Evaluating Large Language Models

- Principles, Approaches, and Applications

NeurIPS Tutorial

Bo Li · Irina Sigler · Yuan (Emily) Xue

Dec 2024

Agenda

01 Intro

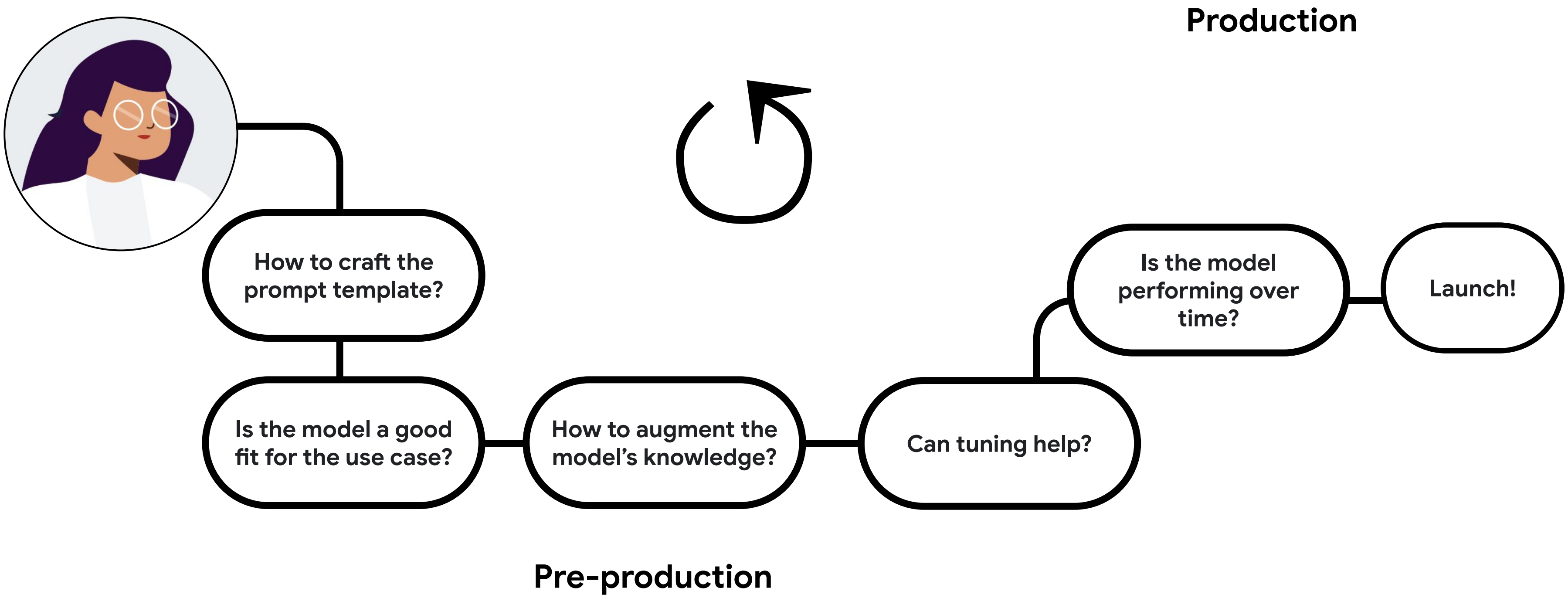
02 Quality evaluation

03 Safety evaluation

04 Wrap up & QA

01

Introduction


Production
How to craft the prompt template?
Is the model a good fit for the use case?
How to augment the model's knowledge?
Can tuning help?
Is the model performing over time?
Launch!
Pre-production

Task-specific evaluation

Use Case

Data representing your application

Context

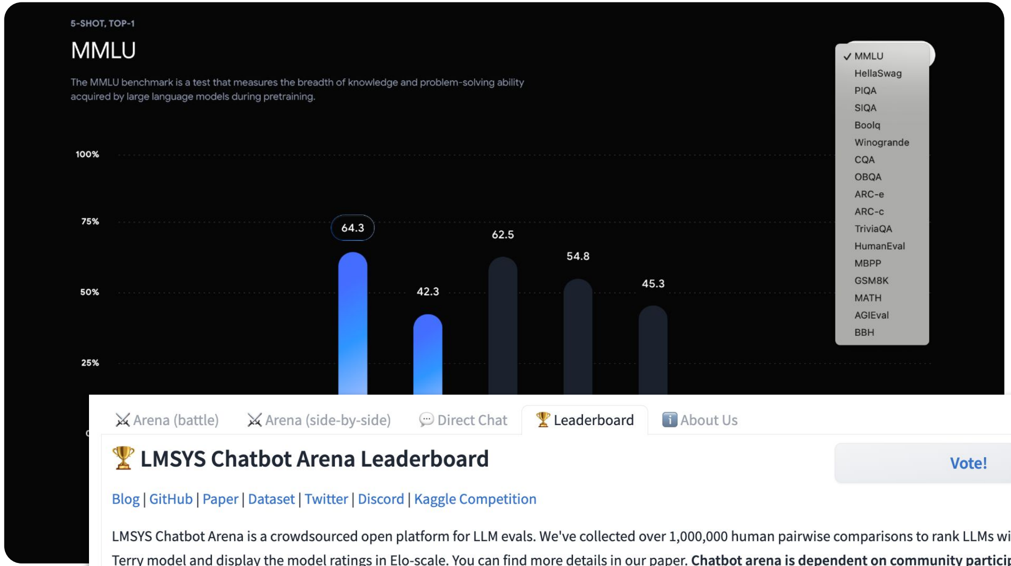
Model is only one of the lego bricks

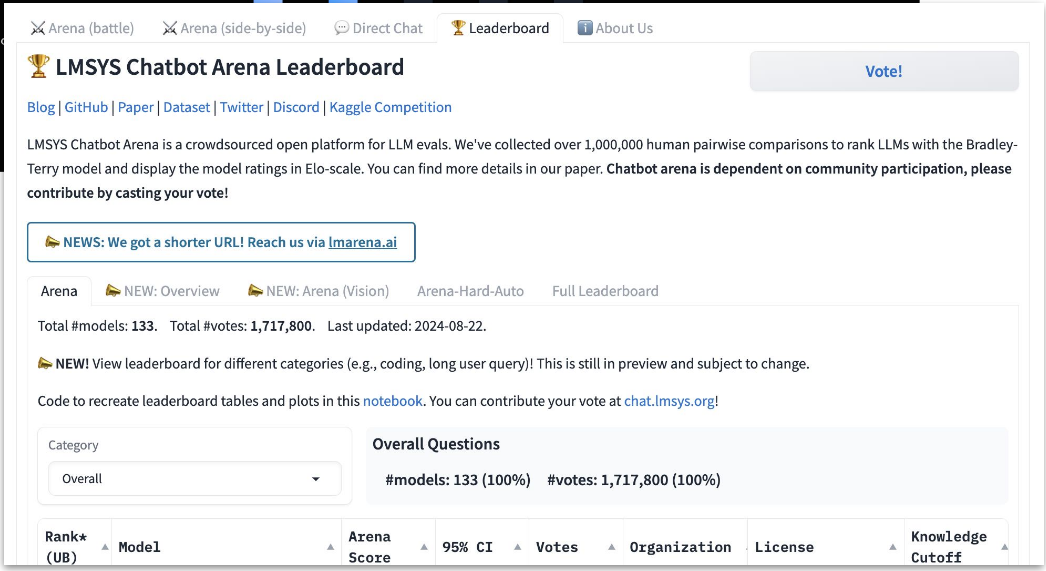
Criteria

Your definition of success

Use Case

Data representing your application





Use Case

Data representing your application

Manual

Synthetic

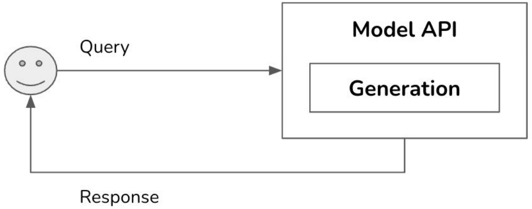
Traffic

Use Case

Data representing your application

Context

Model is only one of the lego bricks



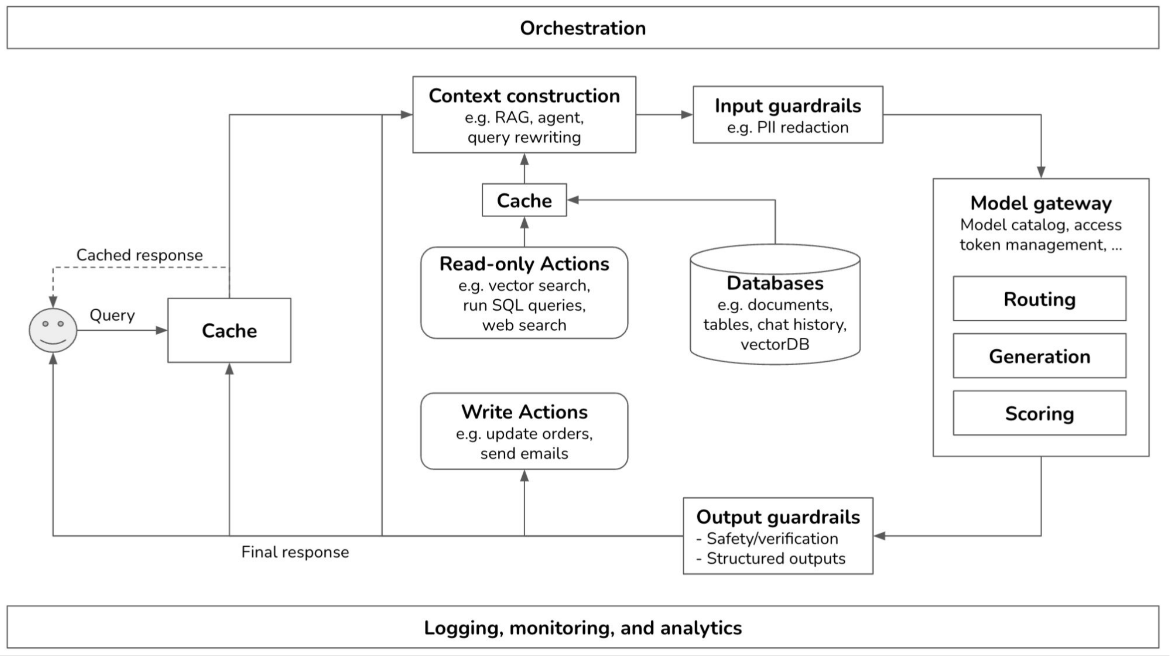
Source: Huyen, 2024

Use Case

Data representing your application

Context

Model is only one of the lego bricks



Source: Huyen, 2024

Use Case

Data representing your application

Manual	Synthetic	Traffic

Context

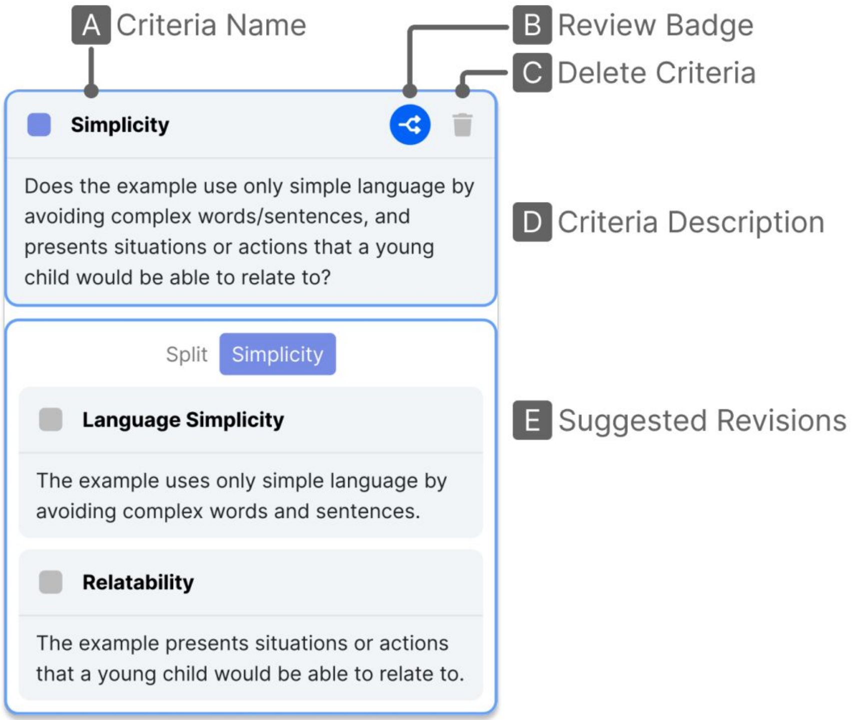
Model is only one of the lego bricks

Final	Intermediate	Trajectory

Use Case
Data representing your application

Context
Model is only one of the lego bricks

Criteria
Your definition of success



See: Kim et al. 2024, for details on specific criteria & Shankar et al. 2024 for iterative criteria refinement

Use Case Data representing your application	Manual	Synthetic	Traffic
Context Model is only one of the lego bricks	Final	Intermediate	Trajectory
Criteria Your definition of success	Similarity	Criteria per task	Rubrics per data point

See: [Wiles et al. 2024 for text to image evaluation with gecko]

Use Case Data representing your application	Manual	Synthetic	Traffic
Context Model is only one of the lego bricks	Final	Intermediate	Trajectory
Criteria Your definition of success	Similarity	Criteria per task	Rubrics per data point

Automatic evaluation is the holy grail, but still a work in progress. Without it, engineers are left with eye-balling results and testing on a limited set of examples, and having a 1+ day delay to know metrics.

Linkedin team, 2024, Musings on building a Generative AI product

The model eval was the key to success in order to put a LLM in production. We couldn't afford a manual check and refinement in a non-static ecosystem.

Stefano Frigerio, Head of Technical Leads, Generali Italia

02

Quality Evaluation

Evaluation – Problem Statement

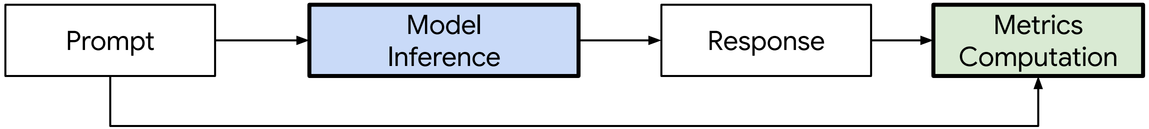
$$F\,(subject,\ criteria) \longrightarrow result$$

F (subject, criteria) ⟶ result

Evaluation – Subject

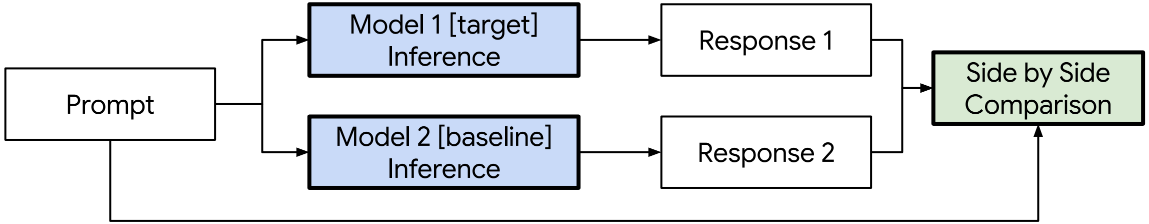
Point-wise:
prompt ⟶ response
Result: absolute measures

Point-wise



Pair-wise:
prompt ⟶ (response 1, response 2)
Result: relative preference

Pair-wise (Side by Side)

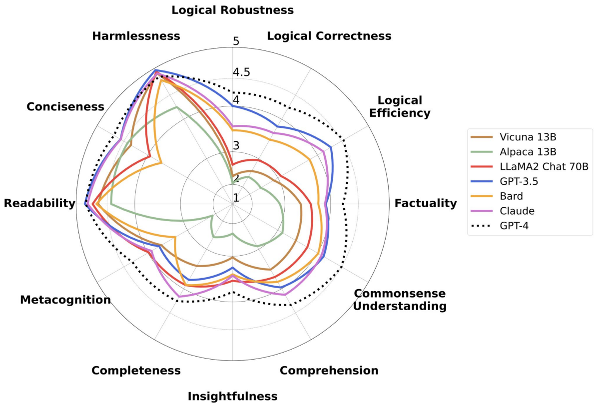


Evaluation – Criteria

Aspect (Dimension):

- General text generation: e.g., fluency, coherence,
- Task related
  - Summary: e.g., Conciseness, Comprehensiveness,
  - Openbook Q/A: Groundedness
  - Code: correctness of execution result
  - Tool use: tool selection accuracy, parameter value correctness
- User specific
  - Entertaining, Engaging, intuitive

Rubrics



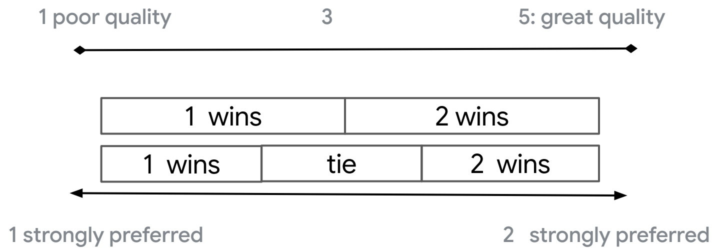
Source: FLASK (Ye 2023)

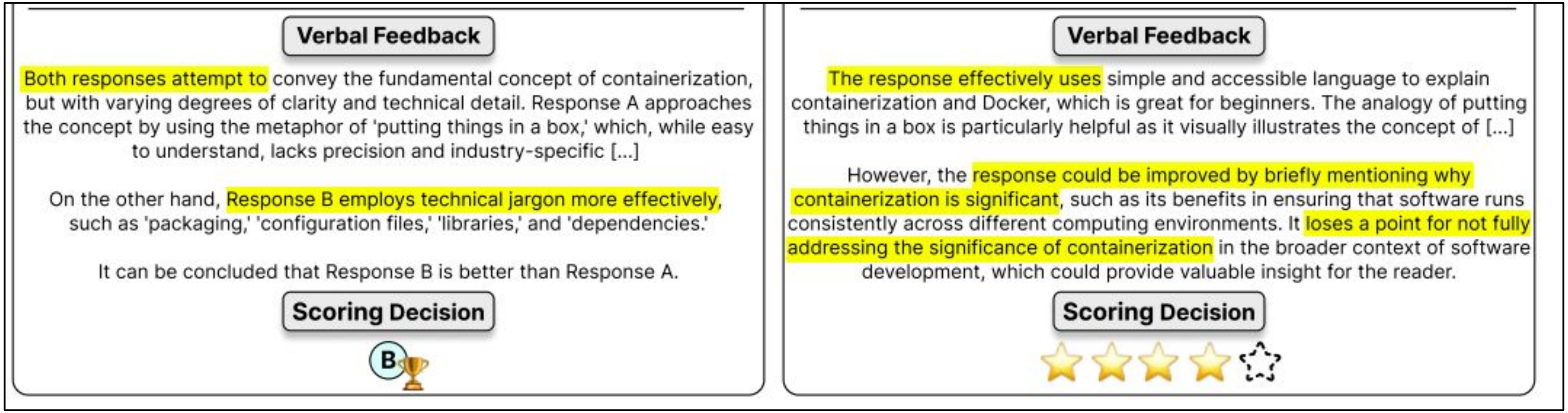
5: (Very good). The summary follows instructions, is grounded, concise, fluent and aligned with reference summary.
4: (Good). The summary follows instructions, is grounded, concise, and fluent but not aligned with reference summary.
3: (Ok). The summary mostly follows instructions, is grounded, but is not concise, not fluent, not aligned with reference summary.
2: (Bad). The summary is grounded, but does not follow the instructions.
1: (Very bad). The summary is not grounded.

F (subject, criteria) ⟶ result

Evaluation – Result

- **Rating**: qualitative measure
    - Point-wise: Absolute measure
    - Pair-wise: Relative preference
- **Rationale**: verbal feedback
    - Explanation to user
    - Captures reasoning thoughts and improves rating quality





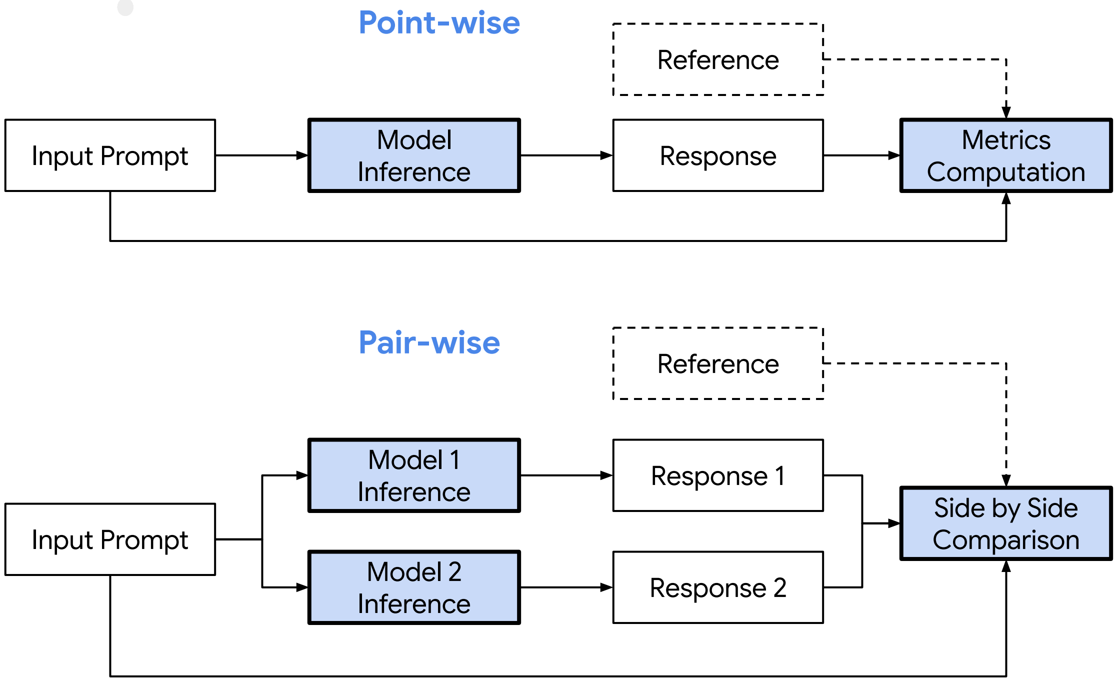
Source: Prometheus (Kim 2024)

Evaluation – Reference

F (subject, criteria, reference) ⟶ result*

- Can be optional
- Evaluation Perspective: Similarity to Reference

- Discriminative task:
  - Ground truth
- Generative task:
  - Representative sample



Google

Evaluation – Method

- Computation
- Human
- LLM (LLM as Judge, as critic, **Autorater**)

Method – Computation (1)

F (subject, criteria, reference) ⟶ result*

Quantify the similarity between response and reference

- Reference Required
- Support point-wise eval
- Only provide score as result
- Does not support fine-grained criteria specificification

*F ((prompt, response), **reference**) ⟶ score*

Metrics	Naturalness ρ	Naturalness τ	Coherence ρ	Coherence τ	Engagingness ρ	Engagingness τ	Groundedness ρ	Groundedness τ	Average ρ	Average τ
ROUGE-L	0.146	0.176	0.203	0.193	0.300	0.295	0.327	0.310	0.244	0.244
BLEU-4	0.175	0.180	0.235	0.131	0.316	0.232	0.310	0.213	0.259	0.189
BERTScore	0.209	0.226	0.233	0.214	0.335	0.317	0.317	0.291	0.274	0.262
G-EVAL-3.5	0.539	0.532	0.544	0.519	0.691	0.660	0.567	0.586	0.585	0.574
G-EVAL-4	0.565	0.549	0.605	**0.594**	0.631	0.627	0.551	0.531	0.588	0.575
ChatGPT(SA)	0.474	0.421	0.527	0.482	0.599	0.549	0.576	0.558	0.544	0.503
ChatGPT(MA)	0.441	0.396	0.500	0.454	0.664	0.607	0.602	0.583	0.552	0.510
GPT-4(SA)	0.532	0.483	0.591	0.535	0.734	0.676	**0.774**	**0.750**	0.658	0.611
GPT-4(MA)	**0.630**	**0.571**	**0.619**	0.561	**0.765**	**0.695**	0.722	0.700	**0.684**	**0.632**

On SummEval Spearman (ρ) and Kendall-Tau (τ)

Source: G-Eval (Liu 2023)

Approaches

- Lexicon similarity: e.g., ROUGE, BLEU
- Embedding similarity: E.g. BERTScore, BARTscore

Limitation

- Sensitive to the choice of reference.
- Lexicon similarity only measures syntactical matches rather than semantics
- Weak correlation with human judgment in complex, open-ended tasks.

Usage

- Scalable evaluation in simple settings
- Break down big eval tasks into smaller pieces (e.g. in Function Calling evaluation, parameter value comparison)
- Low-cost sanity check and monitoring of tuning progress
- Complement other approaches (human, autorater) to provide an objective assessment

F ((prompt, response), reference) -> score

Method – Computation (2)

Example: **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)

- The score ranges from 0 (poor similarity) to 1 (strong similarity)
- A set of metrics:
  - ROUGE-n examines word groups (n-grams).

$$RECALL = \frac{Overlapping\ number\ of\ n\!-\!grams}{Number\ of\ n\!-\!grams\ in\ the\ reference}$$

$$PRECISION = \frac{Overlapping\ number\ of\ n\!-\!grams}{Number\ of\ n\!-\!grams\ in\ the\ candidate}$$

  - ROUGE-L is based on the longest common subsequence (LCS) appear in the same order.
  - ROUGE-Lsum: based on ROUGE-L at the sentence level; aggregates all the results for the final score; suitable for tasks where sentence level extraction is valuable such as extractive summarization tasks.
- Best Practice: Preprocessing to remove any noise or irrelevant information (e.g., punctuation, stop words) that might interfere with the evaluation process.

```
from rouge_score import rouge_scorer
scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL', 'rougeLsum'])

scores = scorer.score('The quick brown fox jumps over the lazy dog',  'The quick brown dog jumps on the log.')
print(scores)

{
'rouge1': Score(precision=0.75, recall=0.67, fmeasure=0.71),
'rouge2': Score(precision=0.29, recall=0.25, fmeasure=0.27),
'rougeL': Score(precision=0.625, recall=0.56, fmeasure=0.59),
'rougeLsum': Score(precision=0.625, recall=0.56, fmeasure=0.59)
}
```

Google

Method – Human

F (subject, criteria, reference) -> result*

*F ((prompt, **response**), criteria) -> score, rational*
*F ((prompt, **response1, response2**), criteria) -> preference, rational*

Goal: Ensure quality and control cost

Phased Approach:

- Start with Samples: train human evaluators and calibrate their judgments using a clear rubric.
- Proceed to Full Scale: expand evaluation to a larger set; allows for iterative refinement of the evaluation process.

Limitations:

- Expensive and time-Consuming
- Human Expertise Matters: The quality of human evaluation depends on the expertise and consistency of the evaluators.
  - Crowdsourcing.
  - Annotator Services: Engage professional annotation services for higher precision.
  - Domain Expertise: For specialized tasks, prioritize evaluators with relevant domain knowledge to ensure meaningful assessments.

Usage:

- Production Release: directly inform decision-making for product readiness, ensuring that quality standards meet production requirements.
- calibrate and optimize Autorater: Use a small number of human labelled data to assess the quality of autorater, iterate its quality as needed, and use autorater for scalable evaluation.

Method – AutoRater

F (subject, criteria, reference) -> result*

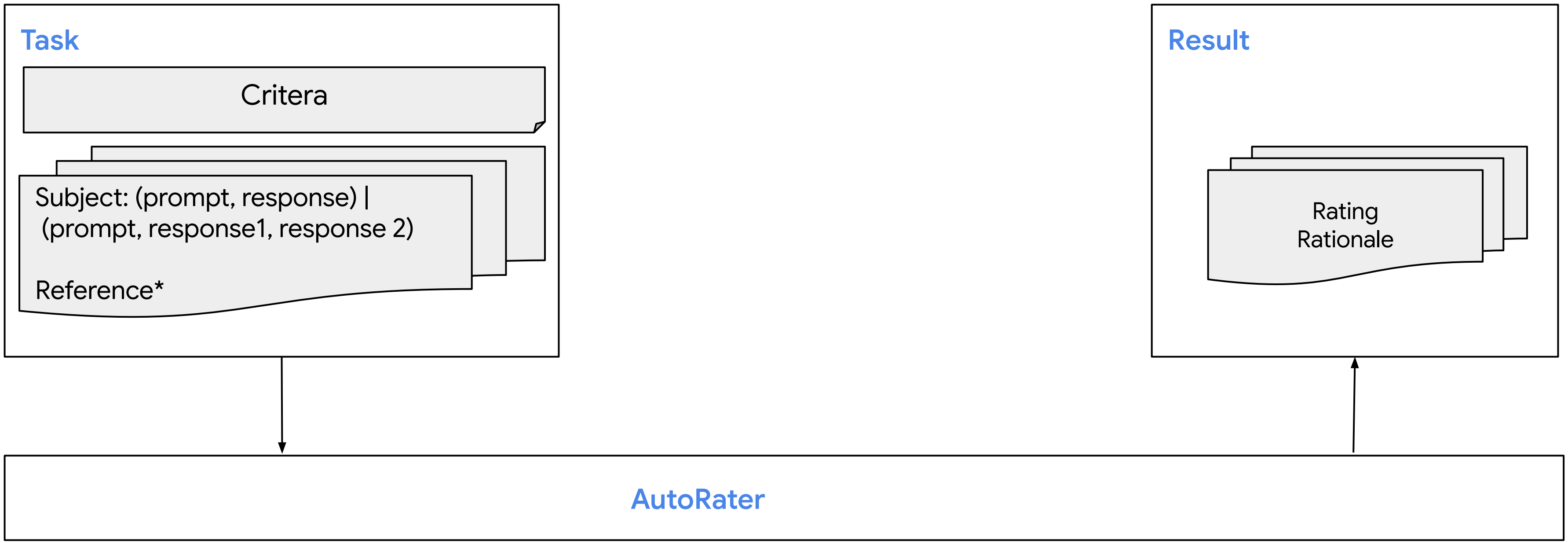
*F ((prompt, **response**), criteria, reference*) -> score, rational*
*F ((prompt, **response1**, **response2**), criteria, reference*) -> preference, rational*

⟶ *Same scope as human evaluation*

- How to use
- How to design
- How to evaluate (meta-evaluation)
- How to align with your needs
- Limitations and migations

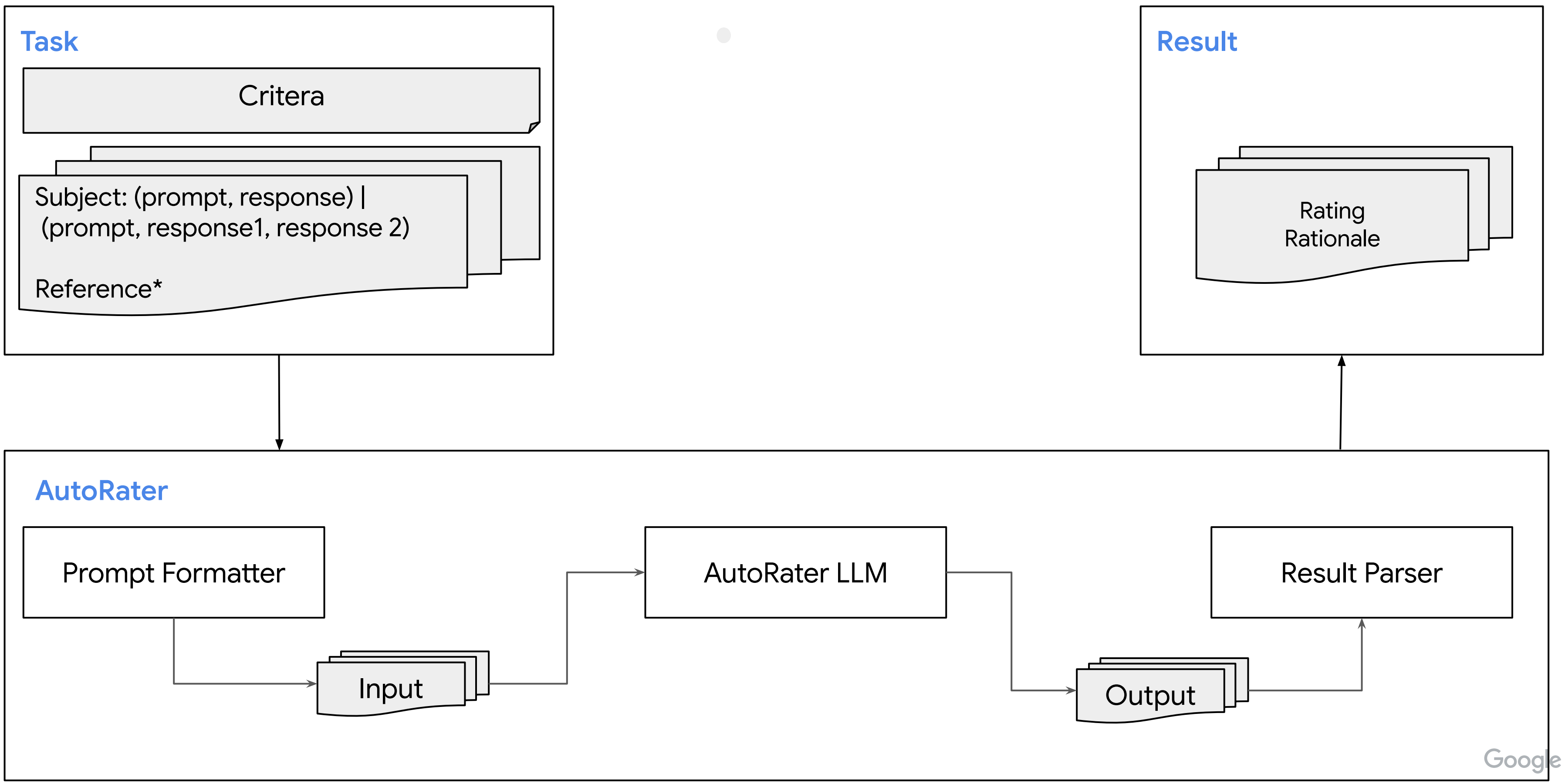
AutoRater – How to Use

*F ((prompt, **response**), criteria, reference*) -> score, rational*
*F ((prompt, **response1**, **response2**), criteria, reference*) -> preference, rational*

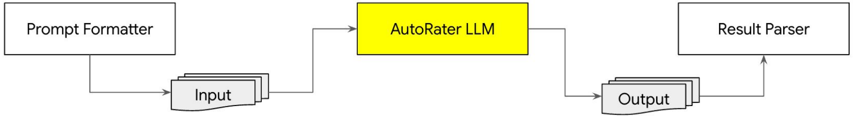


Google

AutoRater – Design Framework



AutoRater – Types of Model

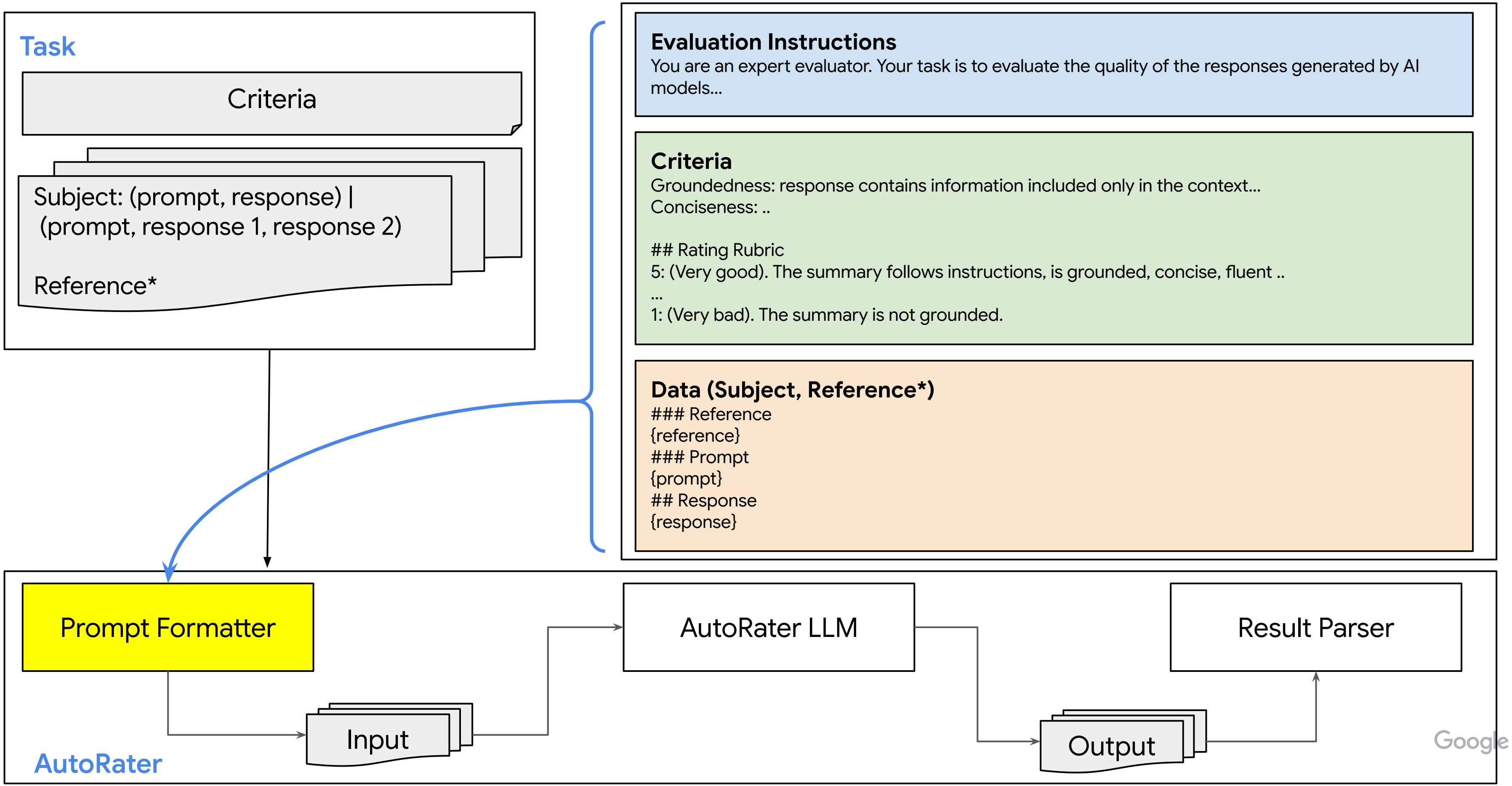


- **Generative Models**
  - Leverage language generation capabilities to deliver both score and detailed rationales (e.g.,CoT explanations).
  - General (foundation model) vs fine-tuned specialized autorater model
  - Flexibility in output formatting: Support both pointwise scoring and pairwise comparisons
  - Need a result parser to get the score from the text output, sometimes this may fail due to malformatting.
  - Can directly prompt foundation model without fine-tuning or be fine-tuned for improved accuracy
- **Discriminative Models** (Reward Models).
  - Trained to predict scalar scores
  - Optimized to deliver precise and consistent evaluations based on specified criteria
  - Support both pointwise scoring and pairwise comparisons
  - No support for rationale and nuanced reasoning
- Implicit Reward Models via DPO, Although less common, generally underperform compared to discriminative and generative models and are not the primary focus here.

	Model	Model Type
1	Skywork/Skywork-Reward-Gemma-2-27B-v0.2	Seq. Classifier
2	nvidia/Llama-3.1-Nemotron-70B-Reward *	Custom Classifier
3	Skywork/Skywork-Reward-Gemma-2-27B ⚠️	Seq. Classifier
4	SF-Foundation/TextEval-Llama3.1-70B *	Generative
5	meta-metrics/MetaMetrics-RM-v1.0	Custom Classifier
6	Skywork/Skywork-Critic-Llama-3.1-70B ⚠️	Generative
7	Skywork/Skywork-Reward-Llama-3.1-8B-v0.2	Seq. Classifier
8	nicolinho/QRM-Llama3.1-8B ⚠️	Seq. Classifier
9	LxzGordon/URM-LLaMa-3.1-8B ⚠️	Seq. Classifier
10	Salesforce/SFR-LLaMa-3.1-70B-Judge-r *	Generative
11	Skywork/Skywork-Reward-Llama-3.1-8B ⚠️	Seq. Classifier
12	general-preference/GPM-Llama-3.1-8B ⚠️	Custom Classifier
13	nvidia/Nemotron-4-340B-Reward *	Custom Classifier
14	Ray2333/GRM-Llama3-8B-rewardmodel-ft ⚠️	Seq. Classifier
15	SF-Foundation/TextEval-OffsetBias-12B *	Generative

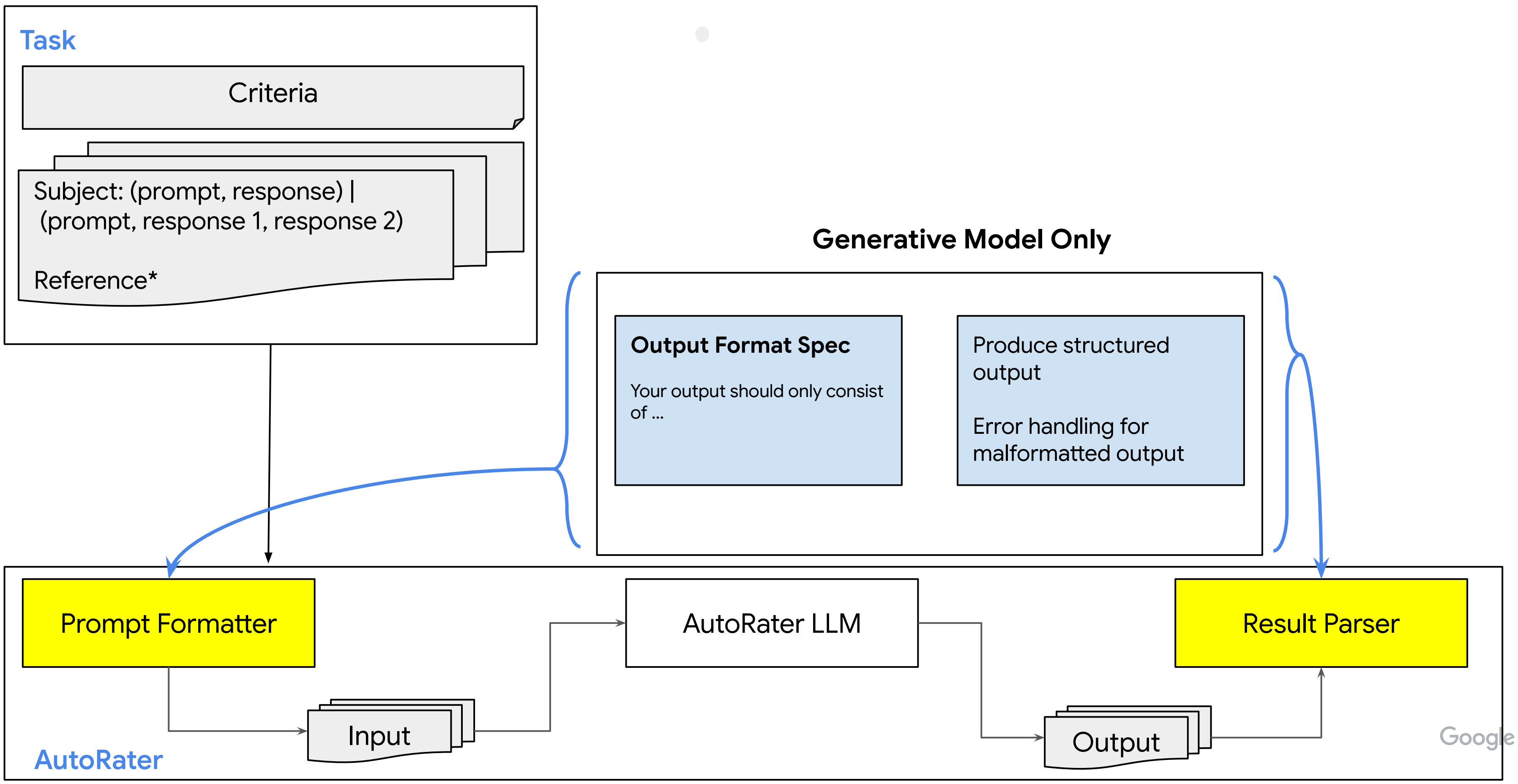
Source: RewardBench

AutoRater – Prompt Formatter



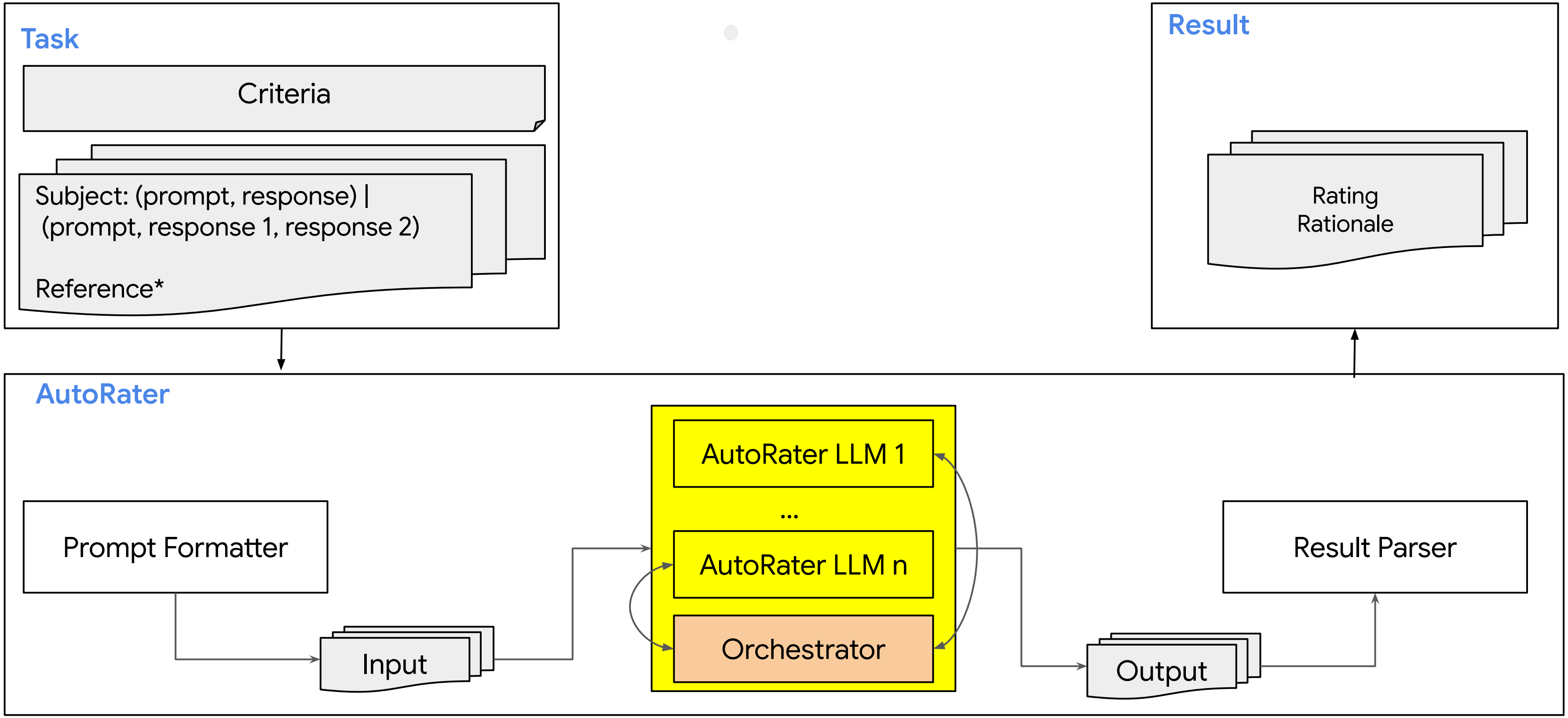
Google

AutoRater – Prompt Formatter



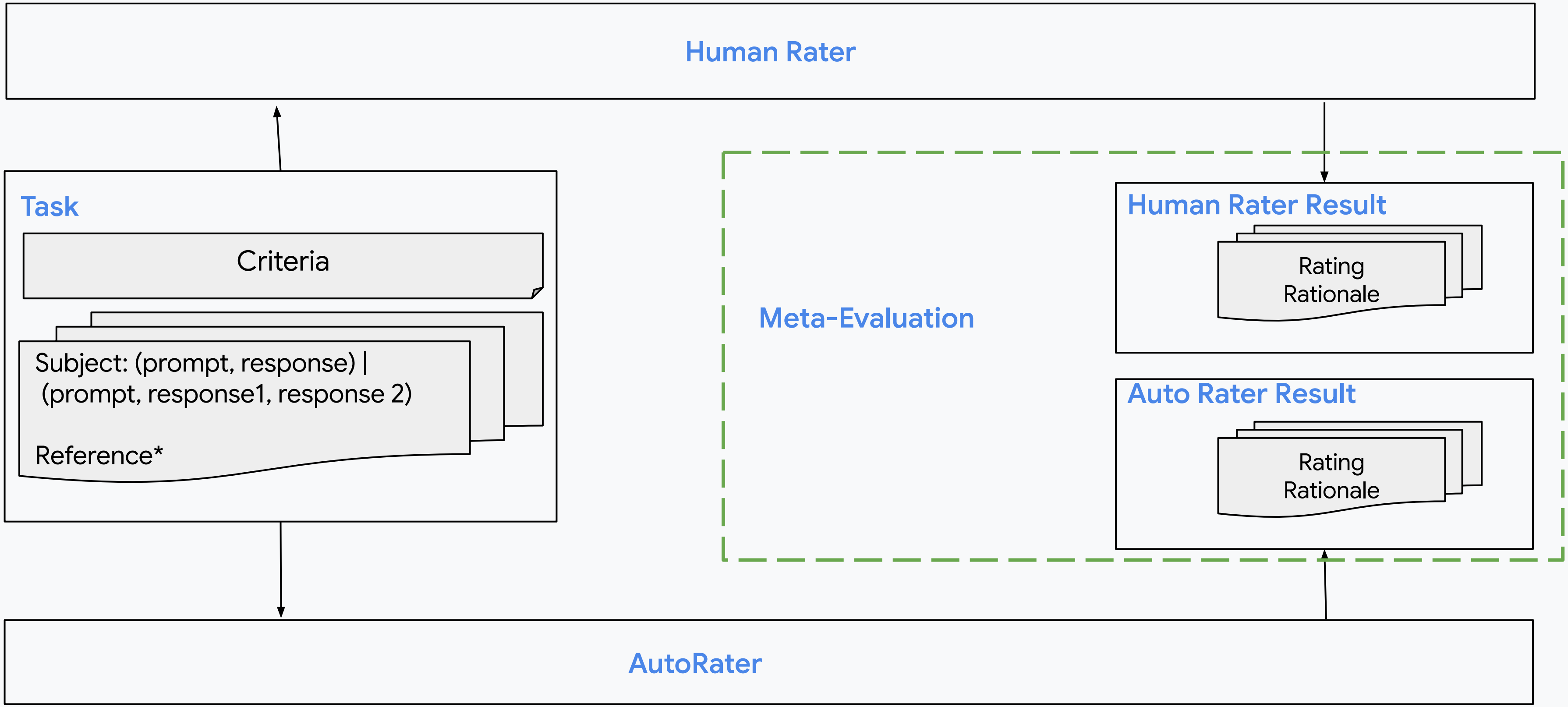
Google

AutoRater – Multiple Rater Orchestration



Reference: Juries (Verga 2024), ChatEval (Chan 2023), Agent-as-Judge (Zhuge 2024), MATEval (Li 2024),

Meta Evaluation - Overview



Google

Meta Evaluation - Metrics

- **Correlations** (Point-wise score)
  - **Spearman correlation**: Good for monotonic relationships, less sensitive to outliers.
  - **Kendall's Tau**: Suitable for ranked data and assessing concordance/discordance, handles ties well.
  - **Pearson correlation**: Best for linear relationships with normally distributed data.
- **Agreement** (Pair-wise preference)
  - **Cohen's Kappa**: Measures the agreement between two raters on categorical data, accounting for chance agreement [weight=quadric]
  - Opinions vary on how scores should be interpreted, but in general κ > 0.8 is considered a strong correlation and κ > 0.6 is a moderate correlation.
  - Confusion matrix and accuracy

Metrics	Naturalness		Coherence		Engagingness		Groundedness		Average	
	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
ROUGE-L	0.146	0.176	0.203	0.193	0.300	0.295	0.327	0.310	0.244	0.244
BLEU-4	0.175	0.180	0.235	0.131	0.316	0.232	0.310	0.213	0.259	0.189
BERTScore	0.209	0.226	0.233	0.214	0.335	0.317	0.317	0.291	0.274	0.262
G-EVAL-3.5	0.539	0.532	0.544	0.519	0.691	0.660	0.567	0.586	0.585	0.574
G-EVAL-4	0.565	0.549	0.605	**0.594**	0.631	0.627	0.551	0.531	0.588	0.575
ChatGPT(SA)	0.474	0.421	0.527	0.482	0.599	0.549	0.576	0.558	0.544	0.503
ChatGPT(MA)	0.441	0.396	0.500	0.454	0.664	0.607	0.602	0.583	0.552	0.510
GPT-4(SA)	0.532	0.483	0.591	0.535	0.734	0.676	**0.774**	**0.750**	0.658	0.611
GPT-4(MA)	**0.630**	**0.571**	**0.619**	0.561	**0.765**	**0.695**	0.722	0.700	**0.684**	**0.632**

Spearman (ρ) and Kendall-Tau (τ)

Source: [G-Eval (Liu 2023)]

Meta-Evaluation – Datasets and Benchmarks

Datasets

- MTBench and Chatbot Arena [pair-wise] Multi-turn conversations, crowdsource preference annotations.
- HelpSteer and HelpSteer2 [pair-wise] helpful, factually correct and coherent, leveraging human annotators.
- LLMBar [pair-wise] manually curated challenging meta-evaluation to assess instruction-following.
- AlpacaEval and AlpacaFarm [pair-wise], chat, low-cost simulation of pairwise feedback from API models.
- Anthropic Helpful and Anthropic HHH [pair-wise]: human alignment capability on helpful, honest, harmless.
- summarize_from_feedback [pair-wise], summary comparison.
- HuanEvalPack [point-wise] coding abilities.
- FLASK [point-wise]: fine-grained scoring with 4 primary abilities divided into 12 fine-grained skills.

Benchmarks

- RewardBench: [5 category with 27 datasets], comprehensive benchmark that covers chat, reasoning, and safety.
- LLM-AggreFact; [11 datasets] fact verification benchmark covering: fact verification, faithfulness of summary, etc.
- JudgeBench: benchmark on challenging response pairs spanning knowledge, reasoning, math, and coding.
- WildBench: WB-Reward and WB-Score with fine-grained outcomes. e.g. for pairwise comparison: much better, slightly better, slightly worse, much worse, or a tie.
- EvalBiasBench: bias benchmark
- CoBBLEr : bias benchmark

Google

Meta-Evaluation – From Benchmark to Your Task

- **Prompt curation**:
  - **Align** closely with your production usage **distribution**
  - For benchmarks such as HelpSteer, crowdsourcing is used to cover the diverse range of LLM use cases.
  - Prompts from benchmark datasets may not align with your production usage pattern. You need to build your own prompt sets (e.g., initially manually and/or sampling from production traffic).

- **Candidate Responses**:
  - Ensure candidate responses **covers** the specific model candidates you plan to deploy.
  - For benchmarks such as MT-Bench/Chatbot Arena, a wide range of models are selected to produce responses with the goal of comparing all models, which may not be necessary for you.

- **Annotation**:
  - **Quality** is critical
  - Human annotation (pay attention to inter-rater agreement)
  - Use powerful models cautiously (to avoid self-promotion bias).

AutoRater – Model Fine-tuning

Representative Models

Model	Base Model	Type	Training data	Training Method
FLAMe-24B	PaLM-2-24B (IT)	generative	100+ quality assessment tasks comprising 5M+ human judgments	Text-to-text multitask SFT
FLAMe-RM-24B; FLAMe-Opt-RM	PaLM-2-24B (IT)	discriminative	HelpSteer, PRM800K, CommitPack, HH Harmlessness (covering chat, reasoning and safety)	Fine-tuning with pairwise preference data Tail-patch fine-tuning to optimize multitask mixture
Skywork-Reward	Gemma-2-27b-it; Llama-3.1-8B	discriminative	Skywork-Reward-Preference-80K-v0.1 (HelpSteer2, OffsetBias, WildGuard, Magpie DPO series, In-house human annotation data)	BT-based pair-wise ranking loss with a few variants and careful curation and filtering of training data.
Skywork-Critic	Llama-3.1-8B-Instruct; Llama-3.1-70B-Instruct	generative	Skywork-Reward-Preference-80K-v0.1	instruction-tuning focusing on pairwise preference evaluation and general chat tasks.
Nemotron-Reward	Llama-3.1-70B-Instruct; Nemotron-4-340B	discriminative	HelpSteer2	Linear layer converts the final layer of the end token into 5 scalar values, train with MSE loss
PROMETHEUS 2	Mistral 7B & 8x7B	discriminative	PREFERENCE COLLECTION (1K score rubrics, 20K instructions & reference answers, 200K responses pairs & feedback)	SFT Joint point-wise and pair-wise training with weight merging to produce final model
InstructScore	Llama-2-7B	generative	10k raw from 100 domains	Multitask SFT over reference output and diagnostic report

AutoRater – Limitation and Mitigation

Biases

- Position bias (favor certain position)
- Verbosity/Length bias (favor longer responses)
- Self-enhancement/EGOCENTRIC bias (prefer self-generated answers)

Lack of consistency

- Prompt sensitivity
- Randomness in autorater output

Mitigation

- Prompt engineering and orchestration
  - Swapping Positions: call the AutoRater LLM twice with the order of options reversed to reduce position bias
  - Self-consistency: call the AutoRater LLM multiple times, analyze the multiple outputs generated and determine a consensus result
  - Panel of Diverse Models: use a LLM jury panel composed of disjoint model families.
  - In-context Learning: Providing a few demonstration examples of good judgments.
- Fine-tuning
  - Fine-tuning model via de-biasing dataset.

[Ref: MT-Bench (Zheng 2023), OffsetBias (Park 2024), CoBBLEr (Koo 2024), Juries (Verga 2024), Length-Controlled AlpacaEval (Dubois 2024), Position Bias (Shi 2024)]

Summary

Three Approaches to LLM Evaluation

- Computation
- Human
- AutoRater

Support Your Application and Task

- **Choose**
  - trade off between cost and quality
  - Work complementary depending on use cases

- **Customize**
  - Prompt engineering
  - Fine-tuning

- **Calibrate** (Meta Evaluation)
  - Stay truthful to your business needs
  - Fit to your domain and criteria
  - Avoid Bias

02

Hands-on Experience

Colab link to be posted on the google dev website

03

Safety Evaluation

Google

Colab link to be posted on the google dev website

04

QA

Google