SAMPLE EFFICIENT LLM EVALUATION AND PREFERENCE MEASUREMENT

Minha Hwang

AGENDA

LLM 101: 3 TRAINING STAGES

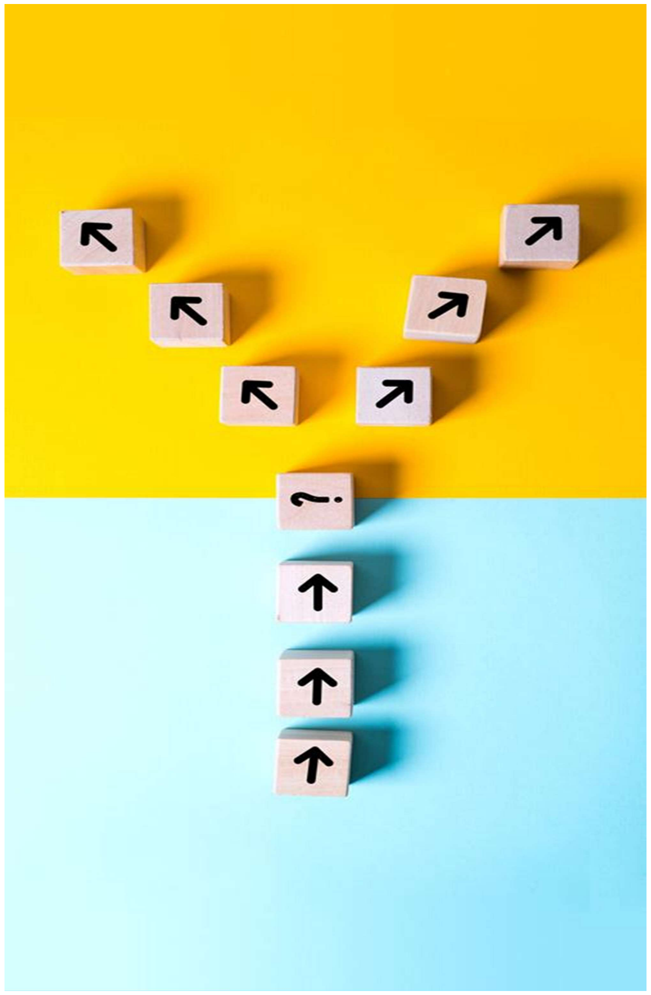
PRE-TRAINING EVAL

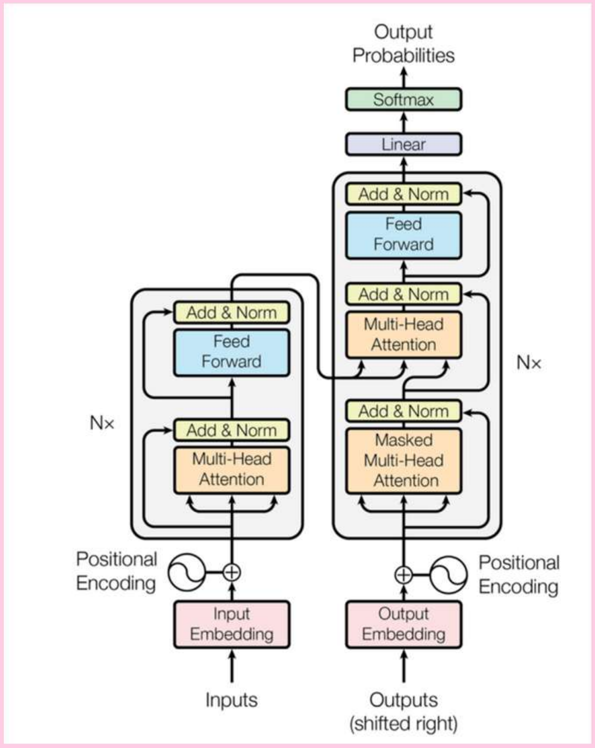
POST-TRAINING EVAL

RAG EVAL

MEASUREMENT THEORY

MAX DIFF





LLM 101: 3 TRAINING STAGES

AUTOREGRESSIVE LANGUAGE MODELS: PROBABILITY KERNEL

- A language model is a **probability kernel μ** given a prefix of words: $\mu: X \rightarrow Pr(Y)$
  - Stochastic in nature: **A same prefix X** can give a **random output** sampled from a probability distribution μ_X(i.e., generative) → A key reason for factual inaccuracy, inconsistency or hallucination (making stuff up)

- A language model calculates $Pr(s)$ given a sequence of words: $s = (w_1, w_2, \ldots\ldots., w_{T-1}, w_T)$

- An autoregressive language model calculates this **conditional on a previous sequence of words**:

$$Pr(s) = Pr(w_1, w_2, \ldots\ldots., w_{T-1}, w_T)$$
$$= \prod_{t=1}^{T} Pr(w_t | w_1, w_2, \ldots\ldots., w_{t-1})$$

  - **Next-word prediction**: Given a prefix $(w_1, w_2, \ldots\ldots., w_{t-1})$, calculate the probability of the next word w_t (Conceptually same to time series with path dependence)

Source: Prof. Kyunghyun Cho

AUTOREGRESSIVE LANGUAGE MODELS: SIMPLE EXAMPLE

- 4-word sentence example: "I am a student"

$$Pr(s) = Pr(w_1, w_2, w_3, w_4) = Pr(w_1) \times Pr(w_2|w_1) \times Pr(w_3|w_1, w_2) \times Pr(w_4|w_1, w_2, w_3)$$

- All you need is **"counting"** (if there are large amounts of data)

$$Pr(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)} \longrightarrow \text{2-grams (Bigrams)}$$

$$Pr(w_3|w_1, w_2) = \frac{count(w_1, w_2, w_3)}{count(w_1, w_2)} \longrightarrow \text{3-grams (Trigrams)}$$

$$Pr(w_4|w_1, w_2, w_3) = \frac{count(w_1, w_2, w_3, w_4)}{count(w_1, w_2, w_3)} \longrightarrow \text{4-grams}$$

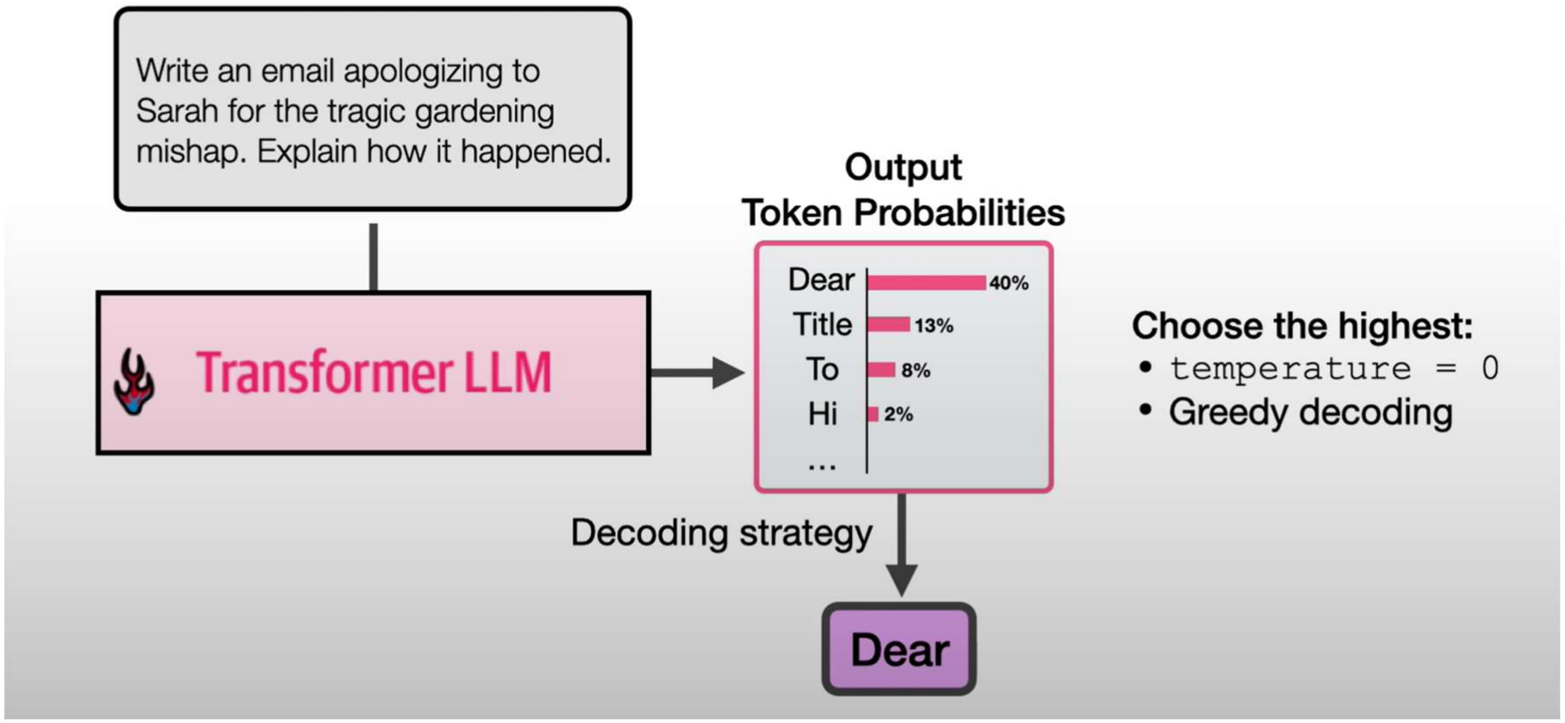
- Problems:
  - This requires **a lot of space (RAM)**
  - Count-based language models **cannot generalize:** A certain sentence **does not appear** in the corpus

Source: Prof. Kyunghyun Cho

TRANSFORMER LLM: INPUT AND OUTPUT

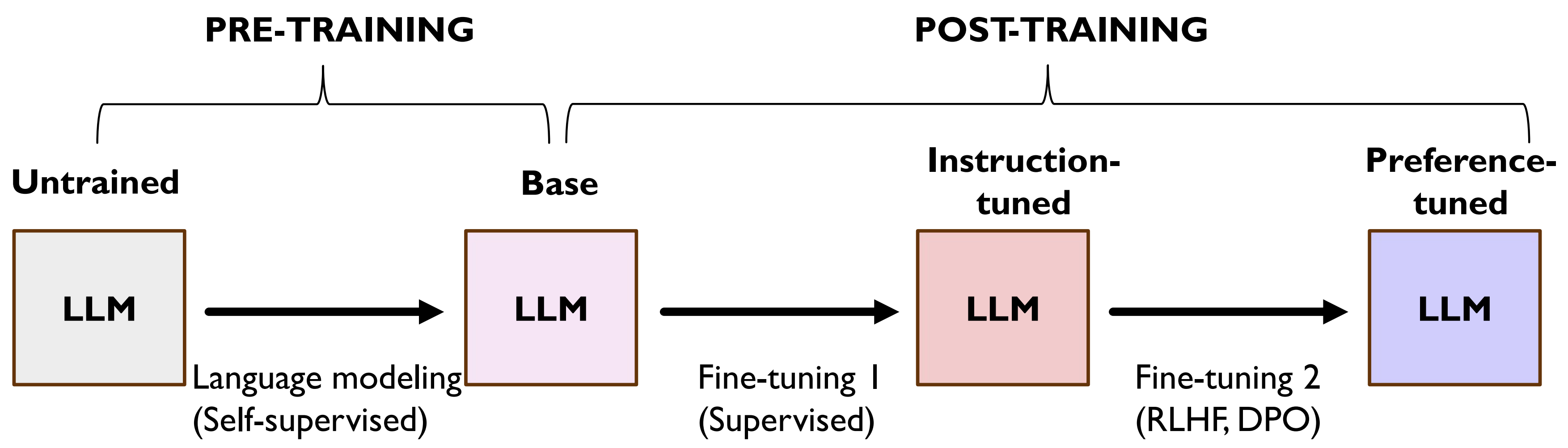
Input

Output


Write an email apologizing to Sarah for the tragic gardening mishap. Explain how it happened.
Transformer LLM
Output Token Probabilities
Dear 40%
Title 13%
To 8%
Hi 2%
...
Choose the highest:
• temperature = 0
• Greedy decoding
Decoding strategy
Dear

Source: How Transformer LLMs Work - DeepLearning.AI

LLM TRAINING: 3 STEP RECIPE


PRE-TRAINING
POST-TRAINING
Untrained
Base
Instruction-tuned
Preference-tuned
LLM
LLM
LLM
LLM
Language modeling (Self-supervised)
Fine-tuning 1 (Supervised)
Fine-tuning 2 (RLHF, DPO)

Pretraining Dataset:
Predict **next word**, based on text training data

Instruction Dataset: {Prompt, Desired Response}

Preference Dataset: {Prompt, Winning Response, Losing Response, Choice}

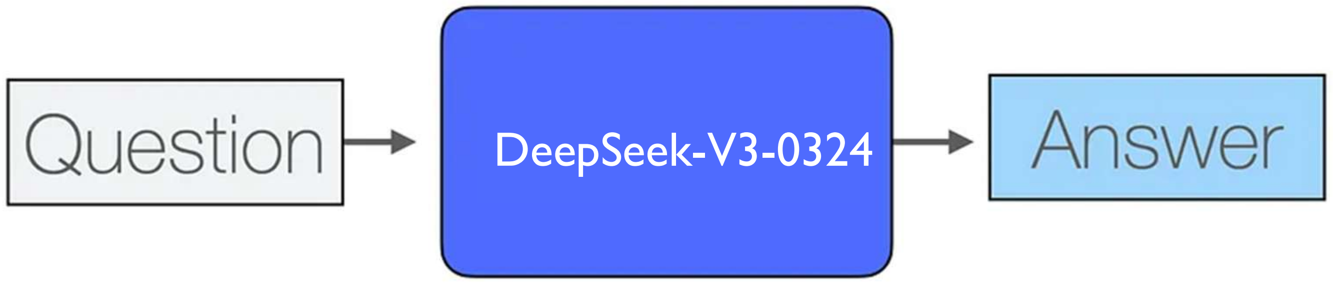
What is the capital of France?
What is France's largest city?
What is France's population?

What is the capital of France?
The Capital of France is Paris.

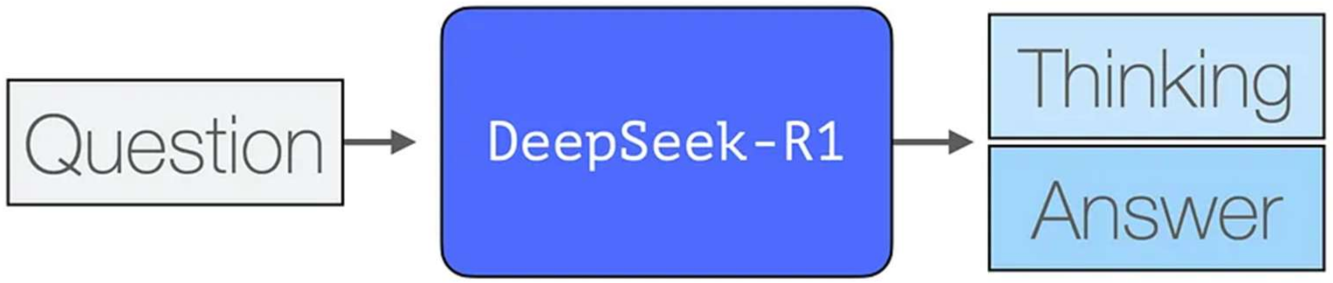
Source: Minha Hwang; [ChatGPT Prompt Engineering for Developers - DeepLearning.AI]

REASONING MODEL (TEST-TIME COMPUTE)

Normal LLM:



Reasoning LLM:



https://newsletter.languagemodels.co/p/the-illustrated-deepseek-r1
(together.ai)

PRE-TRAINING EVAL

PRE-TRAINED EVAL - PERPLEXITY: PREDICTIVE ACCURACY (1/2)

Pre-Training: How well a language model predicts a sequence of words.

- **Low perplexity: better predictive performance, more capabilities**
- **Not a good measure** to evaluate model that have been **post-trained**
- **Perplexity of 3** – Interpretation: This model has a **1 in 3 chance of predicting next token correctly**.
- More structured data: more predictable, lower perplexity
- Bigger vocabulary: higher perplexity
- Longer context window: lower perplexity
- Exponentiation of the average negative log-likelihood of the predictive probability of each word in a test dataset

$$\text{Perplexity}(PP) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_{<i})\right)$$

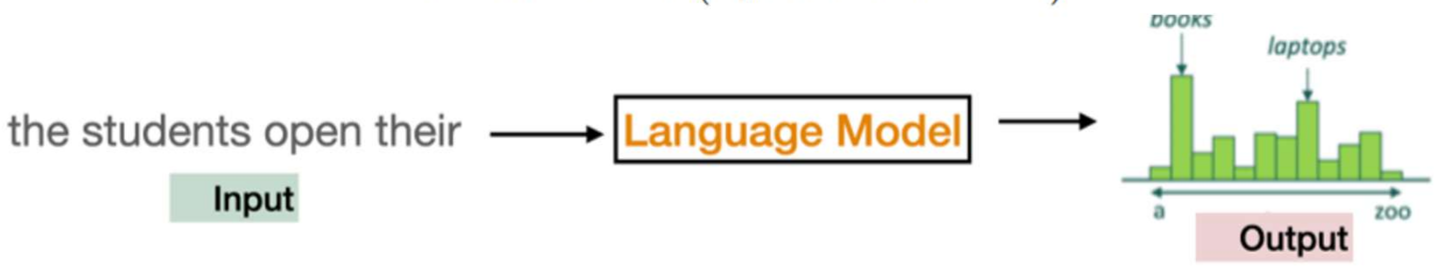


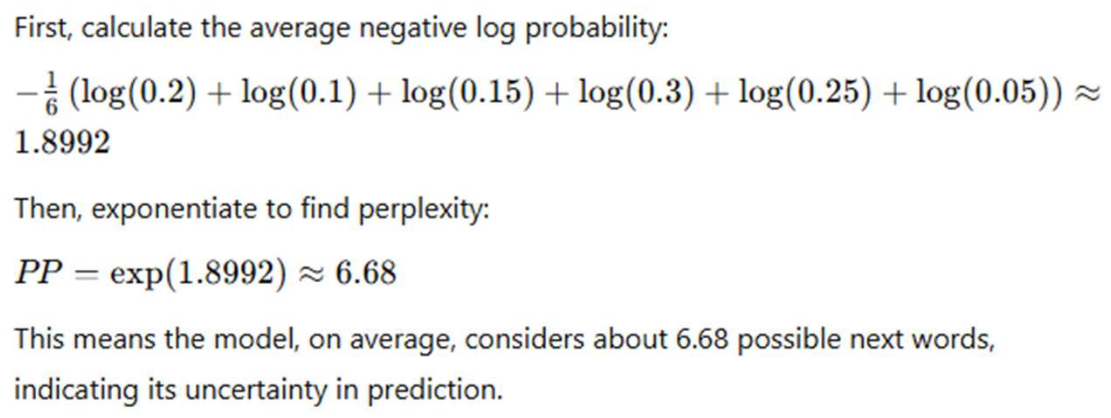
Image source: https//web.Stanford.edu/class/cs224n, AI Engineering by Chip Huyen

PRE-TRAINED EVAL - PERPLEXITY: PREDICTIVE ACCURACY (2/2)

Example Calculation: “The cat sat on the mat”

- P(“The”) = 0.2
- P(“cat”|”The”) = 0.1
- P(“sat”|”The cat”) = 0.15
- P(“on”|”The cat sat”) = 0.3
- P(“the”|”The cat sat on”) = 0.25
- P(“mat”|”The cat sat on the”) = 0.05

$$\text{Perplexity}(PP) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i \mid w_{<i})\right)$$



First, calculate the average negative log probability:

$$-\frac{1}{6}\left(\log(0.2)+\log(0.1)+\log(0.15)+\log(0.3)+\log(0.25)+\log(0.05)\right)\approx 1.8992$$

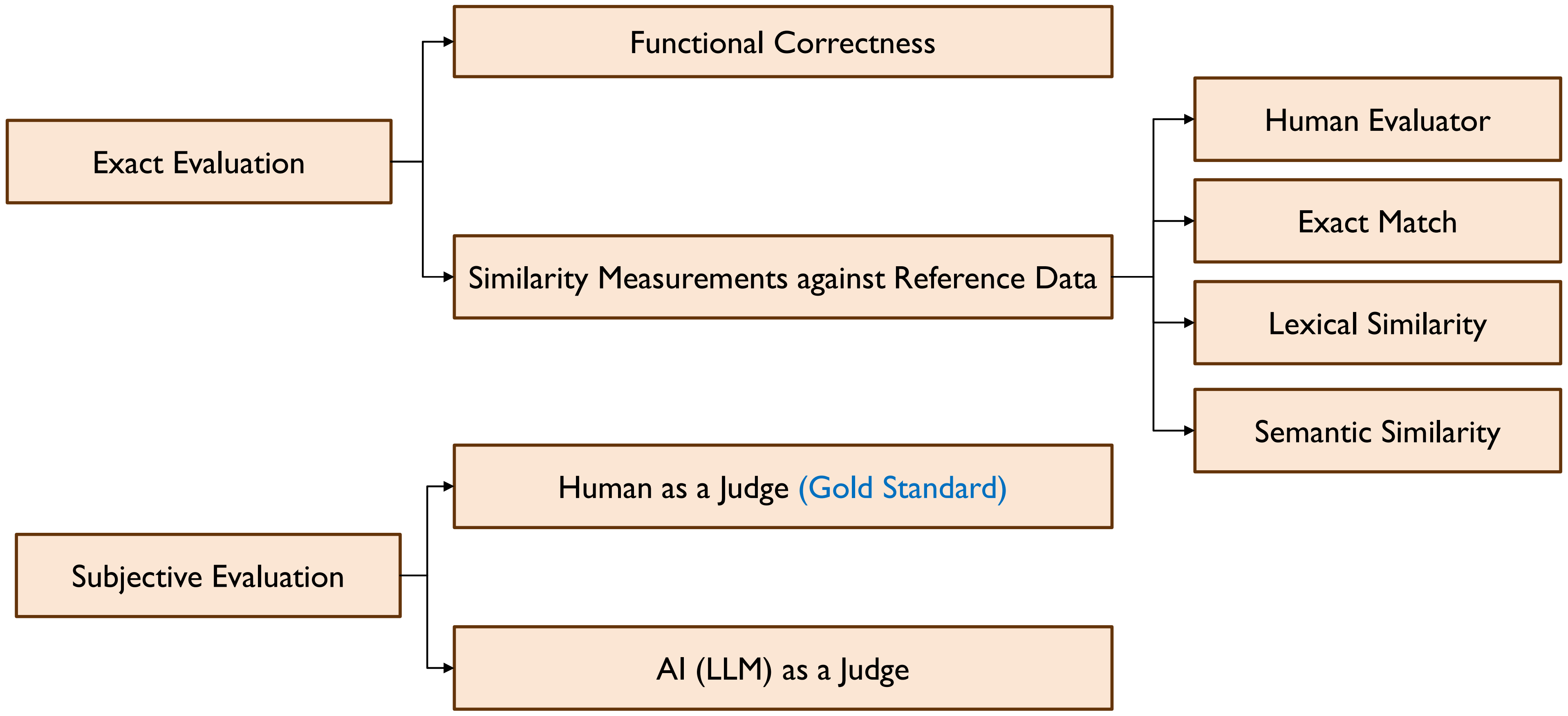
Then, exponentiate to find perplexity:

$$PP = \exp(1.8992) \approx 6.68$$

This means the model, on average, considers about 6.68 possible next words, indicating its uncertainty in prediction.

POST-TRAINING EVAL

POST-TRAINED MODEL EVALUATION: OVERVIEWS



Source: AI Engineering by Chip Huyen, Minha's Synthesis

POST-TRAINED MODEL EVALUATION (1/3): EXACT EVALUATION

Functional Correctness

What it is

- Measures whether your application does what it is intended to do
- Not always straightforward to measure
- Can be automated for certain tasks: code generation, math

Examples

- **Code generation capabilities**: Automated functional correctness measurements (e.g., unit tests) – OpenAI's HumanEval, Google's MBPP, Spider (text-to-SQL)
- **Mathematical problem-solving**, **Fact-based Q&A**

How to measure

- For each coding problem, a number of code samples (i.e., k) are generated
- **pass@k:** A fraction of solved problems out of all problems

Source: AI Engineering by Chip Huyen

Similarity Measurements against Reference Data

What it is

- Evaluate AI's outputs against reference data – ground truths (reference-based)
- Reference data: (input, reference responses) - multiple reference responses possible
- Generated responses that are more similar to the reference responses are considered better
- Bottlenecked by how much and how fast reference data can be generated (human or AI)

Examples

- **BLEU, ROUGE, METEOR++, TER, CIDEr**
- Common for **translation tasks**

How to measure

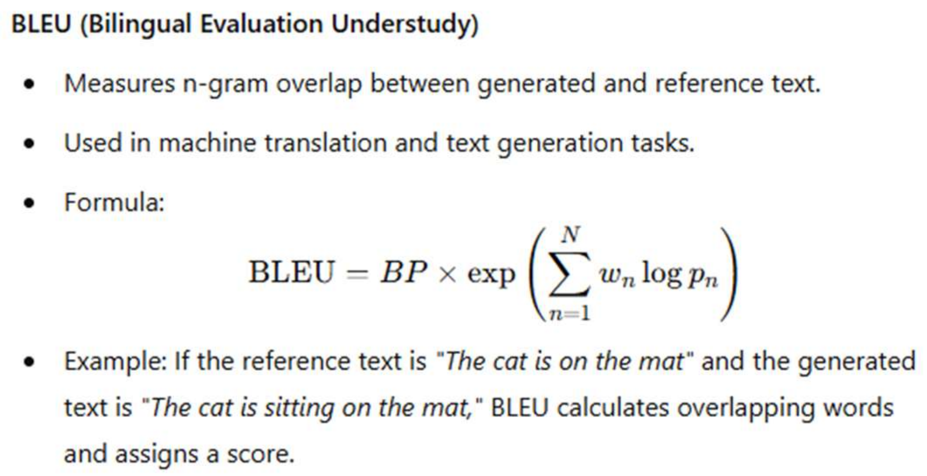
- 4 ways to measure similarity: **human evaluator**, **exact match**, **lexical similarity** (fuzzy matching, edit-distance, counting how many tokens overlap), **semantic similarity** (embedding-based)

POST-TRAINED MODEL EVALUATION (2/3): EXACT EVALUATION

Functional Correctness

- **Pass@k:** or code evaluation (k: number of trial)
- 10 problems and a model solves 5 with k=3: pass@3 score of 50%
- The more code samples a model generates, the more chance the model has at solving each problem, hence the greater the final score
- In expectation, pass@1 score should be lower than pass@10
- Unit tests: run automated tests against generated outputs to check correctness
- Game bots: what score it gets in playing "Tetris"
- Logical validation: Ensure responses follow logical consistency (e.g., in reasoning tasks)

Similarity Measurements against Reference Data



BLEU (Bilingual Evaluation Understudy)

- Measures n-gram overlap between generated and reference text.
- Used in machine translation and text generation tasks.
- Formula:

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

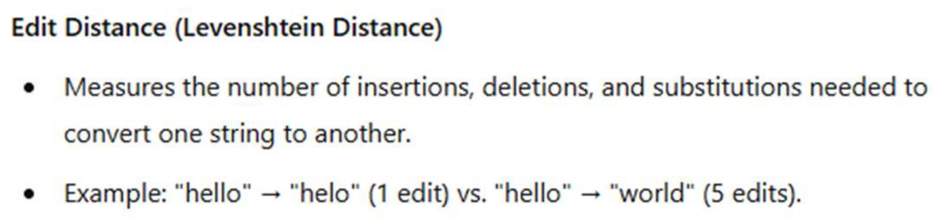
- Example: If the reference text is "*The cat is on the mat*" and the generated text is "*The cat is sitting on the mat,*" BLEU calculates overlapping words and assigns a score.

BP: Brevity Penalty

Source: AI Engineering by Chip Huyen

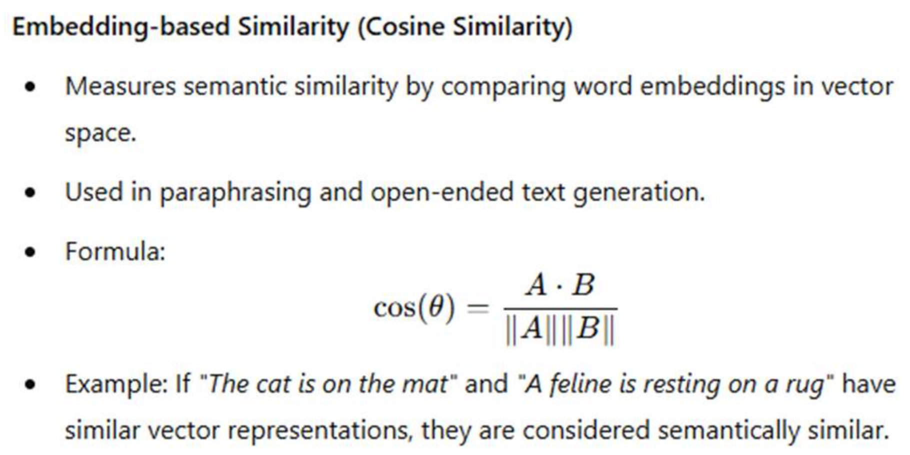
POST-TRAINED MODEL EVALUATION (3/3): EXACT EVALUATION

Similarity Measurements against Reference Data



Edit Distance (Levenshtein Distance)

- Measures the number of insertions, deletions, and substitutions needed to convert one string to another.
- Example: "hello" → "helo" (1 edit) vs. "hello" → "world" (5 edits).



Embedding-based Similarity (Cosine Similarity)

- Measures semantic similarity by comparing word embeddings in vector space.
- Used in paraphrasing and open-ended text generation.
- Formula:

$$\cos(\theta) = \frac{A \cdot B}{||A|| \, ||B||}$$

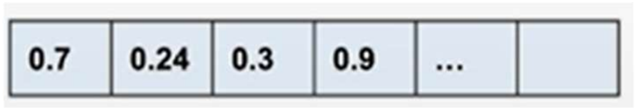
- Example: If "*The cat is on the mat*" and "*A feline is resting on a rug*" have similar vector representations, they are considered semantically similar.

Source: AI Engineering by Chip Huyen

SENTENCE EMBEDDING CAPTURES SEMANTIC MEANING AND CONTEXT-AWARE

Sentence Embeddings: Dense vector representations of sentence that capture their semantic meaning. Context-aware

Generation "I have a dream that one day this nation will rise up and live out the true meaning of its creed: We hold these truths to be self-evident, that all men are created equal."



Sentence

Embedding Vector

Source: Deeplearning.ai – Embedding Models from Architecture to Implementation
https://learn.deeplearning.ai/courses/embedding-models-from-architecture-to-implementation

POST-TRAINED MODEL EVALUTION: SUBJECT EVALUATION (HUMAN PREFERENCE)

Example Criteria

Satisfaction

- **Satisfaction**: Which response makes the user satisfied with the results?
  - **Task Completion (+):** Was the user able to complete his/her tasks?
  - **Efforts (-):** How much efforts did the user put?
- **Helpfulness**: Which response answers the query and follow the specified instructions better?

Factuality (Against Hallucination)

- **Factuality**: Which response provides more accurate answers without hallucinations, even for questions that require very precise or niche knowledge?

Freshness (Against Stale Knowledge)

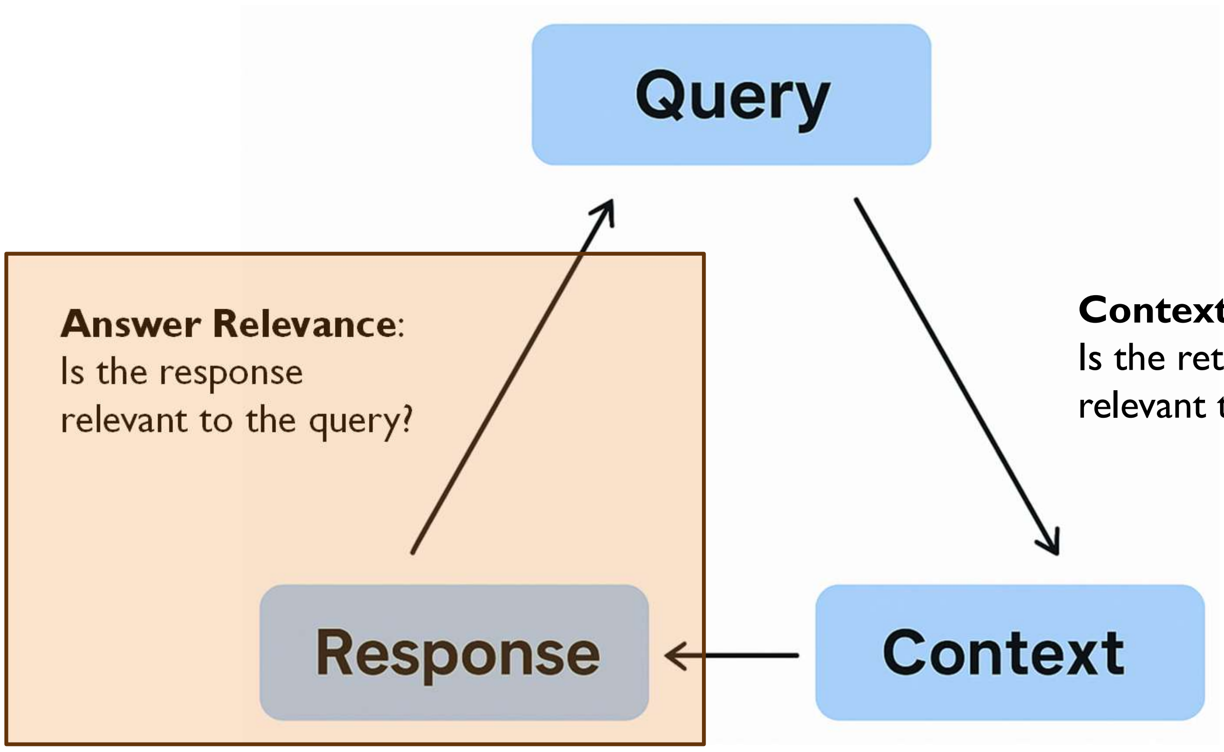
- **Freshness (inspired by FreshLLMs):** Which response contains more up-to-date information? A model excels in this criterion if it is able to answer queries with "fresh" information.

Source: Introducing PPLX Online LLMs
Perplexity enhances LLMs with holistic quality evaluation

Other Criteria: Relevance, Coherence, Fluency, Conciseness, Harmfulness, Maliciousness, Controversiality, Misogyny, Insensitivity, Criminality

RAG EVAL

THE RAG TRIAD: EVALUATING RAG SYSTEM



Answer Relevance: Is the response relevant to the query?

Context Relevance: Is the retrieved context relevant to the query?

Groundedness: Is the response supported by the context?

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: ANSWER RELEVANCE (1/2)

Generation component: synthesizes responses based on the retrieved data

Answer Relevance

- Evaluates the relevance of the generated answer to the user's (original) query, penalizing answers that are incomplete or contain redundant information.
  - How relevant is the generated answer to the question?
  - The metric does not consider factual accuracy but rather focuses on the directness and appropriateness of the response in addressing the question.

- Evaluation process: Similarity-based - Quantify the relevance using BERT-style models, sentence embedding models, or LLMs. The metric is computed in a reference-free manner, using the mean cosine similarity between multiple questions generated from the answer and the original question.
  - 1. Prompt the LLM to generate multiple appropriate questions based on the generated answer.
  - 2. Calculate the cosine similarity between the embedding of each generated question and the original question
  - 3. Compute the mean of these cosine similarity scores to obtain the final answer relevance score.

$$Answer\ Relevance = \frac{\sum cosine\_similarity(embedding\ of\ original\ question, embedding\ of\ question\ generated\ from\ the\ answer)}{Total\ number\ of\ generated\ questions}$$

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: ANSWER RELEVANCE (2/2)

Example

- Question: **“Where is France and what is its capital?”**
- Answer:
  - High relevance “France is in western Europe, and Paris is its capital.” (directly and completely address both parts of the original question.)
  - Low relevance: “France is in western Europe.” (only partially addresses the original query)

Answer relevance metric ensures that the answers provided by RAG systems are not only accurate but also complete and directly address the user’s query without including unnecessary or redundant information

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: FAITHFULNESS (GROUNDEDNESS) (1/2)

Generation component: synthesizes responses based on the retrieved data

Faithfulness (Groundedness)

- Assess the factual alignment and semantic similarity between the model's response and the retrieved documents, ensuring that the generated answer is contextually appropriate and factually grounded in the retrieved information.
  - How factually accurate is the generated answer?
  - Can all the claims made in the model's answer be directly inferred from the given context?

- Evaluation process: hybrid – automated system (CoT prompting; semantic matching) + human judgement
  - 1. Identify a set of claims made in the generated answer
  - 2. Cross-check each claim against the given context to determine its factual consistency and whether it can be directly inferred from the retrieved information
  - 3. Calculate the Groundedness score using the following formula (0 – 1)

$$Groundedness = \frac{Number\ of\ claims\ inferable\ from\ the\ given\ context}{Total\ number\ of\ claims\ in\ the\ generated\ answer}$$

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: FAITHFULNESS (GROUNDEDNESS) (2/2)

Example

- Question: **“Where and when was Einstein born?”**
- Context: **“Albert Einstein (born March 14, 1879) was a German-born theoretical physicist.”**
- Answer:
  - High groundedness: “Einstein was born in Germany on March 14th, 1879.” (aligns with the factual information provided in the context)
  - Low groundedness: “Einstein was born in Germany on March 20th, 1879.” (contradicts the date of birth mentioned in the context)

The Groundedness metric is crucial in ensuring the reliability and trustworthiness of responses generated by RAG systems, as it directly relates to the accuracy and factual consistency of the information provided in response to a user's query.

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: CONTEXT RELEVANCE (1/2)

Retrieval component: responsible for fetching relevant information from a (vector) database or corpus in response to a user's query

Context Relevance

- Measures the alignment of the retrieved information with the user's query, ensuring that only essential information is included to address the query effectively.
  - Is the retrieved context pertinent and appropriate for effectively addressing the given query?
  - Is only essential and directly relevant information included?

- Evaluation process: a two-step procedure
  - 1. Sentence-level Relevance Scoring: Each sentence in the retrieved context is assigned a relevance score based on its semantic similarity to the query, using measures like cosine similarity between embeddings.
  - 2. Overall Context Relevance Quantification: The final context relevance score is calculated as the ratio of relevant sentences to the total number of sentences in the retrieved context (0 – 1)

$$Context\ Relevance = \frac{Number\ of\ relevant\ sentences\ within\ the\ retrieved\ context}{Total\ number\ of\ sentences\ in\ retrieved\ context}$$

Source: (1) Evaluating RAG Systems: A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

EVALUATING RAG SYSTEM: CONTEXT RELEVANCE(2/2)

Example

- Question: **"What is the capital of France?"**

- Context: **"Albert Einstein (born March 14, 1879) was a German-born theoretical physicist."**
  - High context relevance:"*France, in Western Europe, encompasses medieval cities, alpine villages and Mediterranean beaches.* ***Paris, its capital****, is famed for its fashion houses, classical art museums including the Louvre and monuments like the Eiffel Tower."* (includes information directly mentioning Paris as the capital)
  - Low context relevance:"*The country is also renowned for its wines and sophisticated cuisine. Lascaux's ancient cave drawings, Lyon's Roman theater and the vast Palace of Versailles attest to its rich history."* (includes additional, tangential information)

By evaluating Context Relevance, the RAG system can ensure that the retrieved information is concise, focused, and directly addresses the user's query, enhancing the efficiency and accuracy of the generated response.

Source: (1) Evaluating RAG Systems:A Comprehensive Approach to Assessing Retrieval and Generation Performance | LinkedIn

LLM PREFERENCE MEASUREMENT

- Insights from Measurement Theory
- Offline Metric: Measurement Scale Options
- Pros and Cons: MaxDiff as Next Generation
- MaxDiff Example: Answer Card Optimization

MEASUREMENT SCALES

(1) Nominal (Categorical) Scale

- Classifies data into **distinct groups without any inherent order**.
- Example: Gender, Eye Color.
- Statistical Operations: Counting, **mode** calculation.

(2) Ordinal Scale

- Data is **ranked in a meaningful order**, but **intervals between values are not uniform**.
- Example: Education Levels, Customer Satisfaction, Rank-order
- Statistical Operations: **Median,** mode, **non-parametric tests**.

(2) Interval Scale

- Numeric scale with **equal intervals** but **no true zero point**.
- Example: Temperature in Celsius/Fahrenheit.
- Statistical Operations: **Mean**, median, mode, **addition/subtraction**, parametric tests.

(3) Ratio Scale

- Numeric scale with **equal intervals** and an **absolute zero**.
- Example: Weight, Height, Age, Income.
- Operations: All arithmetic operations, including **multiplication and division**.

OFFLINE METRIC - LLM MEASUREMENT SCALE OPTIONS

1. Standard Rating (Likert Scales): Widely Used Before / Less Used No

- Simple but suffers from bias and lack of differentiation.
- Example: Rate each feature from 1-5.

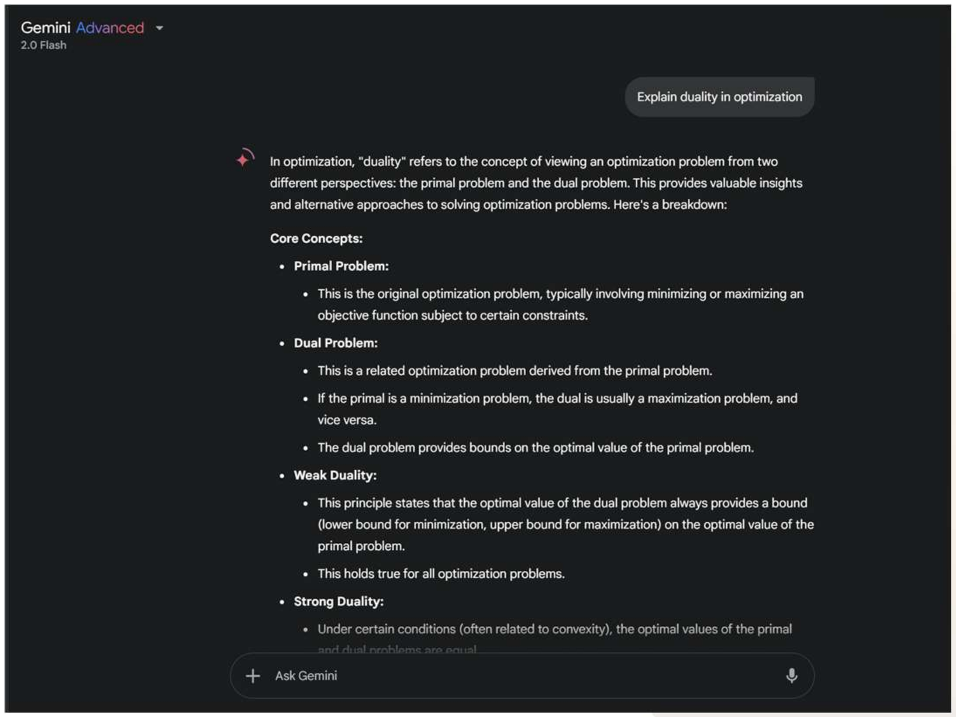
2. Pairwise Comparison: Industry Standard (From 2022) (OpenAI, Anthropic)

- Precise but impractical for large sets.
- Example: **Do you prefer A or B?**

3. MaxDiff (Maximum Difference / Best Worst Scaling): Not Explored Yet for LLM Application

- Reduces bias, handles large lists efficiently, and delivers clear, actionable insights.
- Example: Among A, B, C, D – pick the most preferred & least preferred.

STANDARD RATING (5-POINT LIKERT SCALE) - STATED


Gemini Advanced
2.0 Flash

Explain duality in optimization

In optimization, "duality" refers to the concept of viewing an optimization problem from two different perspectives: the primal problem and the dual problem. This provides valuable insights and alternative approaches to solving optimization problems. Here's a breakdown:

Core Concepts:

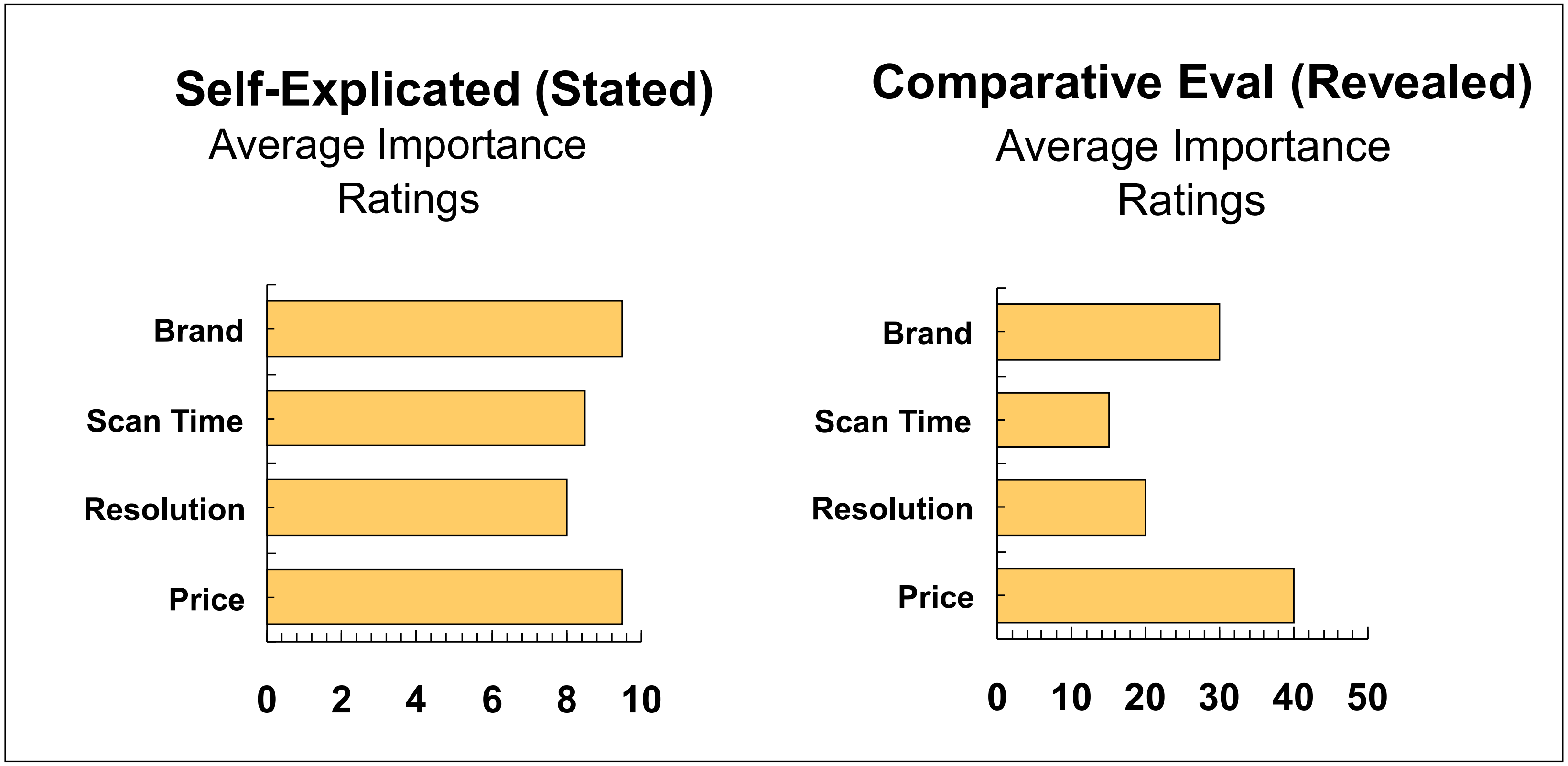
- **Primal Problem:**
  - This is the original optimization problem, typically involving minimizing or maximizing an objective function subject to certain constraints.
- **Dual Problem:**
  - This is a related optimization problem derived from the primal problem.
  - If the primal is a minimization problem, the dual is usually a maximization problem, and vice versa.
  - The dual problem provides bounds on the optimal value of the primal problem.
- **Weak Duality:**
  - This principle states that the optimal value of the dual problem always provides a bound (lower bound for minimization, upper bound for maximization) on the optimal value of the primal problem.
  - This holds true for all optimization problems.
- **Strong Duality:**
  - Under certain conditions (often related to convexity), the optimal values of the primal and dual problems are equal.

Ask Gemini

How satisfied was the user?

○ Very Satisfied ○ Satisfied ○ Neutral [✓ Dissatisfied] ○ Very dissatisfied

MORE DISCRIMINATION FROM TRADE-OFF QUESTIONS



User Preference

EXAMPLE: NEW JOB CHOICE

	Location	Salary
Option 1		$70K
Option 2		$100K

User Preference

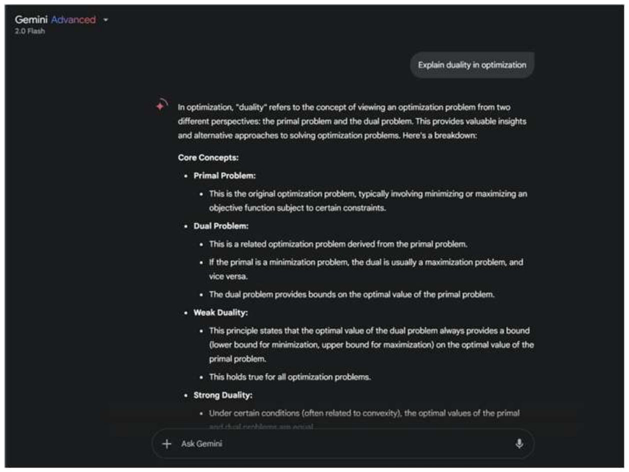
COMPARATIVE EVAL REVEALS WHAT PEOPLE REALLY WANTS

Job Attribute	What MBA's Said... (Stated Rating)	Rank Based on Comparative Eval
Salary	6th	1st
Region of US	2nd	2nd
Job Location	5th	3rd
People/Culture	1st	4th
Functional Area	3rd	5th
Firm Growth	7th	6th
Business Travel	8th	7th
Opp. to Advance	4th	8th

User Preference

PAIR-WISE COMPARISON: GEMINI VS. CHATGPT 4O

Version A: Gemini 2.0



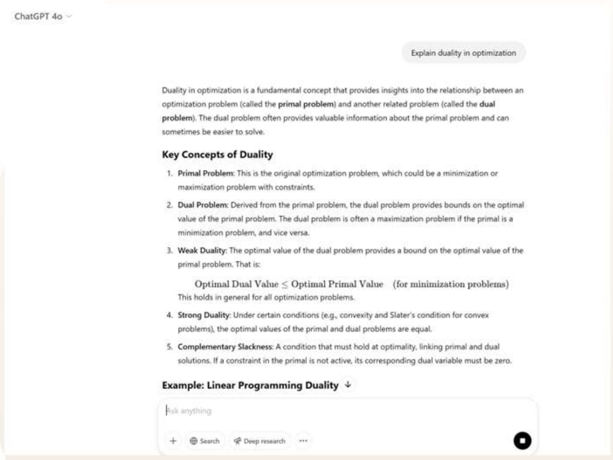
Explain duality in optimization

In optimization, "duality" refers to the concept of viewing an optimization problem from two different perspectives: the primal problem and the dual problem. This provides valuable insights and alternative approaches to solving optimization problems. Here's a breakdown:

Core Concepts:

- **Primal Problem:**
  - This is the original optimization problem, typically involving minimizing or maximizing an objective function subject to certain constraints.
- **Dual Problem:**
  - This is a related optimization problem derived from the primal problem.
  - If the primal is a minimization problem, the dual is usually a maximization problem, and vice versa.
  - The dual problem provides bounds on the optimal value of the primal problem.
- **Weak Duality:**
  - This principle states that the optimal value of the dual problem always provides a bound (lower bound for minimization, upper bound for maximization) on the optimal value of the primal problem.
  - This holds true for all optimization problems.
- **Strong Duality:**
  - Under certain conditions (often related to convexity), the optimal values of the primal and dual problems are equal.

Version B: GPT-4o



Explain duality in optimization

Duality in optimization is a fundamental concept that provides insights into the relationship between an optimization problem (called the **primal problem**) and another related problem (called the **dual problem**). The dual problem often provides valuable information about the primal problem and can sometimes be easier to solve.

Key Concepts of Duality

1. **Primal Problem**: This is the original optimization problem, which could be a minimization or maximization problem with constraints.
2. **Dual Problem**: Derived from the primal problem, the dual problem provides bounds on the optimal value of the primal problem. The dual problem is often a maximization problem if the primal is a minimization problem, and vice versa.
3. **Weak Duality**: The optimal value of the dual problem provides a bound on the optimal value of the primal problem. That is:
   $$\text{Optimal Dual Value} \leq \text{Optimal Primal Value} \quad \text{(for minimization problems)}$$
   This holds in general for all optimization problems.
4. **Strong Duality**: Under certain conditions (e.g., convexity and Slater's condition for convex problems), the optimal values of the primal and dual problems are equal.
5. **Complementary Slackness**: A condition that must hold at optimality, linking primal and dual solutions. If a constraint in the primal is not active, its corresponding dual variable must be zero.

Example: Linear Programming Duality

Which chatbot response do you prefer?

○	○	○	○	○
A >> B	A > B	A == B	A < B	A << B

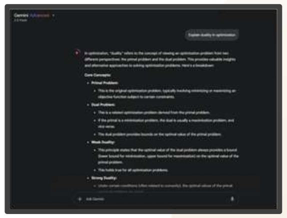
MAXDIFF: GEMINI, CHATGPT 4O, GROK 3

Based on Chatbot responses, please choose your Favorite and Least Favorite

Task 1/7

Favorite **Worst**

○ ○



Gemini Advanced
2.0 Flash

Explain duality in optimization

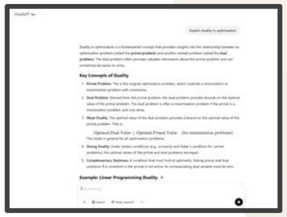
In optimization, "duality" refers to the concept of viewing an optimization problem from two different perspectives: the primal problem and the dual problem. This provides valuable insights and alternative approaches to solving optimization problems. Here's a breakdown:

Core Concepts:

- **Primal Problem:**
  - This is the original optimization problem, typically involving minimizing or maximizing an objective function subject to certain constraints.
- **Dual Problem:**
  - This is a related optimization problem derived from the primal problem.
  - If the primal is a minimization problem, the dual is usually a maximization problem, and vice versa.
  - The dual problem provides bounds on the optimal value of the primal problem.
- **Weak Duality:**
  - This principle states that the optimal value of the dual problem always provides a bound (lower bound for minimization, upper bound for maximization) on the optimal value of the primal problem.
  - This holds true for all optimization problems.
- **Strong Duality:**
  - Under certain conditions (often related to convexity), the optimal values of the primal

Ask Gemini

○ ○



ChatGPT 4o

Explain duality in optimization

Duality in optimization is a fundamental concept that provides insights into the relationship between an optimization problem (called the **primal problem**) and another related problem (called the **dual problem**). The dual problem often provides valuable information about the primal problem and can sometimes be easier to solve.

Key Concepts of Duality

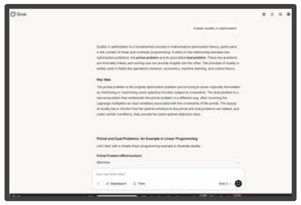
1. **Primal Problem:** This is the original optimization problem, which could be a minimization or maximization problem with constraints.
2. **Dual Problem:** Derived from the primal problem, the dual problem provides bounds on the optimal value of the primal problem. The dual problem is often a maximization problem if the primal is a minimization problem, and vice versa.
3. **Weak Duality:** The optimal value of the dual problem provides a bound on the optimal value of the primal problem. That is:
   $$\text{Optimal Dual Value} \leq \text{Optimal Primal Value} \quad \text{(for minimization problems)}$$
   This holds in general for all optimization problems.
4. **Strong Duality:** Under certain conditions (e.g., convexity and Slater's condition for convex problems), the optimal values of the primal and dual problems are equal.
5. **Complementary Slackness:** A condition that must hold at optimality, linking primal and dual solutions. If a constraint in the primal is not active, its corresponding dual variable must be zero.

Example: Linear Programming Duality

Ask anything

Search Deep research

○ ○



Grok

Explain duality in optimization

Duality in optimization is a fundamental concept in mathematical optimization theory, particularly in the context of linear and nonlinear programming. It refers to the relationship between two optimization problems: the **primal problem** and its associated **dual problem**. These two problems are intricately linked, and solving one can provide insights into the other. The principle of duality is widely used in fields like operations research, economics, machine learning, and control theory.

Key Idea

The primal problem is the original optimization problem you're trying to solve—typically formulated as minimizing or maximizing some objective function subject to constraints. The dual problem is a derived problem that reinterprets the primal problem in a different way, often involving the Lagrange multipliers (or dual variables) associated with the constraints of the primal. The beauty of duality lies in the fact that the optimal solutions to the primal and dual problems are related, and under certain conditions, they provide the same optimal objective value.

Primal and Dual Problems: An Example in Linear Programming

Let's start with a simple linear programming example to illustrate duality.

Primal Problem (Minimization)

Minimize:

How can Grok help?

DeepSearch Think Grok 3

PROS AND CONS

(1) Standard Rating Tasks (Likert Scales)

- Pros: Simple and intuitive for respondents.
- Cons: **Limited discrimination** between items; susceptible to biases.

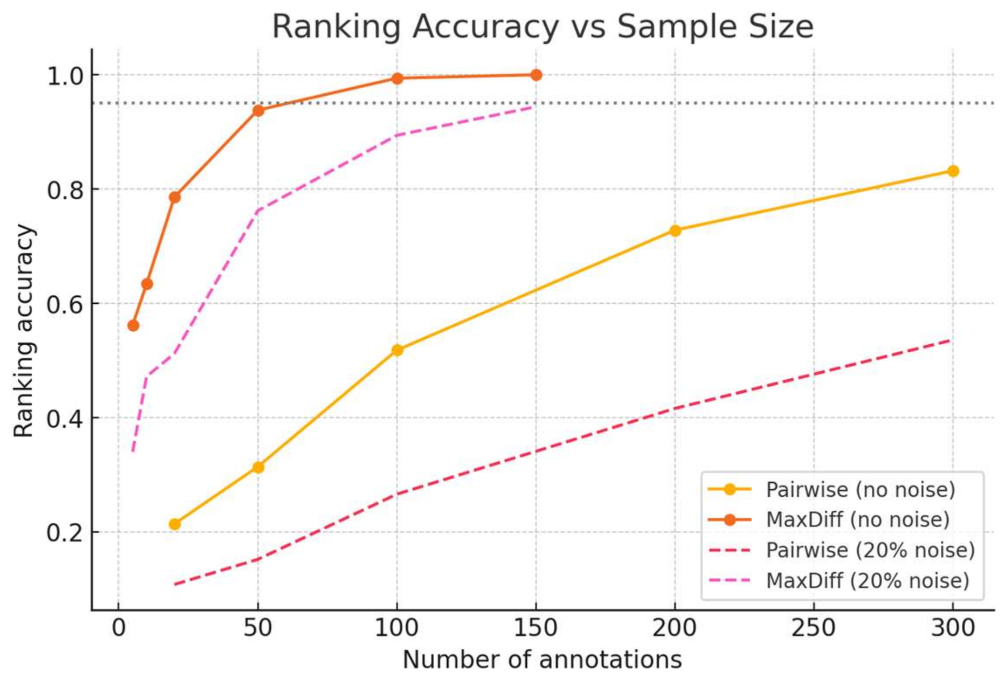
(2) Pairwise Comparison

- Pros: Provides precise preference data.
- Cons**: Impractical with large item sets**; can lead to **respondent fatigue**.

(3) MaxDiff (Maximum Difference Scaling)

- Pros: Enhances discrimination by **forcing trade-offs**; reduces biases; allows more robust optimization beyond measurement
- Cons: **More complex to design** and analyze; may be cognitively demanding.

MAXDIFF IS MORE SAMPLE EFFICIENT AND ROBUST: SIMULATION RESULTS



5 Model Candidates: showing 4 candidates per each task for MaxDiff

MAXDIFF EXAMPLE: ANSWER CARD OPTIMIZATION

(1) Key Attributes to Consider:

- **Presentation Format**: Text, Rich details with picture, Text with box.
- **Font Size**: Small, Medium, Large.
- **Color Scheme**: Light theme, Dark theme, Colorful theme.
- Implementing MaxDiff for Answer Card Optimization:

(2) Define the combinations: options to show

(3) Design the MaxDiff survey: Sawtooth software

(4) Conduct MaxDiff survey question: Sawtooth software

(5) Analyzing the results: Hierarchical Bayes to Retrieve underlying preference

CHATBOT ARENA / MAXDIFF RESOURCE

- Chatbot Arena (LMSYS): Crowd-based pair-wise comparison
  - https://lmarena.ai/
- MaxDiff Exercise YouTube Video: Sawtooth Software
  - https://youtu.be/Sg-cxgCOVf4
- MaxDiff Explainer from Sawtooth Software
  - https://sawtoothsoftware.com/maxdiff
- Interactive Apps and Data Collection and Voting Results
  - Gradio: https://www.deeplearning.ai/short-courses/building-generative-ai-applications-with-gradio/
  - Chatbot Response Collector: Claude.ai: file:///C:/Users/h_min/Downloads/prompt-collector-MVP.html