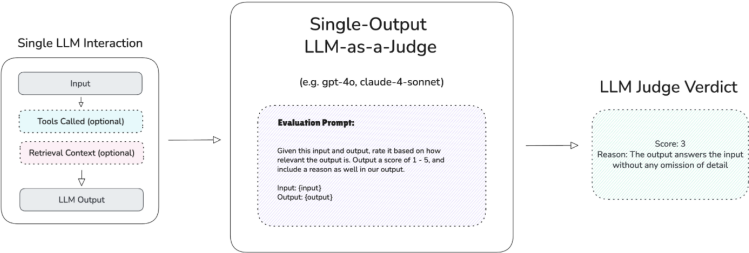


Multi-Turn LLM-as-a-Judge



Technique 1: LLM-as-a-Judge

Single-Output LLM as a Judge:



Single-Output LLM-as-a-Judge

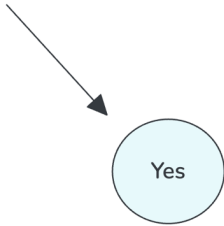
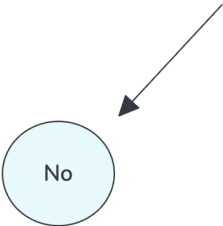
Technique 1: LLM-as-a-Judge

Binary LLM-as-a-Judge



Binary LLM-as-a-Judge

"Is the output coherent based on the given input? Only output 'Yes' or 'No' "



Technique 1: LLM-as-a-Judge

Multi-Turn Dialogues between a User and two AI Assistants:

Question: If the FED buys bonds in the secondary market (A) the money supply will increase. (B) the money supply will decrease. (C) the money supply will not be affected. (D) the discount rate would be affected. Output the answer only.

Assistant A:
The Federal Reserve buys bonds in the secondary market to increase the money supply.

Assistant B:
(A) the money supply will increase.

Follow-up Question: How does it affect my daily life? Give 3 examples.

Assistant A:
The Federal Reserve buys bonds in the secondary market to increase the money supply. This affects your daily life by:
1. Increasing the money supply.
2. Increasing the money supply and lowering the discount rate.
3. Increasing the money supply and lowering the discount rate and increasing the money supply.

Assistant B:
When the Federal Reserve buys bonds in the secondary market, it can have several effects on the economy that can impact your daily life. Here are three examples:
1. Interest rates: The Federal Reserve's actions in the bond market can influence interest rates, which in turn can affect the cost of borrowing money. For example, if the Federal Reserve buys bonds and drives up demand for them, interest rates on things like mortgages, car loans, and credit cards may go up, making it more expensive to borrow money.
2. Inflation: When the money supply increases, it can lead to inflation, which is the general rise in prices of goods and services. This can impact your daily life by making it more expensive to buy things like groceries, gas, and other necessities.
3. Employment: The Federal Reserve's actions in the bond market can also have an impact on employment. For example, if the Federal Reserve's actions lead to economic growth and increased demand for goods and services, it can create more job opportunities and boost employment.

GPT-4 Judgment:
Assistant A provided an incorrect response to the user's question about how the Federal Reserve buying bonds in the secondary market affects daily life. The answer given is repetitive and lacks clear examples of how the action impacts daily life. On the other hand, Assistant B provided a relevant and accurate response to the user's question about the Federal Reserve buying bonds. The answer includes three clear examples of how the action impacts daily life, such as interest rates, inflation, and employment.
Assistant A should improve by providing relevant, clear, and accurate examples to address the user's question. Assistant B's response is more helpful, relevant, and detailed.

Figure 1: Multi-turn dialogues between a user and two AI assistants—LLaMA-13B (Assistant A) and Vicuna-13B (Assistant B)—initiated by a question from the MMLU benchmark and a follow-up

Technique 1: LLM-as-a-Judge

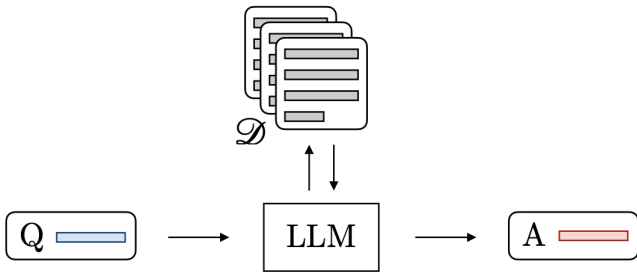
Weaknesses of LLM as a Judge:

- **Can't Make Up Their Minds:** Their scores are non-deterministic, which means that for a given LLM output they are evaluating, the scores might be different depending on the time of day. You'll need a good way such as **DAG** to make them deterministic if you want to rely fully on them.
- **Narcissistic Bias:** It has been shown that LLMs may favor the answers generated by themselves. We use the word "may" because [research has shown](#) that although GPT-4 and Claude-v1 favors itself with a 10% and 25% higher win rate respectively, they also favor other models and GPT-3.5 does not favor itself.
- **More is More:** We humans all know the phrase less is more, but LLM judges tend to prefer more verbose text over more concise ones. This is a problem in LLM evaluation because LLM computed evaluation scores might not accurately reflect the quality of the LLM generated text.
- **Not-so-Fine-Grained Evaluation Scores:** LLMs can be reliable judges when making high-level decisions, such as determining binary factual correctness or rating generated text on a simple 1—5 scale. However, as the scoring scale becomes more detailed with finer intervals, LLMs are more likely to produce arbitrary scores, making their judgments less reliable and more prone to randomness.
- **Position Bias:** When using LLM judges for pairwise comparisons, it has been shown that LLMs such as GPT-4 generally prefer the first generated LLM output over the second one.

Technique 2: Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation

Given a knowledge base and a question, a **retriever** fetches the most relevant documents, then **augments** the prompt with the information before **generating** the output.



Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG)

Transformer

Applications

LLM

Applications

References

- Amidi A., and Amidi S., Super Study Guide: Transformers and Large Language Models (First Edition), ISBN 979-8-836693-31-2, 2024.
- Lakshmanan V., and Hapke H., Generative AI Design Patterns: Solutions to Common Challenges when building GenAI Agents and Applications, O'Reilly, 2026.
- Negro A., Futia G., Kus V., and Montagna F., Knowledge Graphs and LLMs in Action, Manning Publications, 2026.
- Raieli S., and Luculano G., Building AI Agents with LLMs, RAG and Knowledge Graphs, Packt Publishing, 2025.