AUTOMETRICS: APPROXIMATE HUMAN JUDGMENTS WITH AUTOMATICALLY GENERATED EVALUATORS

Anonymous authors

Paper under double-blind review

ABSTRACT

Evaluating user-facing AI applications remains a central challenge, especially in open-ended domains such as travel planning, clinical note generation, or dialogue. The gold standard is user feedback (e.g., thumbs up/down) or behavioral signals (e.g., retention), but these are often scarce in prototypes and research projects, or too-slow to use for system optimization. We present **AutoMetrics**, a framework for synthesizing evaluation metrics under low-data constraints. AutoMetrics combines retrieval from **MetricBank**, a collection of 48 metrics we curate, with automatically generated LLM-as-a-Judge criteria informed by lightweight human feedback. These metrics are composed via regression to maximize correlation with human signal. AutoMetrics takes you from expensive measures to interpretable automatic metrics. Across 5 diverse tasks, AutoMetrics improves Kendall correlation with human ratings by up to 33.4% over LLM-as-a-Judge while requiring fewer than 100 feedback points. We show that AutoMetrics can be used as a proxy reward to equal effect as a verifiable reward. We release the full AutoMetrics toolkit and MetricBank to accelerate adaptive evaluation of LLM applications.

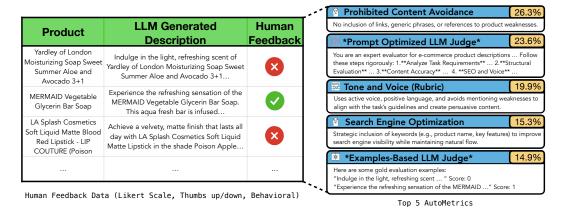


Figure 1: AutoMetrics takes you from expensive measures to interpretable automatic metrics. Here AutoMetrics generates useful metrics for evaluating LLM written product descriptions from user reviews from EvalGen (Shankar et al., 2024b). Percentages indicate relative importance of each metric derived from regression coefficients.

1 Introduction

Modern AI systems now demonstrate massively multitask capabilities imparted through extensive pretraining (Radford et al., 2019; Brown et al., 2020). Practitioners can rapidly prototype new AI-enabled tasks – from travel planning to code completion – at a pace much faster than the community can craft domain specific metrics (Papineni et al., 2002; Lin, 2004; Xu et al., 2016). This new era, in which large language models can be adapted to virtually any domain, places mounting pressure on evaluation practices. A divide is growing between tasks with easily verifiable rewards, such as math (Glazer et al., 2024; Shao et al., 2024) and coding (Chen et al., 2021), while subjective and

 open-ended tasks such as writing (Gurung & Lapata, 2025) remain difficult to measure. For these tasks, human evaluation remains the gold standard (Shankar et al., 2024b; Chiang et al., 2024).

Unfortunately, human evaluation is costly, slow, and not scalable for every prototype or user population. Reward models offer an alternative (Mnih et al., 2015; Christiano et al., 2017), but they typically require thousands of labels. The common alternative is rubric-based LLM-as-a-Judge methods (Li et al., 2023; Zheng et al., 2023; Liu et al., 2024), which rely on the assumption that system behavior is clearly defined and are not guaranteed to follow given rubrics strictly (Tripathi et al., 2025). In reality, practitioners typically have access only to non-descriptive human signals (e.g., thumbs up/thumbs down collected from users). In this setting, the problem is not only formulating the rubric, but also discovering the underlying criteria that matter.

This highlights the need for **dynamic, task-specific metric learning**. Instead of relying exclusively on human judgment or fixed rubrics, evaluation itself must become adaptive. Current efforts have emphasized making LLMs better evaluators of task-specific criteria (Liu et al., 2024; Kim et al., 2025; Anugraha et al., 2025) or leveraging rubrics to optimize LLMs (Gunjal et al., 2025; Viswanathan et al., 2025) but comparatively little work has focused on automatically generating the rubrics and criteria to be adaptively aligned with human judgment (Biyani et al., 2024; Ryan et al., 2025; Dunlap et al., 2025). Such adaptive evaluation is essential not only for easily assessing new tasks but also for optimizing evaluated systems based on real-time user feedback.

We introduce **AutoMetrics**, a method for dynamic metric induction that turns sparse, non-descriptive human feedback into actionable and interpretable evaluators (Figure 1). Starting from a task description and fewer than 100 human signals, AutoMetrics synthesizes candidate criteria, retrieves and adapts existing metrics, and composes them through regression into predictive measures of quality. Beyond simply identifying criteria, **AutoMetrics grounds and weighs them**, producing metrics that are both predictive and interpretable. This approach achieves up to **33.4% higher Kendall correlation** with human judgments than LLM-as-a-Judge baselines (§4), is **data-efficient** only requiring ~80 feedback points (§4.6), and even **matches verifiable rewards** when optimizing downstream AI systems (§5). Beyond accuracy, **AutoMetrics reveals actionable insights into what users value**. We release AutoMetrics as an open-source toolkit¹, offering the community a powerful new way to evaluate and optimize AI applications at the speed of modern development.

2 RELATED WORK

Metric Collections Prior work has organized collections of metrics primarily for the ease of use on the part of the practitioner. When already using a library such as PyTorch (Paszke et al., 2019) or Huggingface (Wolf et al., 2020) it's simple to utilize TorchMetrics (Nicki Skafte Detlefsen et al., 2022) or HuggingFace lighteval (Fourrier et al., 2023). Scikit Learn Metrics (Pedregosa et al., 2011) and NLTK metrics (Bird & Loper, 2004) were created with the same intentions. All textgeneration metrics covered by these collections are also contained in our MetricBank collection. Beyond integrating with existing open source libraries, some metric collections are part of ML observability frameworks like Evidently (Evidently AI, 2025), Galileo (Galileo, 2025), Scorecard.io (Doe & Devireddy, 2024), and DeepEval (ConfidentAI, 2025). Most metrics are tightly coupled with their observability platform, although Evidently and DeepEval offer open-source versions. While DeepEval offers a metric recommendation feature, it is based on a predefined decision tree of questions like "Does your LLM application use Retrieval-Augmented Generation (RAG)?" and "Is LLM safety a priority for you?". Most similar to our work is the MetaMetrics collection (Winata et al., 2025), which computes a regression over multiple task-specific metrics for tasks like image captioning and summarization to select the best combination of metrics. We compare our approach with MetaMetrics in Section 4 and find that our core thesis of adaptive metric generation is critical for evaluation in the low-data, novel task settings of interest.

LLM Based Evaluation LLM-as-a-Judge (Zheng et al., 2023) evaluation is increasingly popular with the frequent improvement of LLM capabilities. Several works devise task-specific prompts to enable LLM-based evaluation for storytelling (Chiang & Lee, 2023), summarization (Wang et al., 2023; Hada et al., 2024; Wu et al., 2023), dialogue (Lin & Chen, 2023; Fu et al., 2024), knowledge

¹URL withheld for anonymity

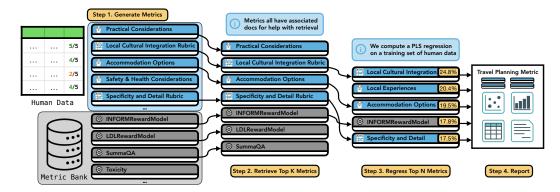


Figure 2: **AutoMetrics** comprises four steps. (1) *Generate*: create task-specific candidate metrics (Single criteria, Rubric, Examples, MIPROv2). (2) *Retrieve*: from the generated candidates plus MetricBank, use ColBERT to prefilter to k' metric cards and an LLM to select the final k. (3) *Regress*: fit a PLS model on the training set to weight and select metrics that predict human judgments. (4) *Report*: produce a writeup with weights and correlations and details to guide adoption.

(Bai et al., 2023), translation (Kocmi & Federmann, 2023), and more (Brake & Schaaf, 2024). Another promising direction is devising frameworks and general methods for making LLM-as-a-Judge more reliable. G-Eval (Liu et al., 2023) proposes breaking LLM evaluation into a step-by-step chain of thought and taking a weighted sum over the log probabilities of generating different scores. Chat-Eval (Chan et al., 2024) simulates multiple perspectives by evaluating through multi-agent debate. SPADE (Shankar et al., 2024a) generates assertions for LLMs to verify based on labeled good and bad examples. VERDICT (Kalra & Tang, 2025) introduces judge-time scaling by decomposing judgments into composable units of reasoning, verification, debate, and aggregation steps. Though we take inspiration from many of these frameworks, the most directly similar to our LLM-as-a-Judge steps in the AutoMetrics pipeline are DnA-Eval (Li et al., 2025) and EvalGen (Shankar et al., 2024b). DnA-Eval (Li et al., 2025) decomposes evaluation into rubric criteria and aggregates the results across the criteria. EvalGen (Shankar et al., 2024b) elicits limited human feedback on generated outputs, proposes criteria for evaluation based on this feedback, and iteratively refines the criteria with a human-in-the-loop and LLM.

3 THE AUTOMETRICS METHOD

The purpose of AutoMetrics is to produce metrics for subjective and novel AI-enabled tasks. Our goal is to induce metrics that correlate strongly with human judgments while requiring minimal data collection. To accomplish this, we present a general pipeline with four stages: (1) generate, (2) retrieve, (3) regress, and (4) report. These steps are visualized in Figure 2. Each stage involves design choices among several alternatives, which we empirically validate (§4.5).

3.1 METRIC PRODUCTION

Generate For sufficiently novel settings, generating criteria for LLM-as-a-Judge evaluation is essential. Broad coverage of evaluation criteria allows us to later filter down to what matters most. Accordingly, our default configuration generates 10 Single Criterion LLM Judge metrics, 5 Rubric LLM-Judge metrics, 1 Example-based optimized LLM-Judge metric (fewshot), and 1 Prompt-Optimized LLM-Judge metric per run of AutoMetrics². Optimized metrics require more LLM calls/tokens to produce, while criteria and rubrics are relatively inexpensive. Unless otherwise specified, we use this configuration throughout the paper. Empirically, we find this mix of generated metrics generalizes across diverse domains and tasks. For each metric, we also generate a Metric Card documenting its description, intended use, implementation details, and limitations (Appendix B).

²Design details and ablations are in Appendix E.2; we validate these choices across nearly 30 settings.

Retrieve In addition to generated metrics, we leverage our MetricBank: a collection of 48 metrics (Appendix Table 4) drawn from the NLP literature, each implemented and documented with a Metric Card. Directly evaluating all metrics would be prohibitively expensive, so we instead use retrieval as a filtering step. We treat Metric Cards as documents, and use a description of the evaluation setting or task as the search query. Retrieval is performed using a hybrid **ColBERT + LLM** approach, narrowing the candidate pool to metrics most relevant to the task at hand.

Regress The filtered pool of candidate metrics must still be combined into a predictive signal for human judgment. We normalize all metric scores to their z-scores and fit a **Partial Least Squares** (**PLS**) regression model. Intuitively, PLS projects the metric space onto the direction most predictive of human labels, then regresses labels along that axis. We choose PLS regression because it works well under the constraints of our setting that: (1) the number of predictors (metrics) may be comparable to or larger than the number of observations (data points), and (2) the predictors are often highly correlated. Concretely, with a single latent component, PLS finds a unit weight vector $w^* \in \mathbb{R}^d$ that maximizes

$$w^* = \arg\max_{\|w\|_2=1} \text{cov}(Xw, y)^2,$$

where X is the matrix of normalized metric scores and y is the vector of human labels. The latent score is $t = Xw^{\star}$, and PLS then regresses the human labels on this latent score, yielding predictions $\hat{y} = t\beta$ with coefficient $\beta = \frac{t^{\top}y}{t^{\top}t}$.

We apply this procedure in two stages. In the first stage, we fit PLS using all candidate metrics and rank them by the magnitude of their weights in w^* . We then select the top n metrics according to this ranking. In the second stage, we refit PLS on this reduced set of n metrics to obtain a new projection t and corresponding predictions \hat{y} . As a final step, we remove negatively correlated LLM-generated metrics, as they are designed to target positive correlation. We don't apply this to existing measures (e.g., length can negatively correlate with conciseness).

3.2 METRIC EVALUATION

To evaluate the quality of induced metrics, we draw on concepts of measurement validity from research (Borsboom et al., 2004) and testing (American Educational Research Association et al., 2014). We focus on three forms: "Content Validity", "Criterion Validity", and "Construct Validity".

Content Validity asks whether a metric represents the construct it is intended to measure. Although direct quantification is difficult, we encourage transparency by releasing metric reports. Because our generated metrics rely on LLM judges, we also expose the reasoning traces of the judge LLM, allowing users to inspect whether assessments appear justified. These traces can further aid system optimization with AutoMetrics (§5).

Criterion Validity Criterion validity measures correlation with a reference standard. In NLP, correlation with human labels has been the most widely used criterion (Banerjee & Lavie, 2005; Xu et al., 2016; Gehrmann et al., 2021). We assess criterion validity by comparing AutoMetrics to ground-truth human labels. We report Kendall's τ , which makes no distributional assumptions and simply checks whether the rank order induced by a metric matches that of human judgments. This provides a conservative estimate compared to Spearman's ρ or Pearson's r.

Construct Validity measures whether a metric captures an underlying abstract concept, such as "quality." Both human judgments and AutoMetrics attempt to approximate "quality". We draw from convergent—discriminant validity (Campbell & Fiske, 1959) and operationalize construct validity as robustness. A useful metric should penalize quality degradations (sensitivity) while remaining stable under equivalent-quality variation. In order to quantify convergent-discriminant validity, we introduce two measurements: Sensitivity and Stability. To construct test cases, we use an LLM to generate strategies for degrading outputs on a given dataset, and apply these to produce worse-quality perturbations. In contrast, same-quality perturbations are produced from a fixed set of hand-crafted transformations—such as rephrasing, reordering, synonym replacement, or stylistic edits—that are designed to preserve the target evaluation dimension. Prompts are provided in Appendix C.

³We ablate the selection algorithm in Appendix E.1.

• **Sensitivity** measures whether a metric assigns lower scores to degraded outputs. Let $s_{\text{orig}}^{(i)}$ and $s_{\text{worse}}^{(i)}$ denote the normalized scores for the original and worse-quality perturbed outputs of sample i from a dataset of size |N|. Sensitivity is defined as:

Sensitivity =
$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \left[s_{\text{worse}}^{(i)} < s_{\text{orig}}^{(i)} \right]$$

• Stability measures whether a metric produces consistent scores when quality should be preserved. Let $s_{\mathrm{same}}^{(i)}$ be the normalized score for a same-quality perturbation of sample i from a dataset of size |N|. Stability is defined as:

$$\label{eq:Stability} \text{Stability} = 1 - \frac{1}{N} \sum_{i=1}^{N} \big| s_{\text{orig}}^{(i)} - s_{\text{same}}^{(i)} \big|.$$

High sensitivity indicates strong penalization of degraded outputs, while high stability indicates invariance to irrelevant variation. Both are desirable, and together they provide a general-purpose lens for evaluating how well a metric generalizes.

4 EXPERIMENTS AND EVALUATIONS: SHOWING AUTOMETRICS ARE VALID

For our experiments, we focus on showing that our AutoMetrics are valid across many tasks/domains and that they correlate better with human judgements than competitive baselines. We showcase AutoMetrics have high Criterion Validity and Construct Validity across several tasks.

4.1 TASKS

Dataset (Citation)	Task	Domain	# Data	Feedback	# Eval Dim	Refs
In-Distribution Tasks: some metrics in our bank were designed to directly evaluate these tasks.						
SimpEval (Maddela et al., 2023) HelpSteer2 (Wang et al., 2024)	Simplification Dialogue	i	360 20,324	1–100 Likert 1–5 Likert	1 5	✓ ×
Out-of-Distribution Tasks: no metric is specifically designed for these – tests generalization and metric generation.						
EvalGen (Shankar et al., 2024b) RealHumanEval (Mozannar et al., 2025) Co-Gym (Shao et al., 2025)	Product description Code completion Travel planning	□ ・ ・	100 5,204 72	Binary Behavioral 1–5 Likert	1 1 3	X X X

Table 1: Overview of tasks. **Icons: \(\)** Code Generation; **\(\)** Data-to-Text Generation; **\(\)** Dialogue/Chat; **\(\)** Education/Readability; **\(\)** Travel Planning.

In order to evaluate our AutoMetrics method, we collect two types of tasks: *In-Distribution Tasks*, which are tasks where some of the metrics in our Metric Bank were designed to directly evaluate the task, and *Out-of-Distribution Tasks*, which are tasks where no metric in particular was designed to assess the task. All of our tasks utilize human feedback for evaluation, encompassing behavioral feedback, binary feedback (thumbs up/down), and Likert scale feedback, which is already collected as part of the dataset. We introduce all tasks in Table 1. In our main tables we present results for five datasets and a single evaluation dimension from each: **SimpEval** (Maddela et al., 2023) (sentence simplification score 1–100), **HelpSteer2** (Wang et al., 2024) (Chatbot helpfulness 1–5), **EvalGen** (Shankar et al., 2024b) (Product Review Thumbs Up/Down), **RealHumanEval** (Mozannar et al., 2025) (accepted or rejected code edit), **CoGym** (Shao et al., 2025) (travel plan outcome rating 1–5). We report evaluations on more settings in the Appendix results.

4.2 BASELINES

We include the following baselines: **Best Existing Metric**, where we run all 48 metrics (or 19 metrics for reference-free tasks), record their Kendall correlation on the validation set, and select the best metric to use for the task based on the validation correlation. **MetaMetrics**, where we take

270

282 283

284

305

300

321

322

323

all the metrics from the MetaMetrics paper and compute an XGBoost Regression on the metrics on the trainset (Winata et al., 2025). Finetuned LLM refers to training a ModernBERT-large (Warner et al., 2024) to predict the human annotation. We implement it by training LoRA adapters (Hu et al., 2021) with rank = 16 on all the attention, dense layers, and regression head, using a learning rate of 5e-5 and a batch size of 16 for three epochs over the training data. For the **LLM**-Judge baseline, we use the original human annotation prompt for each task and provide it to an LLM. We include all of these prompts in Appendix C. **DnA-Eval** (Li et al., 2025) involves using an LLM to generate three dimensions where a user request may benefit from evaluation, along with weights for how to aggregate these dimensions. Then each of those dimensions is scored with an LLM-as-a-Judge, and finally aggregated based on the LLM-generated weights.

4.3 CRITERION VALIDITY (CORRELATION)

We report Kendall's τ of all methods with GPT-40-mini and Qwen-3-32B Reasoning in Table 2.

	In-Dist	ribution	Out-of-Distribution		n
Method	SimpEval	HelpSteer2	EvalGen	RealHumanEval	CoGym
Model Agnostic					
Best Existing Metric	0.246 ± 0.00	0.327 ± 0.00	0.193 ± 0.00	0.138 ± 0.00	0.074 ± 0.00
MetaMetrics (Winata et al., 2025)	0.127 ± 0.01	0.204 ± 0.00	-0.214 ± 0.01	0.025 ± 0.01	-0.119 ± 0.02
Finetuned LLM	0.076 ± 0.08	0.039 ± 0.03	0.054 ± 0.05	0.049 ± 0.06	0.223 ± 0.20
GPT-4o-mini Backbone					
LLM-Judge	0.272 ± 0.02	0.259 ± 0.01	0.161 ± 0.14	0.069 ± 0.01	0.199 ± 0.13
DnA Eval (Li et al., 2025)	0.234 ± 0.03	0.255 ± 0.02	0.174 ± 0.16	0.152 ± 0.01	0.185 ± 0.10
AutoMetrics (Ours)	0.321 ± 0.04	$\textbf{0.324} \pm 0.01$	0.334 ± 0.06	$\textbf{0.160} \pm 0.00$	$\textbf{-0.034} \pm \textbf{0.17}$
Qwen-3-32B Backbone					
LLM-Judge	0.294 ± 0.04	0.334 ± 0.02	0.272 ± 0.13	0.025 ± 0.01	0.276 ± 0.19
DnA Eval (Li et al., 2025)	0.042 ± 0.04	0.260 ± 0.02	0.232 ± 0.19	0.071 ± 0.15	0.353 ± 0.25
AutoMetrics (Ours)	0.316 ± 0.02	$\textbf{0.342}\pm 0.01$	0.382 ± 0.05	$\textbf{0.145}\pm 0.00$	$\textbf{0.365} \pm 0.08$

Table 2: Criterion Validity results showing Kendall's Tau with 95% confidence intervals over 5 independent runs. AutoMetrics outperforms the baselines on all five tasks with Qwen3-32B and is within 95% confidence of the best for 4/5 tasks with GPT-4o-mini. On EvalGen, AutoMetrics improves performance by 33.4% over the closest baseline (LLM Judge).

AutoMetrics correlates better than all basel that AutoMetrics outperforms all other existing $\widehat{\overline{\mathbb{Q}}}^{1.0}$ best performing baseline is both incor & 0.8 SimpEval, HelpSteer, EvalGen; DnA Eval on RealHumanEval and CoGym) and on the underlying model used (Existing Metrics outperform GPT-4omini but not Qwen3-32B). In contrast, AutoMetrics is consistently the best option regardless of dataset or underlying model. On all datasets besides HelpSteer and CoGym, the AutoMetrics performance exceeds all baselines by greater than the 95% confidence interval. In general, AutoMetrics is the best choice for higher correlation with human ratings.

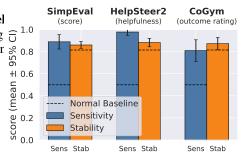


Figure 3: Sensitivity/Stability of AutoMetrics for SimpEval, HelpSteer2, and CoGym. AutoMetrics are sensitive to negative perturbations and stable on neutral perturbations.

4.4 Construct Validity (Robustness)

To measure construct validity, we take inspiration from convergent-discriminant validity and show that AutoMetrics are strong predictors when output quality degrades and that they are stable under unimportant perturbations. To do so we introduced **Sensitivity** and **Stability** (§3.2). Sensitivity measures the rate of detection of negative perturbations and Stability measures the magnitude of score preservation under meaningless changes. We report Sensitivity and Stability for all metrics on 30 trials in Figure 3. We compare against a normal distribution baseline.

AutoMetrics are sensitive and stable. AutoMetrics are sensitive to degradation in output quality in 81.0-97.8% of cases, depending on the dataset, which is significantly greater than the 50% baseline. AutoMetrics can be a strong tool for identifying degradations in output quality. Similarly, AutoMetrics also always outperforms the baseline for stability by greater than 95% confidence intervals. Under insignificant modifications to evaluated outputs, AutoMetrics are consistently stable.

4.5 DESIGN DECISIONS (HYPERPARAMETER SWEEPS)

Our sweeps/ablations test three parts of the AutoMetrics method: the MetricBank, the retrieval step, and the regression step. We report Kendall's τ rank correlation across our six main tasks with 95% confidence intervals over five runs in Table 3. All sweeps and ablations are instead done on the dev set for all datasets. We never make design decisions based on runs of our test sets.

	In-Dist	ribution		Out-of-Distribution	n
Method	SimpEval	HelpSteer2	EvalGen	RealHumanEval	CoGym
MetricBank Ablations (k=30; n=5)				
Existing Metrics Only	0.238 ± 0.04	$\underline{0.376 \pm 0.00}$	0.389 ± 0.00	$\textbf{0.155}\pm\textbf{0.00}$	0.258 ± 0.00
Generated Metrics Only	$\textbf{0.276} \pm \textbf{0.03}$	0.308 ± 0.01	$\textbf{0.503} \pm \textbf{0.03}$	0.132 ± 0.00	$\textbf{0.433} \pm \textbf{0.04}$
Full MetricBank	$\underline{0.275\pm0.02}$	$\textbf{0.387}\pm\textbf{0.00}$	0.474 ± 0.03	0.152 ± 0.01	$\underline{0.329 \pm 0.02}$
Retrieval Ablations (n=	5)				
Retrieve k=5	0.257 ± 0.03	0.336 ± 0.03	0.414 ± 0.12	0.124 ± 0.02	$\textbf{0.385} \pm \textbf{0.04}$
Retrieve k=10	0.245 ± 0.02	0.352 ± 0.01	0.469 ± 0.06	0.128 ± 0.01	0.371 ± 0.02
No Metric Cards (k=20)	0.281 ± 0.04	0.328 ± 0.02	0.427 ± 0.09	0.134 ± 0.01	0.292 ± 0.06
Retrieve k=20	$\overline{0.286\pm 0.02}$	0.378 ± 0.01	$\textbf{0.522} \pm \textbf{0.02}$	0.141 ± 0.01	0.302 ± 0.06
Retrieve k=30	0.275 ± 0.02	$\overline{\textbf{0.387} \pm \textbf{0.00}}$	0.474 ± 0.03	$\overline{\textbf{0.152}\pm\textbf{0.01}}$	0.329 ± 0.02
Regression Ablations (k	=30)				
No Regression (n=1)	0.232 ± 0.08	$\textbf{0.393} \pm \textbf{0.00}$	0.353 ± 0.23	0.145 ± 0.00	0.356 ± 0.00
Regress n=3	0.255 ± 0.02	0.389 ± 0.02	$\textbf{0.503} \pm \textbf{0.10}$	0.152 ± 0.01	0.302 ± 0.04
Regress n=5	0.275 ± 0.02	0.387 ± 0.00	0.474 ± 0.03	0.152 ± 0.01	0.329 ± 0.02
Regress n=10	$\overline{0.309 \pm 0.01}$	0.358 ± 0.01	0.461 ± 0.05	$\overline{0.147 \pm 0.01}$	0.297 ± 0.05
Regress n=20	0.268 ± 0.03	0.350 ± 0.01	0.498 ± 0.04	$\textbf{0.153}\pm\textbf{0.01}$	$\textbf{0.361} \pm \textbf{0.02}$

Table 3: Kendall correlation with 95% confidence intervals on in-distribution and out-of-distribution datasets over five runs with Qwen3 32B (Reasoning). The Full MetricBank and Metric Cards prove useful, and the best settings for retrieval and regression are k=30 and n=5 respectively.

Both Generated and Existing Metrics Help. In all of our tasks, the Full MetricBank was either the best or second-best performing setting for the ablations. When it was second best, it was typically within 95% confidence intervals. The primary exception is CoGym, where "Full MetricBank" fell 0.104 below "Generated Metrics Only" and, to a lesser extent, EvalGen, where "Full MetricBank" was short by 0.029. CoGym and EvalGen are also our smallest training sets (37 and 57 training samples respectively). We hypothesize this is because on out-of-distribution tasks, existing metrics tend to be noisy predictors which can spuriously correlate during the regression. Generated metrics tend to be less noisy predictors. Larger training sets provide a more effective filter for identifying useful metrics. We further explore this hypothesis in our data scaling experiment (§4.6).

Metric Cards Help Retrieval and Larger k Is Better. Across all five tasks, retrieval with Metric Cards (k=20) is better than retrieval without metric cards (using a single sentence description of the metric). Furthermore, we see roughly linear growth of correlation with higher k metrics retrieved to run on the train set. The single exception to this trend is CoGym, which can be attributed to the noisiness of the small dataset and the generated metrics being less noisy predictors. The top 5 retrieved metrics are often generated ones, reducing the risk of recommending spuriously correlated existing metric on the small dataset. We ran all retrieval experiments by regressing with n=5, so it is worth noting that future improvements to the retrieval algorithm (possibly including historical usage data) mean that it is feasible for k=5 numbers to match our k=30 results, so long as the proper metrics are recommended.

Number to regress to varies from dataset to dataset, but five is a good average case. The best case for regression only repeats once (with n=20), suggesting that the number of metrics needed is highly dependent on the complexity of the evaluation task and domain. Since there is no clear winner, we select n=5 as a default because it is the second best in two of five tasks, and it is the cheapest option that still maintains lower variance from run to run. A higher N means producing more expensive metrics to run downstream, so n=5 is a useful compromise of cost and performance.

4.6 How much data do you need to use AutoMetrics?

To test how much data is needed to use AutoMetrics, we test on three distinct datasets large enough to be useful in this experiment. We take a relatively simple *In-Distribution* dataset, SimpEval, a more challenging *In-Distribution* dataset, HelpSteer2, and an *Out-of-Distribution* dataset RealHumanEval. We vary the train set size from N=5, 10, 20, 40, 80, 160, and (for RealHumanEval and Helpsteer2) 320 and 640. We run these settings for both the "Generated Only" Metric Bank and "Full" Metric Bank (with existing metrics). We plot the correlation on the full test set in Figure 4.

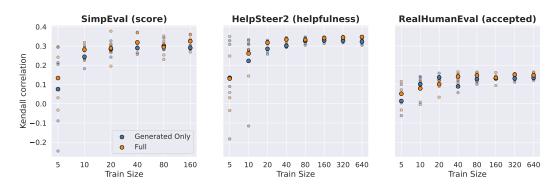


Figure 4: All correlations plotted for various training set sizes with "Generated Only" and "Full" Metric Banks. Individual trials are translucent while average performance at a scale is solid.

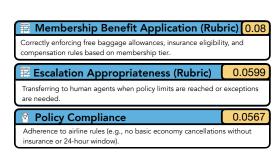
About 80 samples saturates performance. Across all three datasets and both settings, performance levels off after about 80 samples. It is possible with more sophisticated metric generation/learning methods more data could continue to help, however with the current architecture between 80-100 examples is all you need. Below 80 examples most of the lower performance is due to the high variance of fitting a regression to a small training set.

On out-of-distribution datasets "Generated Only" can outperform "Full" with low-resources. Looking to the RealHumanEval plot we see at training size 10 and 20 the "Generated Only" metrics outperform the "Full" Bank. Recall back to the ablations (§4.5) where we observed on the small, out-of-distribution datasets, CoGym and EvalGen, that "Generated Only" outperformed the "Full" MetricBank. Since most tasks will be out of distribution by nature, we default to using "Generated Only" when the user provides less than 80 training samples. Beyond 80, both "Generated Only" and "Full" level off, however "Full" asymptotes higher than "Generated Only" on all datasets.

5 Case Study: AutoMetrics for Optimizing An Agentic Task

A natural extension to using AutoMetrics is to take the limited data one has available in order to learn a useful set of metrics that can then be used for optimizing a system. In this way AutoMetrics would operate similar to the purpose of a Reward Model or a Verifiable Reward. In order to test if AutoMetrics can be useful in this setting we optimize an airline assistance agent for τ -bench (Yao et al., 2024), a testbed for tool-use agents to interact with simulated users to accomplish tasks. We split the 50 τ -airline tasks into 25 for training and 25 for evaluation.

Simulating a verifiable reward. To run AutoMetrics we rollout the 25 training examples 8 times each with temperatures [0.0, 0.01, 0.02, 0.03, 0.05, 0.1, 0.15, 0.2]. Then we obtain the true reward



433

434

435 436

437

438

439

440

441 442

443

444 445 446

447

448

449

450

451

452

453

454

455 456 457

458

459

460

461

462

463

464

465

466

467 468 469

470 471

472

473

474

475

476

477

478

479

480

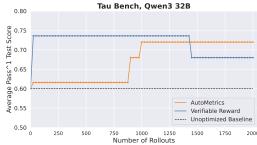
481

482

483

484

485



 τ -Bench. Regression coefficients in yellow.

Figure 5: AutoMetrics produces three metrics for Figure 6: τ -Bench performance over GEPA optimization steps when using AutoMetrics.

signal for each of these rollouts. In practice rather than a verifiable reward this could be a subjective human label. We run AutoMetrics in "Generate Only" mode and allocate more resources to generated metrics ($10\rightarrow20$ llm judge metrics; $5\rightarrow8$ rubric metrics). Otherwise we run with default hyperparameters (k=30; n=5). We show the generated metrics in Figure 5.

AutoMetrics recommends three metrics for Tau-Bench evals: two rubric based metrics and one single criterion metric. Originally our (n=5) setting recommended five metrics, however our final filtering step removed two metrics for having negative coefficients. Since the trajectories are only derived from 25 examples it is likely that metrics will begin to learn things about the data itself. This reflects the importance of both human oversight and our metric filtering.

Optimizing without a Verifiable Reward We implement a simple ReAct (Yao et al., 2023) agent in DSPy (Khattab et al., 2024) for performing the τ -Airline task. Our baseline agent gets 60% accuracy on the 25 test examples averaged over five trials. We then run a baseline optimization where we use the DSPy GEPA optimizer (Agrawal et al., 2025) to optimize an agent on the 25 training tasks with Verifiable Reward. Next we run optimization with our AutoMetrics as the metric for GEPA optimization. We show the performance on the test set after N rollouts in Figure 6. We find that AutoMetrics can match performance of a verifiable reward. After 2000 rollouts the GEPA optimization with verifiable reward achieves 0.680 ± 0.11 accuracy over 5 trials while the AutoMetrics run gets 0.720 ± 0.06 . AutoMetrics statistically significantly exceeds the baseline performance (p < 0.05) of 0.6. This demonstrates that AutoMetrics can match or exceed Verifiable Rewards as optimization signal.

DISCUSSION AND CONCLUSION

In this paper, we introduced AutoMetrics, a method for producing metrics that correlate with human judgments on subjective tasks. Requiring only ~80 human-labeled examples, AutoMetrics achieve high criterion validity (§4.3) and construct validity. (§4.4). AutoMetrics improve upon existing baselines by up to 33.4% in Kendall correlation with human ratings. In a case study on Tau-Bench, AutoMetrics matched or exceeded gains obtained from optimizing on a verifiable reward (§5).

We draw two key lessons for practitioners. First, **data diversity is critical**: while only \sim 80 feedback points suffice for moderate correlation (§4.6), scaling up synthetic data from limited sources can produce metrics that reflect dataset artifacts rather than system quality (§5). Second, human oversight remains essential: domain experts can help remove spuriously correlated metrics which the automatic filtering process misses. When using metrics for optimization, practitioners should monitor metric feedback and improvement with observability tools (Chavez, 2025).

Overall, AutoMetrics provides a practical first step for exploring data and guiding optimization when collecting preliminary human evaluation in new domains. The metrics it produces are interpretable, actionable, and informative for system improvement. We release AutoMetrics publicly and invite community contributions of new metrics and methods to strengthen the framework.

REPRODUCIBILITY STATEMENT

AutoMetrics is intended to be an open source library and framework. As such we take great effort to make the running and evaluation of AutoMetrics user-friendly. We have attached an anonymized repository for AutoMetrics with this submission. In addition to the core algorithm, the repository also contains the python scripts to reproduce all experimental results in this paper. All of our design decisions, hyperparameters, and ablations are rigorously documented throughout the paper across Section 4.5 and Appendix E. We provide system-specs needed to run the metrics in Table 5. We also share the exact prompts and DSPy signatures used in calling LLMs in Appendix C. For all main experimental results (e.g. Table 2 and Table 3) results are reported over five independent random seeds to ensure findings are robust and statistically significant.

REFERENCES

- Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2025. URL https://arxiv.org/abs/2507.19457.
- American Educational Research Association, American Psychological Association, and National Council on Measurement in Education. *Standards for Educational and Psychological Testing*. American Educational Research Association, Washington, DC, 7 edition, 2014. ISBN 978-0-935302-35-6. Prepared by the Joint Committee on the Standards for Educational and Psychological Testing of AERA, APA, and NCME.
- David Anugraha, Zilu Tang, Lester James V. Miranda, Hanyang Zhao, Mohammad Rifqi Farhansyah, Garry Kuwanto, Derry Wijaya, and Genta Indra Winata. R3: Robust rubric-agnostic reward models, 2025. URL https://arxiv.org/abs/2505.13388.
- Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. Benchmarking foundation models with language-model-as-an-examiner. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=IiRHQ7gvnq.
- Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.
- Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/P04-3031/.
- Param Biyani, Yasharth Bajpai, Arjun Radhakrishna, Gustavo Soares, and Sumit Gulwani. Rubicon: Rubric-based evaluation of domain-specific human ai conversations. In *Proceedings of the 1st ACM International Conference on AI-Powered Software*, AIware 2024, pp. 161–169, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706851. doi: 10.1145/3664646.3664778. URL https://doi.org/10.1145/3664646.3664778.
- Denny Borsboom, Gideon J Mellenbergh, and Jaap Van Heerden. The concept of validity. *Psychological review*, 111(4):1061, 2004.
- Nathan Brake and Thomas Schaaf. Comparing two model designs for clinical note generation; is an LLM a useful evaluator of consistency? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), Findings of the Association for Computational Linguistics: NAACL 2024, pp. 352–363, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024. findings-naacl.25. URL https://aclanthology.org/2024.findings-naacl.25/.

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.
 - Jose Camacho-collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, and Eugenio Martínez Cámara. TweetNLP: Cutting-edge natural language processing for social media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–49, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-demos.5.
 - Donald T Campbell and Donald W Fiske. Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological bulletin*, 56(2):81, 1959.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.
- Rogerio Chavez. Langwatch: The open llm ops platform langwatch/langwatch. https://github.com/langwatch/langwatch, 2025. Accessed: 2025-09-24.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.
- Shikai Chen, Jin Yuan, Yang Zhang, Zhongchao Shi, Jianping Fan, Xin Geng, and Yong Rui. Ldl-reward-gemma-2-27b-v0.1. https://huggingface.co/ShikaiChen/LDL-Reward-Gemma-2-27b-v0.1, 2025. Hugging Face Model Repository.
- Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15607–15631, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.870. URL https://aclanthology.org/2023.acl-long.870/.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024. URL https://arxiv.org/abs/2403.04132.
- Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- Pierre Colombo, Chloe Clave, and Pablo Piantanida. Infolm: A new metric to evaluate summarization & data2text generation. In *AAAI Conference on Artificial Intelligence*, 2021. URL https://api.semanticscholar.org/CorpusID:244896426.

- ConfidentAI. Deepeval: Open-source Ilm evaluation framework. https://github.com/confident-ai/deepeval, 2025. URL https://github.com/confident-ai/deepeval. Version 2.7.6, released April 22, 2025. Accessed April 24, 2025.
- Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. Handling divergent reference texts when evaluating table-to-text generation. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4884–4895, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1483. URL https://aclanthology.org/P19-1483/.
- George Doddington. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pp. 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- Dare Doe and Vishal Devireddy. Scorecard ai: Llm evaluation and testing platform. https://pypi.org/project/scorecard-ai/, 2024. URL https://www.scorecard.io/. Version 1.1.1, released December 10, 2024. Accessed April 24, 2025.
- Lisa Dunlap, Krishna Mandal, Trevor Darrell, Jacob Steinhardt, and Joseph E Gonzalez. Vibecheck: Discover and quantify qualitative differences in large language models, 2025. URL https://arxiv.org/abs/2410.12851.
- EvidentlyAI. Evidently: Open-source ml and llm evaluation and observability framework. https://github.com/evidentlyai/evidently, 2025. URL https://www.evidentlyai.com/. Version 0.7.2, released April 21, 2025. Accessed April 24, 2025.
- R. Flesch. Marks of readable style; a study in adult education. *Teachers College Contributions to Education*, 897:ix + 69–ix + 69, 1943.
- Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. Lighteval: A lightweight framework for llm evaluation, 2023. URL https://github.com/huggingface/lighteval.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6556–6576, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.365. URL https://aclanthology.org/2024.naacl-long.365/.
- Galileo. Galileo evaluate: Promptquality python package. https://pypi.org/project/promptquality/, 2025. URL https://pypi.org/project/promptquality/. Version 1.10.0, released April 24, 2025. Accessed April 24, 2025.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. The GEM benchmark: Natural language generation, its evaluation and metrics. In Antoine Bosselut, Esin Durmus, Varun Prashant Gangal, Sebastian Gehrmann, Yacine Jernite, Laura Perez-Beltrachini, Samira Shaikh, and Wei Xu (eds.), Proceedings of the First Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pp. 96–120, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.gem-1.10. URL https://aclanthology.org/2021.gem-1.10/.

- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai, 2024. URL https://arxiv.org/abs/2411.04872.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains, 2025. URL https://arxiv.org/abs/2507.17746.
- Alexander Gurung and Mirella Lapata. Learning to reason for long-form story generation, 2025. URL https://arxiv.org/abs/2503.22828.
- Rishav Hada, Varun Gumma, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. METAL: Towards multilingual meta-evaluation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), Findings of the Association for Computational Linguistics: NAACL 2024, pp. 2280–2298, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024. findings-naacl.148. URL https://aclanthology.org/2024.findings-naacl.148/.
- Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.
- David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3466–3495, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.211. URL https://aclanthology.org/2023.emnlp-main.211/.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. FRUIT: Faithfully reflecting updated information in text. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3670–3686, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022. naacl-main.269. URL https://aclanthology.org/2022.naacl-main.269/.
- Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. Bull Soc Vaudoise Sci Nat, 37:547–579, 1901.
- Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2289924.
- F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 08 2005. ISSN 0001-4966. doi: 10.1121/1.2016299. URL https://doi.org/10.1121/1.2016299.
- Nimit Kalra and Leonard Tang. Verdict: A library for scaling judge-time compute, 2025. URL https://arxiv.org/abs/2502.18018.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines. In *The Twelfth International Conference on Learning Representations*, 2024.
- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models, 2024.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 5877–5919, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.303. URL https://aclanthology.org/2025.naacl-long.303/.

- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.
- Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. In Mary Nurminen, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz (eds.), *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 193–203, Tampere, Finland, June 2023. European Association for Machine Translation. URL https://aclanthology.org/2023.eamt-1.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9332–9346, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.750. URL https://aclanthology.org/2020.emnlp-main.750/.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.
- Adrien Lardilleux and Yves Lepage. CHARCUT: Human-targeted character-based MT evaluation with loose differences. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pp. 146–153, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL https://aclanthology.org/2017.iwslt-1.20.
- V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10:707, 1966. English translation of Doklady Akademii Nauk SSSR, 163(4):845–848, 1965.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://aclanthology.org/N16-1014/.
- Minzhi Li, Zhengyuan Liu, Shumin Deng, Shafiq Joty, Nancy Chen, and Min-Yen Kan. DnA-eval: Enhancing large language model evaluation through decomposition and aggregation. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2277–2290, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.156/.

- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.
 - Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.
 - Yen-Ting Lin and Yun-Nung Chen. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In Yun-Nung Chen and Abhinav Rastogi (eds.), *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pp. 47–58, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlp4convai-1.5. URL https://aclanthology.org/2023.nlp4convai-1.5/.
 - Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.
 - Liu, Matei Zaharia, Ritendra Datta. Enhancing Yi and llm-as-aiudge with grading notes. https://www.databricks.com/ blog/enhancing-llm-as-a-judge-with-grading-notes, 2024. URL https://www.databricks.com/blog/ enhancing-llm-as-a-judge-with-grading-notes. Databricks Blog.
 - Chi-kiu Lo. YiSi a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 507–513, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5358. URL https://aclanthology.org/W19-5358/.
 - Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16383–16408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.905. URL https://aclanthology.org/2023.acl-long.905.
 - Xiaoyu Tan Minghao Yang, Chao Qu. Inf-orm-llama3.1-70b, 2024. URL [https://huggingface.co/infly/INF-ORM-Llama3.1-70B] (https://huggingface.co/infly/INF-ORM-Llama3.1-70B).
 - Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pp. 220–229, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287596. URL https://doi.org/10.1145/3287560.3287596.
 - Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

Hussein Mozannar, Valerie Chen, Mohammed Alsobay, Subhro Das, Sebastian Zhao, Dennis Wei, Manish Nagireddy, Prasanna Sattigeri, Ameet Talwalkar, and David Sontag. The realhumaneval: Evaluating large language models' abilities to support programmers. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=hGaWq5Buj7. Expert Certification.

- Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. TorchMetrics Measuring Reproducibility in PyTorch, February 2022. URL https://github.com/Lightning-AI/torchmetrics.
- Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *NeurIPS*, 2021.
- Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina (eds.), *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.
- Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 1776–1826, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146. 3533231. URL https://doi.org/10.1145/3531146.3533231.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Michael J. Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, William Held, and Diyi Yang. SynthesizeMe! inducing persona-guided prompts for personalized reward models in LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8045–8078, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.397. URL https://aclanthology.org/2025.acl-long.397/.

- Jon Saad-Falcon, Rajan Vivek, William Berrios, Nandita Shankar Naik, Matija Franklin, Bertie Vidgen, Amanpreet Singh, Douwe Kiela, and Shikib Mehri. Lmunit: Fine-grained evaluation with natural language unit tests, 2024. URL https://arxiv.org/abs/2412.13091.
- Patricia Schmidtova, Saad Mahamood, Simone Balloccu, Ondrej Dusek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondrej Platek, and Adarsa Sivaprasad. Automatic metrics in natural language generation: A survey of current evaluation practices. In Saad Mahamood, Nguyen Le Minh, and Daphne Ippolito (eds.), *Proceedings of the 17th International Natural Language Generation Conference*, pp. 557–583, Tokyo, Japan, September 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.inlg-main.44/.
- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3246–3256, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1320. URL https://aclanthology.org/D19-1320/.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main. 704. URL https://aclanthology.org/2020.acl-main.704/.
- Shreya Shankar, Haotian Li, Parth Asawa, Madelon Hulsebos, Yiming Lin, J. D. Zamfirescu-Pereira, Harrison Chase, Will Fu-Hinthorn, Aditya G. Parameswaran, and Eugene Wu. spade: Synthesizing data quality assertions for large language model pipelines. *Proc. VLDB Endow.*, 17(12):4173–4186, August 2024a. ISSN 2150-8097. doi: 10.14778/3685800.3685835. URL https://doi.org/10.14778/3685800.3685835.
- Shreya Shankar, J.D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya Parameswaran, and Ian Arawjo. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, UIST '24, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400706288. doi: 10.1145/3654777.3676450. URL https://doi.org/10.1145/3654777.3676450.
- Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative gym: A framework for enabling and evaluating human-agent collaboration, 2025. URL https://arxiv.org/abs/2412.15701.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.
- Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. On the evaluation metrics for paraphrase generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3178–3190, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.208. URL https://aclanthology.org/2022.emnlp-main.208/.
- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pp. 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL https://aclanthology.org/2006.amta-papers.25/.
- Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha,

Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 15725-15788, Bangkok, Thailand, August 2024. Association for Computational Linguis-tics. doi: 10.18653/v1/2024.acl-long.840. URL https://aclanthology.org/2024. acl-long.840/.

- Hong Sun and Ming Zhou. Joint learning of a dual SMT system for paraphrase generation. In Haizhou Li, Chin-Yew Lin, Miles Osborne, Gary Geunbae Lee, and Jong C. Park (eds.), *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 38–42, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-2008/.
- Tuhina Tripathi, Manya Wadhwa, Greg Durrett, and Scott Niekum. Pairwise or pointwise? evaluating feedback protocols for bias in LLM-based evaluation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=uyX5Vnow3U.
- Ken Tsui and Huu Nguyen. Low latency cpu based educational value classifier with generic educational value, 2024.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4566–4575, 2015. doi: 10.1109/CVPR.2015.7299087.
- Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1667–1682, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.132. URL https://aclanthology.org/2021.acl-long.132/.
- Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists are better than reward models for aligning language models, 2025. URL https://arxiv.org/abs/2507.18624.
- Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is ChatGPT a good NLG evaluator? a preliminary study. In Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini (eds.), *Proceedings of the 4th New Frontiers in Summarization Workshop*, pp. 1–11, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.newsum-1.1. URL https://aclanthology.org/2023.newsum-1.1/.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=PvVKUFhaNy.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv* preprint arXiv:2412.13663, 2024.
- Genta Indra Winata, David Anugraha, Lucky Susanto, Garry Kuwanto, and Derry Tanti Wijaya. Metametrics: Calibrating metrics for generation tasks using human preferences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=s103xTt4CG.

William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. 1990.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

- Ning Wu, Ming Gong, Linjun Shou, Shining Liang, and Daxin Jiang. Large language models are diverse role-players for summarization evaluation, 2023. URL https://arxiv.org/abs/2303.15078.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. URL https://arxiv.org/abs/1609.08144.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016. doi: 10.1162/tacl_a_00107. URL https://aclanthology.org/Q16-1029/.
- Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jwh9MHEfmY.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: evaluating generated text as text generation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning, 2025. URL https://arxiv.org/abs/2501.07301.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1053. URL https://aclanthology.org/D19-1053/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2023–2038, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022. emnlp-main.131. URL https://aclanthology.org/2022.emnlp-main.131/.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. *SIGIR*, 2018.

A LLM USAGE ACKNOWLEDGMENT

LLMs were used to rephrase and edit writing in the paper, after an entirely human-written first draft. LLMs were also used as coding assistants in writing code for this project. All code and writing edits produced by LLMs were rigorously verified by the first-author.

B Introducing MetricBank

MetricBank As our first significant contribution, we curate MetricBank, a standardized collection of 48 commonly used metrics in NLP literature. We source the metrics from Schmidtova et al. (2024), which examined all papers from the *International Conference on Natural Language Generation* (INLG) 2023 and all papers in the *Generation* track presented at ACL 2023, totaling 110 papers. They collected a list of all the Natural Language Generation (NLG) metrics used in those works, which totaled 283 different automatic metrics grouped into 34 metric families. We sorted by the most popular and implemented the top metrics from the top 16 families (28 metrics). Then, for completeness, we also implemented any remaining NLG metrics in NLTK Bird & Loper (2004), PyTorch Paszke et al. (2019), Huggingface Lighteval Fourrier et al. (2023), and Metametrics Winata et al. (2025) for an additional 12 metrics. Finally, we source a few additional metrics from recent papers not covered in the 2024 survey. We provide individual justifications for these 8 metrics in Appendix B.1.

We provide interesting stats about our metrics in Table 4. In particular, we collect 29 reference-based metrics, such as BLEU Papineni et al. (2002), which require a gold reference output, and 19 reference-free metrics, such as FKGL Flesch (1943), which measure quality of text without comparison to a reference. Our metrics span 12 distinct domains. We implement each metric with a simple interface of a calculate method that takes in the generated text and produces a floating-point score and optionally text feedback.

Metric Cards Inspired by Model Cards Mitchell et al. (2019) and Data Cards Pushkarna et al. (2022) we design Metric Cards for simple documentation and reporting of the intended usage of metrics. Our Metric Cards contains seven main sections. Metric Details contains the description of the metric as well as core details that are needed to use it, such as the range of outputs, if it's reference based, if an input is required, etc. Intended Use describes the domain/tasks where the metric should be used as well as recommendations for when and when not to use the metric. Metric Implementation links to reference implementations and provides guidance on practical matters about the metric such as it's efficiency and scalability. Known Limitations explains biases, misuse, and known failure cases of the metric. Related Metrics links to similar metrics to help when browsing for the right metric for your task. Further Reading points to papers, blogs, and tutorials covering the metric. Finally, Metric Card Authors makes it clear who wrote the metric card and if they used an AI assistance. We provide a complete example of a metric card for the common BLEU metric Papineni et al. (2002) in Appendix D. We also provide a prompt for using LLMs to write a first pass of a metric card in Appendix C.

Metric (Citation)	Domain	GPU	Type	Sup.	Default LLM
Reference-Based Metrics: rely on a gold reference for a	comparison.				
Jaccard Distance Jaccard (1901)	1111111111111	Х	edit-distance	Х	N.A.
Hamming Distance Hamming (1950)		X	edit-distance	X	N.A.
Levenshtein Distance Levenshtein (1966)	(#), 🖺, ध⊋, 🗹	X	edit-distance	X	N.A.
Levenshtein Ratio Levenshtein (1966)	(1), 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,	X	edit-distance	X	N.A.
Jaro Similarity Jaro (1989)		X	edit-distance	X	N.A.
Jaro–Winkler Winkler (1990)		Х	edit-distance	Х	N.A.
BLEU Papineni et al. (2002)	•	Х	n-gram overlap	Х	N.A.
NIST Doddington (2002)	₩, 🖺	Х	n-gram overlap	Х	N.A.
ROUGE Lin (2004)	■, ④, 乀, 造	Х	n-gram overlap	Х	N.A.
METEOR Banerjee & Lavie (2005)	∰, 眦, ધ⊋, 四	Х	n-gram overlap	X	N.A.
TER Snover et al. (2006)	•	X	edit-distance	X	N.A.
BLEU Sun & Zhou (2012)	1구. 👬 🗣	X	n-gram overlap	X	N.A.
CHRF++ Popović (2015)	●, 酯, ધ, 酯	X	n-gram overlap	X	N.A.
CIDEr Vedantam et al. (2015)		X	n-gram overlap	X	N.A.
GLEU Wu et al. (2016)	•	X	n-gram overlap	X	N.A.
SARI Xu et al. (2016)	B, 🕹	X	n-gram overlap	X	N.A.
CharCut Lardilleux & Lepage (2017)	.	X	edit-distance	X	N.A.
MoverScore Zhao et al. (2019)		1	embedding sim	X	BERT
PseudoPARENT Dhingra et al. (2019)		×	n-gram overlap	X	N.A.
BERTScore Zhang et al. (2020)	●, 首, 仁, 四	1	embedding sim	X	RoBERTa-Lar
BLEURT Sellam et al. (2020)		/	LM regression	Ź	BERT/RemBE
		/	LM regression	X	BART
InfoLM Colombo et al. (2021)		/	divergence-based	x	BERT
MAUVE Pillutla et al. (2021)	3 . /	/	divergence-based	x	GPT-2
ParaScore Shen et al. (2022)	17	/	embedding sim	x	RoBERTa-larg
UniEvalDialogue Zhong et al. (2022)	₽,	1	LM regression	Ź	T5
	10. 11. 11.	1	_	1	T5
UniEvalSum Zhong et al. (2022) UpdateROUGE Iv et al. (2022)		X	LM regression n-gram overlap	X	N.A.
*	8 8. 				
LENS Maddela et al. (2023)		✓	LM regression	✓	T5
Reference-Free Metrics: do not require a gold referenc					
FKGL Kincaid et al. (1975)	i , 	X	rule-based	X	N.A.
Perplexity Jelinek et al. (2005)	4 , 🌶,	✓	fluency	X	GPT-2 Large
DistinctNGrams Li et al. (2016)	2 , /	X	diversity ratio	×	N.A.
SelfBLEU Zhu et al. (2018)	🔩, 🌽, 🕰	X	diversity ratio	X	N.A.
YiSi-2 Lo (2019)	(✓	embedding sim	X	mBERT
SummaQA Scialom et al. (2019)	B	1	LM regression	1	BERT
FactCC Kryscinski et al. (2020)	B	1	LM regression	1	BERT
Toxicity Vidgen et al. (2021)	2 , / , 0	1	classification	1	RoBERTa
ParaScoreFree Shen et al. (2022)	€⊋,	1	embedding sim	X	RoBERTa-larg
Sentiment Camacho-collados et al. (2022)	2 , /	1	classification	/	RoBERTa
UniEvalFact Zhong et al. (2022)		1	LM regression	/	T5
LENS_SALSA Heineman et al. (2023)		1	LM regression	1	T5
FastTextEducationalValue Tsui & Nguyen (2024)	B, i	X	classification	1	FastText
FastTextNSFW Soldaini et al. (2024)	2 0	X	classification	1	FastText
FastTextToxicity Soldaini et al. (2024)	3 . Ď	X	classification	/	FastText
GRMRewardModel Yang et al. (2024)	Ž. Č	1	LM regression	/	Llama-3.2-3B
INFORM Reward Model 70B Minghao Yang (2024)		/	LM regression	/	Llama-3.1-70E
LDL Reward Model 27B Chen et al. (2025)	9 .	/	LM regression	/	Gemma 2-27B
MathProcessRewardModel Zhang et al. (2025)	□ , □	/	classification	1	Qwen2.5 7B
riami rocessic warmivouci Zildlig et al. (2023)	au, 2	V	ciassification	•	QWC112.3 /D

Table 4: Comparison of generative evaluation metrics. **Icons:** Machine Translation, Summarization, Paraphrasing, Dialogue/Chat, Storytelling/Creative Writing, Image Captioning/Multimodal, Safety/Moderation, Data-to-Text Generation, Education/Readability, Code Generation, Math/Problem Solving, String-Distance/Edit-Based.

B.1 Additional Metrics

Here we provide justifications for our additional metrics that we did not collect from the metric survey (Schmidtova et al., 2024), lighteval (Fourrier et al., 2023), torchmetrics (Nicki Skafte Detlefsen et al., 2022), etc.

Reward Models. We choose some of the most performant reward models off the RewardBench leaderboard (Lambert et al., 2024) at development time. The three models we used are **INFORM**-

1134	N/ -4	CDU	CDU	T: ()
1135	Metric	GPU	CPU	Time (ms)
1136	INFORMRewardModel	129.62 GB	$2.04\mathrm{GB}$	1041
1137	LDLRewardModel	104.17 GB	$2.06\mathrm{GB}$	1921
1138	GRMRewardModel	$6.02\mathrm{GB}$	1.96 GB	61
1139	UniEvalDialogue	$3.07\mathrm{GB}$	3.10 GB	262
1140	UniEvalSum	3.07 GB	3.10 GB	211
1141	UniEvalFact	$3.07\mathrm{GB}$	$3.09\mathrm{GB}$	61
1142	Perplexity_gpt2-large	$3.00\mathrm{GB}$	1.47 GB	48
1143	BLEURT	2.15 GB	2.75 GB	43
1144	BARTScore_bart-large-cnn	1.52 GB	1.34 GB	49
1145	SummaQA	1.25 GB	1.51 GB	879
1146	YiSi	687 MB	1.39 GB	35
1147	Sentiment	485 MB	1.36 GB	19
1148	Toxicity	485 MB	1.36 GB	39
1149	FactCC	427 MB	1.29 GB	17
1150	ParaScoreFree	346 MB	1.68 GB	12 428
1151	ParaScore	338 MB	1.05 GB	4 543
1152	MOVERScore_distilbert-base-uncased	262 MB	1.50 GB	2 899
1153	BERTScore_roberta-large	8 MB	1.47 GB	1 303
1154	PRMRewardModel	0 MB	13.64 GB	6359
1155	MAUVE_max	0 MB	4.22 GB	3 2 3 6
1156	FastTextEducationalValue	0 MB	3.73 GB	6
1157	LENS	0 MB	3.25 GB	3 408
1158	LENS_SALSA	0 MB	2.84 GB	426
1159	FastTextToxicity	0 MB	1.67 GB	11
1160	FastTextNSFW	0 MB	1.67 GB	6
1161	InfoLM	0 MB	1.12 GB	2338
1162	METEOR	0 MB	1.08 GB	27
1163	FKGL TER	0 MB	894 MB	6 26 064
1164	CHRF	0 MB 0 MB	731 MB 730 MB	36
1165	DistinctNGram		730 MB	19
1166	iBLEU	0 MB 0 MB	730 MB	18
1167	BLEU	0 MB	730 MB 729 MB	7
1168	LevenshteinDistance_min	0 MB	729 MB	0
1169	SelfBLEU	0 MB	729 MB	6
1170	HammingDistance_min	0 MB	729 MB	0
1171	JaroWinklerSimilarity_max	0 MB	729 MB	0
1172	GLEU	0 MB	728 MB	9
1173	SARI	0 MB	728 MB	95
1174	JaccardDistance_min	0 MB	728 MB	0
1175	CharCut	0 MB	728 MB	1 237
1176	UpdateROUGE	0 MB	728 MB	96
1177	NIST	0 MB	728 MB	21
1178	LevenshteinRatio_max	0 MB	727 MB	0
1179	JaroSimilarity_max	0 MB	727 MB	0
1180	ROUGE	0 MB	727 MB	487
1181	CIDEr_n4_sig6.0	0 MB	726 MB	31
1182	PseudoPARENT	0 MB	726 MB	10
1104		01,115	. 20 1/115	

Table 5: Maximum CI upper-bound GPU/CPU memory and latency per metric.

1183

et al., 2025), and **GRMRewardModel** (Llama 3.2 3B) (Yang et al., 2024). We also add in the Qwen2.5 7B **Process Reward Model** (Zhang et al., 2025).

SALSA. In researching text simplification metrics we found that an extension to the LENS (Maddela et al., 2023) metric exists which is meant to better align with human judgement. It was also a related metric to the SimpEval (Maddela et al., 2023) paper. Thus we chose to implement **SALSA** (Heineman et al., 2023) in our MetricBank as it was intended as one of the recommended "Best" metrics for our *in-distribution* SimpEval task.

FastText Classifiers. We wanted to add diversity to our MetricBank by including more classifiers for various higher-level concepts, but we didn't want to add unnecessary expenses to running the metrics. FastText Classifiers are a nice compromise which are quick to run on CPU but also have reasonable classification accuracy. We implement **FastTextNSFW** and **FastTextToxicity** from Dolma (Soldaini et al., 2024), and we take **FastTextEducationalValue** (Tsui & Nguyen, 2024) which has been used for data filtering to attempt to find Text-Book quality training data.

C PROMPTS AND SIGNATURES

C.1 LLM-AS-A-JUDGE PROMPTS

We use the LLM-as-a-Judge Prompts from the original human annotation process for a given dataset whenever available. We consider these as a strong baseline as these instructions were designed to be useful instructions for human annotators and ideally were the underlying instructions guiding their annotation decisions.

Task: SimpEval

LLM-as-a-Judge Prompt:

```
## Rating Sentences
The goal is to **rate sentences** by how well they **simplify the
original sentence **.
### Scoring Guidelines
| Score | When to assign it |
|-----|
| **100** | The sentence is **fully simplified**, entirely fluent,
and **preserves the core meaning** of the original. |
| **75** | The sentence is **somewhat simpler**, mostly fluent, and
the meaning is **close** to the original. |
| **50** | The sentence is simpler, **somewhat fluent**, and the
meaning is **similar** to the original. |
| **25** | The sentence is equivalently simple, still has some
fluency, but **loses the meaning**. |
         | The sentence is **completely unreadable**. |
> **Most scores will lie somewhere in this range - feel free to give
specific scores (e.g., 83, 67) rather than only the five anchors.**
### Examples
| Score | Example Simplified Sentence | Why this score? |
             -----|
| **100** | *It will then **move away from the river bed** and sink
back to the bottom to digest its food.* | Reads fluently **and**
```

keeps the original meaning ("it" gets unstuck, moves down, digests food). | | **75** | *Due to this, **a lot of mosques don't enforce theserules** but both men and women should follow them.* | Minor fluency issue, but meaning matches the original. | | **0** | *A gadget javascript a is and / checking wikipedia an snippet that can be enabled simply by , or css option in your wikipedia preferences.* | Sentence is **unreadable**. |

Task: HelpSteer2

1242 1243

1244

1245

1246

1247

1248

1253 1254

1255 1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270 1271 1272

1273 1274 1275

1276

1281 1282

1283

1288 1289

1290 1291

1292

1293

1294

1295

LLM-as-a-Judge Prompt:

```
**Helpfulness/Understanding:**
- 4 - The response is extremely helpful and completely aligned with
the spirit of what the prompt
was asking for.
- 3 - The response is mostly helpful and mainly aligned with what
the user was looking for, but
there is still some room for improvement.
- 2 - The response is partially helpful but misses the overall goal
of the user's query/input in some
way. The response did not fully satisfy what the user was looking
for.
- 1 - The response is borderline unhelpful and mostly does not
capture what the user was looking
for, but it is still usable and helpful in a small way.
- 0 - The response is not useful or helpful at all. The response
completely missed the essence of
what the user wanted.
```

Task: EvalGenProduct

LLM-as-a-Judge Prompt:

Is this response good (1) or bad (0)?

Task: RealHumanEval

LLM-as-a-Judge Prompt:

Would you accept this code edit/addition (1) or reject it (0)?

Task: CoGymTravelOutcome

LLM-as-a-Judge Prompt:

Overall rating to the final outcome (i.e., travel plan, analysis result) (1-5 scale)

- (1) "Extremely dissatisfied",
- (2) "Somewhat dissatisfied",

```
1296
1297
         (3) "Neutral",
         (4) "Somewhat satisfied",
         (5) "Extremely satisfied"
1299
1300
1301
1302
1303
      C.2 MISC PROMPTS
1304
1305
         MetricCard Generation
1306
1307
1308
        Prompt:
1309
        You are an expert in natural language processing and technical
1310
        documentation, specializing in metrics for evaluating generative
1311
        models. I am building a metric bank to recommend the best metrics
        for various generative tasks. Each metric in this bank will have a
1312
        corresponding Metric Card, which provides standardized, detailed
1313
        documentation about the metric. These Metric Cards will serve as a
1314
        key resource for researchers and practitioners, helping them select
1315
        the right metric for their task.
1316
        ## Your Task
1317
1318
        Using the provided materials, including the original paper,
1319
        reference implementations, the Metric Card Template, and the BLEU
1320
        Metric Card Example, your task is to draft a comprehensive Metric
1321
        Card for the given metric. The documentation must:
1322
                 1.
                        Follow the provided template closely, ensuring
        adherence to its format and required sections.
1323
                        Incorporate relevant details from the original paper
1324
        and reference materials, ensuring technical accuracy and
1325
        completeness.
1326
                 3.
                         Match the style and quality of the BLEU example,
1327
        which serves as an exemplar for clarity, structure, and precision.
1328
        Specific Instructions
1329
                        Key Sections to Address: Ensure each required
                1.
1330
        section of the template is filled out thoughtfully and thoroughly,
1331
        including:
                                 Metric Description
1332
                                 Inputs and Outputs
1333
                                 Formal Definition
1334
                                 Applicability and Limitations
1335
                                 Known Limitations and Related Metrics
1336
                         If Information is Unclear or Missing: Do not
        fabricate or make assumptions. If information is unavailable,
1337
        unclear, or not included in the provided context, leave that section
1338
        blank or mark it as "Needs more information."
1339
                 3.
                         Markdown Formatting: Output the completed Metric
1340
        Card as a markdown text block rather than rendering or printing the
1341
        markdown directly. This means you must surround your answer in ```.
         Also start the block with "---" as shown in the examples. Do not
1342
        end the block with "---".
1343
                        Focus on Consistency: Use the provided categorical
1344
        suggestions (see below) to ensure uniformity across all Metric
1345
```

Use \$ for inline math expressions (e.g.,

Cards, particularly in fields like "Metric Type," "Domain," and

Mathematical Formatting:

1346

1347

1348

1349

"Tasks."

\$r\$, not \$ r \$).

```
1350
1351
                                                                                  Use $$ for block math expressions and ensure
                      a full line break before and after each block math expression. This
                      formatting ensures proper rendering in markdown.
1353
                                                                                 Example of proper usage for $$:
1354
1355
                      ** Correct **
                      . . .
1356
                     Where:
1357
                      - $CHRP$ is the average precision of character and word n-grams:
1358
1359
1360
                      CHRP = \frac{1}{N} \sum_{n=1}^{N} \frac{n-1}^{n} \frac{1}{N}
1361
                      and reference}}{\text{total n-grams in hypothesis}}
1362
1363
                      - $CHRR$ is the average recall of character and word n-grams:
1364
1365
1366
                      and reference}}{\text{total n-grams in reference}}
                      $$
1368
1369
1370
                      ** Incorrect **
1371
                     Where:
1372
                      - $CHRP$ is the average precision of character and word n-grams:
1373
1374
                          1375
                      and reference}}{\text{total n-grams in hypothesis}}
1376
                      - $CHRR$ is the average recall of character and word n-grams:
1377
1378
                          CHRR = \frac{1}{N} \sum_{n=1}^{N} \frac{n-1}^{n} \frac{1}{n-q}
1379
                      and reference}}{\text{total n-grams in reference}}
1380
                         $$
                      , , ,
1381
1382
                                                              Ensure all block math expressions are clearly
1383
                      separated from list items or inline text.
1384
                                                                                 Add a space after operators like \sum, \max,
1385
                      or any LaTeX commands followed by an underscore (_) to prevent
1386
                     Markdown parsers from interpreting _ as italic markers. Mainly it
                     is critical to put a space before "_". For example:
1387
1388
                      ** Correct **
1389
                      . . .
1390
                      R _{\text{BERT}} = \frac{x _{i} \in x} \text{text}_{idf}(x _{i}) \cdot \max _{\hat{x} _{j} \in \hat{x}} x _{i^{top} \hat{x}} 
1391
1392
                      _{\{j\}}_{\sum_{x}_{x}}  \in x} \text{idf}(x _{\{i\}})}
1393
                     $$
1394
                      , , ,
1395
1396
                      ** Incorrect **
1397
1398
                      R_{\text{BERT}} = \frac{x_i \le x_i \le x}{text\{idf\}(x_i) \le text\{idf\}(x_i)}
1399
                      \max_{\hat{x}_j \in \hat{x}_i \to \hat{x}_j} (x_i \in x_i \in x
1400
                      \text{idf}(x_i)}
                     $$
1401
                      . . .
1402
1403
```

```
1405
                 6. Citation: It is imperative that you do NOT make this up.
1406
         If the user does not explicitly provide the bibtex citation for the
1407
         metric then you must say [More Information Needed]. If a citation
1408
         is provided you must copy it EXACTLY. Do NOT try to simplify any of
1409
         the components such as the author list with an ellipsis.
1410
         ## Categorical Suggestions for Consistency
1411
1412
        Note: These suggestions are not exhaustive. While you should
1413
        prioritize using the categories listed here for consistency, you may
1414
         add new categories if the metric clearly warrants them.
1415
         ### Domains
1416
1417
        These represent broad areas of application for the metric. Choose
1418
         one or more:
1419
                         Text Generation
1420
                         Speech Generation
                         Code Generation
1421
                         Multimodal Generation
1422
                         Image Captioning
1423
                         Dialogue Systems
1424
                         Storytelling
1425
         ### Tasks
1426
1427
         These are specific tasks or use cases where the metric applies.
1428
         Choose one or more:
1429
                         Machine Translation
1430
                         Summarization
                         Paraphrasing
1431
                        Data-to-Text Generation
1432
                        Image-to-Text Generation
1433
                        Dialogue Generation
1434
                         Style Transfer
1435
                         Creative Writing (e.g., poetry, storytelling)
                         Code Completion
1436
                         Response Generation
1437
1438
         ### Metric Type
1439
1440
        These classify the metric based on its design and purpose. Choose
1441
                         Surface-Level Similarity (e.g., BLEU, ROUGE)
1442
                         Semantic Similarity (e.g., BERTScore)
1443
                         Fluency (e.g., perplexity-based metrics)
1444
                         Diversity (e.g., distinct-n)
1445
                         Robustness (e.g., adversarial robustness metrics)
                         Fairness
1446
                         Faithfulness (e.g., factual consistency metrics)
1447
                         Reference-Free (e.g., coherence or novelty scoring)
1448
                         Explainability
1449
1450
         ### Inputs
1451
         These describe what the metric requires for evaluation:
1452
                         Reference-Based
1453
                         Reference-Free
1454
                         Input-Required
1455
                         Input-Optional
1456
```

```
1458
1459
        ## Materials You Will Be Provided
1460
                1.
                       Original Paper: The foundational paper introducing
        or defining the metric.
1461
                2.
                        Reference Implementations (when available):
1462
        Documentation from popular implementations (e.g., SacreBLEU README
1463
        for BLEU).
1464
                        Metric Card Template: The standardized structure for
                3.
        all Metric Cards (see below).
1465
                        BLEU Metric Card Example: A high-quality example for
                4.
1466
        reference.
1467
1468
        === TEMPLATE FOR METRIC CARDS ===
1469
        # Metric Card for {{ metric_name | default("Metric Name", true) }}
1470
1471
        {{ metric_summary | default("A brief description of the metric and
1472
        its purpose.", true) }}
1473
1474
        ## Metric Details
1475
        ### Metric Description
1476
1477
        {{ metric_description | default("Detailed explanation of the metric,
1478
        including how it is calculated and what it measures.", true) }}
1479
         1480
        Needed]", true) }}
1481
        - **Range: ** {{ metric_range | default("[More Information Needed]",
1482
        true) }}
1483
        - **Higher is Better?:** {{ higher_is_better | default("[More
1484
        Information Needed]", true) }}
        - **Reference-Based?:** {{ reference_based | default("[More
1485
        Information Needed]", true) }}
1486
        - **Input-Required?:** {{ input_required | default("[More
1487
        Information Needed]", true) }}
1488
1489
        ### Formal Definition
1490
        {{ metric_definition | default("Mathematical formula or detailed
1491
        algorithmic definition.", true) }}
1492
1493
        ### Inputs and Outputs
1494
        - **Inputs: **
1495
          {{ metric_inputs | default("Description of required inputs (e.g.,
1496
        generated text, reference text, input prompt).", true) }}
1497
1498
        - **Outputs:**
1499
          {{ metric_outputs | default("Description of the metric output
        (e.g., scalar score, distribution).", true) }}
1500
1501
        ## Intended Use
1502
1503
        ### Domains and Tasks
1504
        - **Domain: ** {{ domain | default("[More Information Needed]", true)
1505
        } }
1506
         - **Tasks:** {{ tasks | default("[More Information Needed]", true) }}
1507
1508
        ### Applicability and Limitations
1509
1510
```

```
1512
1513
         - **Best Suited For:** {{ best_suited_for | default("[More
        Information Needed]", true) }}
         - **Not Recommended For: ** {{ not_recommended_for | default("[More
1515
         Information Needed]", true) }}
1516
1517
         ## Metric Implementation
1518
         ### Reference Implementations
1519
1520
         - **Libraries/Packages:** {{ libraries | default("[More Information
1521
        Needed]", true) }}
1522
1523
         ### Computational Complexity
1524
         - **Efficiency: ** {{ efficiency | default("[More Information
1525
        Needed]", true) }}
1526
         - **Scalability:** {{ scalability | default("[More Information
1527
        Needed]", true) }}
1528
         ## Known Limitations
1529
1530
         {{ known_limitations | default("[More Information Needed]", true) }}
1531
1532
         - **Biases:** {{ biases | default("Potential biases inherent in the
        metric.", true) }}
1533
         - **Task Misalignment Risks:** {{ task_misalignment | default("[More
1534
        Information Needed]", true) }}
1535
         - **Failure Cases: ** {{ failure_cases | default("[More Information
1536
        Needed]", true) }}
1537
1538
         ## Related Metrics
1539
         {{ related_metrics | default("[More Information Needed]", true) }}
1540
1541
         ## Further Reading
1542
1543
         - **Papers:** {{ papers | default("[More Information Needed]", true)
        } }
1544
         - **Blogs/Tutorials:** {{ blogs | default("[More Information
1545
        Needed]", true) }}
1546
1547
         ## Citation
1548
         {{ bibtex_citation | default("[More Information Needed]", true) }}
1549
1550
         ## Metric Card Authors
1551
1552
         - **Authors:** {{ metric_authors | default("[More Information
1553
        Needed]", true) }}
         - **Acknowledgment of AI Assistance: **
1554
           {{ ai_assistance | default("Portions of this metric card were
1555
        drafted with assistance from generative AI. All content has been
1556
        reviewed and curated by the author to ensure accuracy.", true) }}
1557
         - **Contact:** {{ metric_contact | default("[More Information
1558
        Needed]", true) }}
1559
1560
         === BLEU Metric Card Example ===
1561
1562
         # Metric Card for BLEU
1563
1564
```

```
1566
1567
        BLEU (Bilingual Evaluation Understudy) is a widely used metric for
        evaluating the quality of text generated in tasks like machine
        translation and summarization. It measures the overlap of n-grams
1569
        between a generated text and one or more reference texts, with a
1570
        brevity penalty to penalize overly short translations. SacreBLEU, a
1571
        modern implementation, ensures reproducibility and standardization
1572
        of BLEU scores across research.
1573
        ## Metric Details
1574
1575
        ### Metric Description
1576
1577
        BLEU evaluates the quality of text generation by comparing n-grams
        in the generated output with those in one or more reference texts.
1578
        It computes modified precision for n-grams and combines scores using
1579
        a geometric mean, with a brevity penalty to ensure the length of the
1580
        generated text matches that of the references. Higher BLEU scores
1581
        indicate closer similarity to the references.
1582
        - **Metric Type:** Surface-Level Similarity
1583
        - **Range: ** 0 to 1
1584
        - **Higher is Better?:** Yes
        - **Reference-Based?:** Yes
1586
        - **Input-Required?:** No
1587
        ### Formal Definition
1588
1589
1590
        1591
        \right)
1592
        $$
1593
        where:
1594
        - $\text{BP} = \min(1, e^{1 - r/c})$ is the brevity penalty,
1595
        - $r$ is the effective reference length (based on the closest
1596
        matching reference length for each sentence),
1597
        - $c$ is the candidate translation length,
        - p_n is the modified precision for n-grams of length n,
         - $w_n$ are weights for each n-gram (commonly uniform, $w_n =
1599
        \frac{1}{N}.
1600
1601
        ### Inputs and Outputs
1602
        - **Inputs: **
1603
          - Generated text (candidate translation)
1604
          - Reference text(s) (gold-standard translations)
1605
1606
        - **Outputs:**
1607
          - Scalar BLEU score (range: 0 to 1)
1608
        ## Intended Use
1609
1610
        ### Domains and Tasks
1611
1612
        - **Domain: ** Text Generation
        - **Tasks:** Machine Translation, Summarization, Data-to-Text
1613
        Generation
1614
1615
        ### Applicability and Limitations
1616
1617
        - **Best Suited For: **
1618
```

```
1620
1621
          Structured tasks with a clear correspondence between generated and
1622
        reference texts, such as translation or summarization.
1623
         **Not Recommended For: **
1624
          Open-ended or creative generation tasks where diversity or
1625
        semantic similarity matters more than lexical overlap (e.g.,
1626
        storytelling, dialogue).
1627
        ## Metric Implementation
1628
1629
        ### Reference Implementations
1630
1631
        - **Libraries/Packages: **
           - [SacreBLEU] (https://github.com/mjpost/sacrebleu) (robust,
1632
        standard implementation)
1633
          - [NLTK] (https://www.nltk.org/api/nltk.translate.html) (basic
1634
        Python implementation)
1635

    [Hugging Face 'evaluate'] (https://huggingface.co/docs/evaluate)

1636
         (integrated metric framework)
1637
         ### Computational Complexity
1638
        - **Efficiency: **
1640
          BLEU is computationally efficient, requiring $0(n \cdot m)$
1641
        operations for n^- gram matching where n^+ is the number of words in
        the candidate text and $m$ is the number of reference words.
1642
        SacreBLEU optimizes tokenization and scoring, making it highly
1643
        suitable for large-scale evaluations.
1644
1645
        - **Scalability:**
1646
          BLEU scales well across datasets of varying sizes due to its
        simple design. SacreBLEU further supports evaluation with multiple
1647
        references, diverse tokenization schemes, and language-specific
1648
        preprocessing, making it adaptable to diverse evaluation setups.
1649
1650
        ## Known Limitations
1651
        - **Biases:**
1652
           - BLEU penalizes valid paraphrases or semantically equivalent
1653
        outputs that do not match reference n-grams exactly.
1654
          - The brevity penalty can overly penalize valid shorter outputs,
1655
        particularly for tasks where shorter text may be acceptable or even
        preferred (e.g., summarization).
1656
1657
        - **Task Misalignment Risks: **
1658
          - BLEU is not designed for evaluating tasks with high diversity in
1659
        acceptable outputs (e.g., open-ended dialogue).
1660
           - Scores depend on the quality and number of references; fewer or
1661
        inconsistent references can lead to misleading evaluations.
1662
        - **Failure Cases:**
1663
          - BLEU struggles to capture semantic adequacy beyond lexical
1664
        similarity. For instance, it cannot identify whether a translation
1665
        preserves the meaning of the original sentence if word choices
1666
        diverge significantly.
1667
         ## Related Metrics
1668
1669
         - **ROUGE: ** Often used for summarization tasks, emphasizing recall
1670
        over precision.
1671
        - **METEOR: ** Incorporates synonym matching for better semantic
        alignment.
1672
```

```
1674
1675
         - **BERTScore: ** Uses contextual embeddings for semantic similarity.
1676
         ## Further Reading
1677
1678
         - **Papers: **
1679
           - [Original BLEU Paper (Papineni et al.,
1680
         2002) ] (https://www.aclweb.org/anthology/P02-1040)
            - [SacreBLEU: A Call for Clarity in Reporting BLEU Scores (Post,
1681
         2018) ] (https://www.aclweb.org/anthology/W18-6319)
1682
1683
         - **Blogs/Tutorials:**
1684

    [Understanding BLEU] (https://machinelearningmastery.com

1685
         /calculate-bleu-score-for-text-python/)
           - [SacreBLEU Documentation] (https://github.com/mjpost/sacrebleu)
1686
1687
         ## Citation
1688
1689
         @inproceedings{papineni-etal-2002-bleu,
1690
             title = "{B}leu: a Method for Automatic Evaluation of Machine
         Translation",
             author = "Papineni, Kishore and
1692
               Roukos, Salim and
1693
               Ward, Todd and
1694
               Zhu, Wei-Jing",
             editor = "Isabelle, Pierre and
1695
               Charniak, Eugene and
1696
               Lin, Dekang",
1697
             booktitle = "Proceedings of the 40th Annual Meeting of the
1698
         Association for Computational Linguistics",
1699
            month = jul,
year = "2002",
1700
             address = "Philadelphia, Pennsylvania, USA",
1701
             publisher = "Association for Computational Linguistics",
1702
             url = "https://aclanthology.org/P02-1040/",
1703
             doi = "10.3115/1073083.1073135",
1704
             pages = "311--318"
1705
1706
         ## Metric Card Authors
1707
1708
         - **Authors: ** ANONYMOUS
1709
         - **Acknowledgment of AI Assistance: **
1710
          Portions of this metric card were drafted with assistance from
         OpenAI's ChatGPT, based on user-provided inputs and relevant
1711
         documentation. All content has been reviewed and curated by the
1712
         author to ensure accuracy.
1713
         - **Contact:** ANONYMOUS
1714
         _____
1715
         The metric you will be designing a card for is {Metric Name}
1716
1717
         === {SUPPLEMENTAL MATERIALS} ===
1718
1719
         _____
1720
         Now please write a high quality metric card for {Metric Name} given
1721
         the provided materials!
1722
1723
         Final **Important** Note: If the provided materials do not give
1724
         enough information about a particular point for the metric (e.g.
1725
         limitations or biases aren't listed) then do NOT make things up.
        You can leave blanks or "Needs more information" where needed. It
1726
1727
```

is absolutely essential not to make things up or guess when producing this documentation otherwise future researchers and engineers will be confused and led astray. Avoid making up links that you aren't fully confident in the url.

Remember to surround your answer in '''. Thanks!

C.3 DSPY SIGNATURES

Signature: **GeneratePerturbationStrategies**

Instruction:

You will be given:

- A Task description
- A Dimension to prioritize when perturbing outputs
- The Example Input, optional Example Reference, and Example Output

Instructions:

Your primary focus should be on degrading performance along the specified Dimension.

- 1. Begin with a rich reasoning paragraph (3-5 sentences) that explores a variety of ways to subtly degrade model outputs. Do not reference the specific example.
- 2. Under the heading **Strategies:**, list 1-3 numbered, high-level perturbation strategies.
 - Each strategy should be a short phrase (5-15 words) naming the category of change, followed by one concise sentence of abstract explanation.
 - Do not include concrete rewrites, instance-specific examples, or example sentences.

Inputs:

Field	Туре	Description
task	str	The task the model was originally
		trying to complete
example_sets	list[str]	Example inputs, outputs, and
		(optionally) references showing task
		performance
dimension	str	The dimension to prioritize for the
		perturbation

Outputs:

Field	Type	Description
perturbation_ strategies	list[str]	1-3 high-level strategies to test robustness

Signature: PerturbWorse

Instruction:

You will be given:

- A Task description
- A Dimension to prioritize when perturbing outputs
- The Example Input, optional Example Reference, and Model Output

 - A perturbation_strength value ("subtle" or "obvious")

- A list of perturbation_strategies to apply

Instructions:

Your goal is to apply each strategy to the Model Output and produce a degraded version that specifically harms performance along the given Dimension, using the specified strength.

Under the heading **Perturbed Outputs:**, return exactly one perturbed output per strategy.

- For **subtle** strength, introduce minimal distortion.
- For **obvious** strength, introduce more pronounced degradation.

Do **not** include any reasoning, explanations, or examples -- only the perturbed text.

In	n		ta
ln	μ	u	ιs

Field	Type	Description
task	str	The task that the model was originally trying to complete
dimension	str	The dimension to prioritize for the perturbation (this should be the aspect of the model output that is most impacted by the perturbation)
input	str	The input provided to the model
references	<pre>Union[list[str], None]</pre>	The references of good outputs (may be None)
model_output	str	The output produced by the model
perturbation_	Literal['subtle',	The strength of the perturbation (subtle
strength	'obvious']	or obvious)
perturbation_ strategies	list[str]	The perturbation strategies to use

Outputs:

Field	Type	Description
perturbed_ outputs	list[str]	Perturbed text that is worse than the original model output. Produce one perturbed output per strategy.

Signature: PerturbSame

Instruction:

You will be given:

- A Task description
- A Dimension to preserve when perturbing outputs
- The Example Input, optional Example Reference, and Model Output
- A perturbation_strength value ("subtle" or "obvious")

Instructions:

Apply a perturbation to the Model Output that **maintains** performance on the specified Dimension.

Under the heading **Perturbed Output:** return exactly one string:

- For **subtle** strength, apply a minimal change that does not impair the target Dimension.
- For **obvious** strength, apply a more noticeable change that still keeps the target Dimension intact.

 Some examples of types of perturbations would include: rephrasing, reordering, replacing words with synonyms, stylistic changes, etc. that do not impair the target Dimension.

If any change would harm the specified Dimension, simply return the original Model Output.

After producing your original plan/reasoning do **not** include any more reasoning, explanations, or examples -- only the perturbed text.

Inputs:

Field	Type	Description
task	str	The task that the model was originally trying to complete
input	str	The input provided to the model
references	Union[list[str],	The references of good outputs (may b
	None]	None)
model_output	str	The output produced by the model
perturbation_	Literal['subtle',	The strength of the perturbation (subtle
strength	'obvious']	or obvious)
dimension	str	The aspect of the model output that MUST be preserved in quality

Outputs:

Field	Туре	Description
perturbed_output	str	Perturbed text that preserves performance along the given Dimension.

LLMAsAJudgeSignature Signature:

Instruction:

Given an input text, the task description that the model was trying to follow, and a measure to rate the text on, return a score on this measure.

Inputs:

Field	Type	Description
text	Any	The input text that we want to rate.
task_description	Any	A description of the task that the mode
		was trying to solve when it generated
		the text. Could be left blank if not
		available.
measure	Any	The measure that we want to rate the
	-	text on.
suggested_range	Any	The suggested range of possible values
	-	for the measure.

Outputs:

Field	Type	Description
score	Any	The score that the text should receive on this measure.

Signature:

LLMMetricRecommendationSignature

Instruction:

- I am looking for a metric to evaluate the attached task. In particular I care about the specific target measurement that I attached.
- Please help me decide from among the metrics that I have attached documentation for which one is most relevant to the task and target.
- Please provide a ranking of the metrics from most relevant to least relevant for the task and target above.
- You can reason first about what makes a metric relevant for the task and target, and then provide your ranking.
- IMPORTANT: The final ranking should be a list of EXACT metric class
 names (no hyphens, no spaces, no extra words). Use the METRIC
 NAME not what it is called in the documentation.
- For example, use "SelfBLEU" not "Self-BLEU", use "BERTScore" not "BERT Score", use "BLEU" not "BLEU Score".
- The final ranking should just be a list of metric names, in order from most relevant to least relevant.
- The list should be exactly 'num_metrics_to_recommend' items long.

Inputs:

Field	Туре	Description
task_description	str	A description of the task that an LLM performed and that I now want to evaluate.
target	str	The specific target measurement that I want to evaluate about the task.
metric_ documentation	List[str]	A list of metric names and their documentation. The documentation will contain the metric name, as well as many details about the metric.
num_metrics_ to_recommend	int	The number of metrics to recommend. It is imperative to target this number or very very close to it. We will do more extensive filtering later.

Outputs:

Field	Type	Description
ranking	List[str]	A numbered list of EXACT metric clarames (no hyphens, no spaces, no extremords), in order from most relevant to least relevant. The list should be of length 'num_metrics_to_recommend'. You should write the number in front of the metric name (e.g '1. METRIC1_NAME', '2. METRIC2_NAME', etc.). REMEMBER: Put quotes around EACH number + metric name pair, not just one set of quotes for the full string IMPORTANT: Refer to "METRIC NAME:" for the exact name of the metric or it won't be a match.

Signature: GenerateRubricSignature

Instruction:

Given a dataset, task description, and an evaluation metric, generate a rubric for the metric scoring from 1 to 5.

Inputs:

Field	Type	Description
task_description	Any	A description of the task that the model is trying to solve.
good_examples	Any	A list of good examples of outputs for a model.
bad_examples	Any	A list of bad examples of outputs for a model.
metric_title	Any	The title of the metric.
metric_ description	Any	A description of the metric.

Outputs:

Field	Type	Description
score_one_ description	Any	A description of what a score of means. This can be a bullet poin what criteria to look for in assig score of 1.
score_two_ description	Any	A description of what a score of means. This can be a bullet poin what criteria to look for in assignments of 2.
score_three_ description	Any	A description of what a score of means. This can be a bullet poin what criteria to look for in assig score of 3.
score_four_ description	Any	A description of what a score of means. This can be a bullet point what criteria to look for in assignments of 4.
score_five_ description	Any	A description of what a score of means. This can be a bullet poin what criteria to look for in assignments of 5.

Signature: GenerateAxisOfVariationSignature

Instruction:

Given a task description, a target metric, and good/bad examples, generate a list of axes of variation which could be used to explain the differences between the good and bad examples. These axes of variation will be used as measures to evaluate the model's performance, so they should be informative and useful for the model to improve on.

Inputs:

Field	Type	Description
task_description	str	A description of the overall task the model is trying to solve.
target_name	Optional[str]	Optional hint of the target metric/column we care about. Could be 'None' or something generic like 'quality' or 'score'.
good_examples	List[str]	A list of examples with *high* quality according to the target metric.
bad_examples	List[str]	A list of examples with *low* quality according to the target metric.
num_axes_to _generate	int	The number of axes of variation to generate.

Outputs:

Field	Type	Description
axes_of _variation	List[str]	An ordered list (most-important first describing possible axes of variation Please bold the name of the axis of variation (e.g. **Axes Name**), and ALSO include a brief sentence-long explanation of the axis of variation. (e.g. **Axes Name** Brief Explanation). Please include exactly 'num_axes_to_generate' axes of variation in the output. Avoid special characters since they sometimes mesup the parsing.

D EXAMPLE METRIC CARD: BLEU

Metric Card for BLEU

2068

20692070

20712072

2073

2074

2075

2076

20772078

20792080

2081

2082

2083

20852086

20892090

2091

20922093

2094

2095209620972098

2099

2100

2101

2102

21032104

2105

BLEU (Bilingual Evaluation Understudy) is a widely used metric for evaluating the quality of text generated in tasks like machine translation and summarization. It measures the overlap of n-grams between a generated text and one or more reference texts, with a brevity penalty to penalize overly short translations. SacreBLEU, a modern implementation, ensures reproducibility and standardization of BLEU scores across research.

Metric Details

Metric Description

BLEU evaluates the quality of text generation by comparing n-grams in the generated output with those in one or more reference texts. It computes modified precision for n-grams and combines scores using a geometric mean, with a brevity penalty to ensure the length of the generated text matches that of the references. Higher BLEU scores indicate closer similarity to the references.

• Metric Type: Surface-Level Similarity

• Range: 0 to 1

Higher is Better?: YesReference-Based?: YesInput-Required?: No

Formal Definition

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

- BP = $\min(1, e^{1-r/c})$ is the brevity penalty,
- r is the effective reference length (based on the closest matching reference length for each sentence),
- c is the candidate translation length,
- p_n is the modified precision for n-grams of length n,
- w_n are weights for each n-gram (commonly uniform, $w_n = \frac{1}{N}$).

Inputs and Outputs

2106	• Inputs:
2107	Generated text (candidate translation)
2108	
2109	 Reference text(s) (gold-standard translations)
2110	• Outputs:
2111	- Scalar BLEU score (range: 0 to 1)
2112	
2113	
2114	Intended Use
2115	
2116 2117	Domains and Tasks
2118	Domain: Text Generation
2119	
2120	• Tasks: Machine Translation, Summarization, Data-to-Text Generation
2121	Applicability and Limitations
2122	
2123 2124	 Best Suited For: Structured tasks with a clear correspondence between generated and reference texts, such as translation or summarization.
2125	• Not Recommended For: Open-ended or creative generation tasks where diversity or se-
2126	mantic similarity matters more than lexical overlap (e.g., storytelling, dialogue).
2127	
2128	Matria Implamentation
2129	Metric Implementation
2130	
2131	Reference Implementations
2132	• Libraries/Packages:
2133 2134	
2135	- SacreBLEU (robust, standard implementation)
2136	- NLTK (basic Python implementation)
2137	 Hugging Face evaluate (integrated metric framework)
2138	Computational Complexity
2139	• Efficiency: BLEU is computationally efficient, requiring $O(n \cdot m)$ operations for n-gram
2140	matching where n is the number of words in the candidate text and m is the number of
2141 2142	reference words. SacreBLEU optimizes tokenization and scoring, making it highly suitable
2142	for large-scale evaluations.
2144	• Scalability: BLEU scales well across datasets of varying sizes due to its simple de-
2145	sign. SacreBLEU further supports evaluation with multiple references, diverse tokenization
2146	schemes, and language-specific preprocessing, making it adaptable to diverse evaluation
2147	setups.
2148	
2149	Known Limitations
2150	INIOWII DIIIIMMOIIS
2151	
2152	• Biases:
2153	- BLEU penalizes valid paraphrases or semantically equivalent outputs that do not
2154	match reference n-grams exactly.
2155	- The brevity penalty can overly penalize valid shorter outputs, particularly for tasks
2156	where shorter text may be acceptable or even preferred (e.g., summarization).
2157	Task Misalignment Risks:
2158 2159	- BLEU is not designed for evaluating tasks with high diversity in acceptable outputs
	(e.g., open-ended dialogue).

ences can lead to misleading evaluations.

2160

2161

2162 Failure Cases: 2163 - BLEU struggles to capture semantic adequacy beyond lexical similarity. For instance, 2164 it cannot identify whether a translation preserves the meaning of the original sentence 2165 if word choices diverge significantly. 2166 2167 2168 **Related Metrics** 2169 2170 2171 • **ROUGE:** Often used for summarization tasks, emphasizing recall over precision. 2172 • **METEOR:** Incorporates synonym matching for better semantic alignment. 2173 • **BERTScore:** Uses contextual embeddings for semantic similarity. 2174 2175 2176 **Further Reading** 2177 2178 2179 • Papers: 2180 Original BLEU Paper (Papineni et al., 2002) 2181 - SacreBLEU: A Call for Clarity in Reporting BLEU Scores (Post, 2018) 2182 • Blogs/Tutorials: 2183 Understanding BLEU 2184 2185 - SacreBLEU Documentation 2186 2187 **Metric Card Authors** 2188 2189 2190 • **Authors:** Anonymous 2191 • Acknowledgment of AI Assistance: Portions of this metric card were drafted with assistance from OpenAI's ChatGPT, based on user-provided inputs and relevant documentation. 2193 All content has been reviewed and curated by the author to ensure accuracy. 2194 • Contact: Not disclosed (anonymous) 2195 2196 2197 **AUTOMETRICS DESIGN ABLATIONS** 2198 2199 E.1 RETRIEVE 2200 2201 For our retrieval experiments we run all metrics in the MetricBank to get the ground truth kendall 2202 correlation on the development set. With this we know the true rank order of the metrics. We then perform retrieval using a set of retrieval algorithms, namely BM25, ColBERT, Faiss, and using an 2203 LLM with all documents in context. We additionally try pipelined versions of all of these retrievers 2204 feeding into an LLM. We report Recall@[1,5,10,20] and NDCG@[1,5,10,20] in Table 6 for Qwen3-2205 32B and Table 7 for GPT-4o-mini. 2206 2207 Overall we find that ColBERT \rightarrow LLMRec is consistently the best approach for retrieval, performing the best across 14/16 of our evaluation settings. Thus, we use ColBERT → LLMRec for all metric 2208 2209 retrieval in the main paper. 2210 2211 E.2 GENERATE 2212 We test eight different approaches to metric generation. Of these approaches five of them are cheap 2213 to produce, while three of them are expensive/time-consuming to produce.

- Scores depend on the quality and number of references; fewer or inconsistent refer-

		ND	CG			Re	call	
Method	@1	@5	@10	@20	@1	@5	@10	@20
BM25	0.208 ± 0.274	0.342 ± 0.171	0.427 ± 0.143	0.567 ± 0.16	0.065 ± 0.095	0.224 ± 0.146	0.418 ± 0.173	$\textbf{0.788} \pm \textbf{0.319}$
ColBERT	0.272 ± 0.293	0.343 ± 0.203	0.442 ± 0.178	0.57 ± 0.174	0.059 ± 0.092	0.212 ± 0.155	0.441 ± 0.273	0.776 ± 0.361
Faiss	0.103 ± 0.059	0.227 ± 0.144	0.326 ± 0.163	0.461 ± 0.199	0.018 ± 0.058	0.171 ± 0.204	0.353 ± 0.256	0.694 ± 0.427
LLMRec	0.31 ± 0.334	0.396 ± 0.249	0.478 ± 0.219	0.602 ± 0.196	0.088 ± 0.101	0.294 ± 0.204	0.465 ± 0.273	0.641 ± 0.264
BM25→LLMRec	0.316 ± 0.323	0.42 ± 0.226	0.498 ± 0.197	0.603 ± 0.186	0.094 ± 0.101	0.312 ± 0.179	0.494 ± 0.257	0.665 ± 0.245
$ColBERT \rightarrow LLMRec$	$\textbf{0.403} \pm \textbf{0.387}$	$\textbf{0.462}\pm\textbf{0.28}$	$\textbf{0.528} \pm \textbf{0.238}$	$\textbf{0.631} \pm \textbf{0.212}$	0.094 ± 0.101	$\textbf{0.329} \pm \textbf{0.225}$	$\textbf{0.518} \pm \textbf{0.266}$	0.694 ± 0.21
$Faiss \rightarrow LLMRec$	0.164 ± 0.186	0.324 ± 0.234	0.393 ± 0.215	0.529 ± 0.216	0.065 ± 0.095	0.247 ± 0.226	0.4 ± 0.27	0.6 ± 0.304

Table 6: Average performance (± std) across all tasks/axes using Kendall ground truth (recommendations from gwen3).

		ND		Re	call			
Method	@1	@5	@10	@20	@1	@5	@10	@20
BM25	0.208 ± 0.274	0.342 ± 0.171	0.427 ± 0.143	0.567 ± 0.16	0.065 ± 0.095	0.224 ± 0.146	0.418 ± 0.173	0.788 ± 0.319
ColBERT	0.272 ± 0.293	0.343 ± 0.201	0.42 ± 0.179	0.568 ± 0.167	0.059 ± 0.092	0.247 ± 0.191	0.429 ± 0.27	$\textbf{0.8} \pm \textbf{0.338}$
Faiss	0.098 ± 0.059	0.21 ± 0.128	0.314 ± 0.167	0.461 ± 0.19	0.012 ± 0.048	0.159 ± 0.169	0.371 ± 0.304	0.729 ± 0.378
LLMRec	0.261 ± 0.302	0.416 ± 0.249	0.502 ± 0.235	0.585 ± 0.217	0.076 ± 0.099	0.347 ± 0.243	0.518 ± 0.316	0.759 ± 0.326
BM25→LLMRec	0.206 ± 0.197	0.394 ± 0.196	0.47 ± 0.175	0.576 ± 0.162	0.076 ± 0.099	0.347 ± 0.233	0.512 ± 0.257	0.794 ± 0.297
$ColBERT \rightarrow LLMRec$	$\textbf{0.328} \pm \textbf{0.31}$	$\textbf{0.475} \pm \textbf{0.251}$	$\textbf{0.55} \pm \textbf{0.214}$	$\textbf{0.628} \pm \textbf{0.198}$	$\textbf{0.1} \pm \textbf{0.102}$	$\textbf{0.388} \pm \textbf{0.246}$	$\textbf{0.565} \pm \textbf{0.301}$	0.759 ± 0.333
Faiss→LLMRec	0.157 ± 0.124	0.325 ± 0.205	0.406 ± 0.201	0.526 ± 0.212	0.065 ± 0.095	0.276 ± 0.226	0.424 ± 0.31	0.635 ± 0.37

Table 7: Average performance (± std) across all tasks/axes using Kendall ground truth (recommendations from gpt4o-mini).

For **CodeGen** we prompt an LLM to propose "axes of variation" from high/low examples and then synthesize small, executable Python snippets that implement a scoring function (compute_score). The generated code is cleaned, validated on a sample, and—if it errors—automatically repaired once by an LLM. We support both reference-free and reference-based variants.

For **G-Eval** (Liu et al., 2023) we convert each axis into a concrete evaluation criterion, auto-generate numbered evaluation steps, and prompt an LLM judge to produce a brief rationale followed by a discrete score (1–5). We request token-level log probabilities and, at the final score position (found by scanning backward), extract the logprobs over tokens $\{1, 2, 3, 4, 5\}$, softmax-normalize, and return the probability-weighted expectation $\hat{s} = \sum_{s=1}^{5} s P(s | \text{prompt}, \text{rationale})$. Both reference-free and reference-based variants are supported.

For **Single Criteria** (Saad-Falcon et al., 2024) LLM-as-a-Judge, we show high-scoring and low-scoring data points to an LLM and ask for "axes of variation." Each axis becomes its own metric, with the LLM prompted to output an integer score from 1–5.

For **Rubric** (Gunjal et al., 2025) we add an additional step to Single Criteria where we ask an LLM to generate explanations of what 1–5 scores should contain for each rubric item.

For **Rubric** (**Prometheus**) (Kim et al., 2024) we first synthesize a five-level rubric (descriptions for scores 1–5) from dataset examples, then use a Prometheus evaluator (e.g., M-Prometheus-14B) to assign scores conditioned on that rubric. This keeps the rubric explicit while using a strong, specialized judge.

Finetune is our first expensive to produce metric. For this we fine-tune a ModernBERT-Large regression head (with LoRA/PEFT) on formatted input-output (and references when available) to directly predict the target score. We use an 80/20 train/validation split, optimize with AdamW, and save the resulting adapter as a learned metric that runs without an LLM at inference.

For **Examples** we separate the provided human-rated examples into quintiles. Based on the context length of the LLM judge we determine how many examples we can reasonably sample from each quintile without exceeding the context length. We try 5 randomly sampled sets of uniformly distributed examples as context in an LLM-judge prompt and select the set that minimizes average distance to human labels on the trainset.

For **Prompt Optimization (MIPROv2)** we run DSPy's MIPROv2 (Opsahl-Ong et al., 2024) optimizer with auto_mode="medium" on the provided data to generate informative examples and rewrite the evaluation prompt for an LLM judge.

			Cheap to Pro	duce		Ex	pensive to Prod	uce
Task (Measure)	Code Gen	G-Eval	Single Criterion	Rubric (DSPy)	Rubric (Prometheus)	Finetune	Examples	MIPROv2
In-Distribution Tasks: some metrics in our bank	were designed t	o directly evaluo	ite these tasks.					
SummEval (coherence)	0.098 ± 0.019	0.105 ± 0.023	$\textbf{0.194} \pm \textbf{0.010}$	0.173 ± 0.017	0.140 ± 0.016	0.104 ± 0.016	0.226 ± 0.019	0.227 ± 0.044
SummEval (consistency)	0.083 ± 0.018	0.102 ± 0.013	$\textbf{0.173} \pm \textbf{0.023}$	0.160 ± 0.026	0.122 ± 0.015	0.095 ± 0.042	$\textbf{0.226} \pm \textbf{0.066}$	0.199 ± 0.030
SummEval (fluency)	0.057 ± 0.016	0.076 ± 0.011	$\textbf{0.121} \pm \textbf{0.009}$	0.110 ± 0.015	0.096 ± 0.017	0.061 ± 0.016	$\textbf{0.146} \pm \textbf{0.015}$	0.136 ± 0.048
SummEval (relevance)	0.097 ± 0.025	0.144 ± 0.026	0.213 ± 0.017	0.189 ± 0.018	0.151 ± 0.014	0.067 ± 0.043	0.243 ± 0.022	0.263 ± 0.022
Primock57 (inc_plus_omi)	0.105 ± 0.036	0.086 ± 0.017	0.247 ± 0.031	0.188 ± 0.043	0.196 ± 0.025	0.090 ± 0.057	0.253 ± 0.057	0.258 ± 0.067
Primock57 (incorrect)	0.145 ± 0.073	0.060 ± 0.014	0.250 ± 0.059	0.169 ± 0.070	0.202 ± 0.026	0.026 ± 0.029	$\textbf{0.266} \pm \textbf{0.039}$	0.213 ± 0.164
Primock57 (omissions)	0.123 ± 0.059	0.061 ± 0.021	0.119 ± 0.029	0.116 ± 0.025	$\textbf{0.129} \pm \textbf{0.020}$	0.125 ± 0.077	$\textbf{0.169} \pm \textbf{0.023}$	0.122 ± 0.097
Primock57 (time_sec)	0.102 ± 0.038	0.053 ± 0.009	$\textbf{0.159} \pm \textbf{0.026}$	0.132 ± 0.016	_	0.058 ± 0.049	0.057 ± 0.050	0.129 ± 0.041
SimpEval (score)	0.100 ± 0.037	0.184 ± 0.028	0.229 ± 0.019	0.192 ± 0.012	0.155 ± 0.017	0.046 ± 0.037	0.216 ± 0.036	0.243 ± 0.130
SimpDA (fluency)	0.180 ± 0.013	0.364 ± 0.022	0.521 ± 0.014	0.511 ± 0.018	0.460 ± 0.025	0.050 ± 0.051	0.582 ± 0.017	0.583 ± 0.058
SimpDA (meaning)	0.252 ± 0.066	0.397 ± 0.030	0.590 ± 0.016	0.570 ± 0.020	0.546 ± 0.026	0.055 ± 0.038	0.632 ± 0.025	0.625 ± 0.024
SimpDA (simplicity)	0.173 ± 0.024	0.305 ± 0.033	0.523 ± 0.030	0.481 ± 0.021	0.442 ± 0.015	0.041 ± 0.058	0.584 ± 0.025	0.628 ± 0.03
HelpSteer (coherence)	0.029 ± 0.004	0.162 ± 0.027	0.229 ± 0.013	0.190 ± 0.013	_	0.014 ± 0.009	0.297 ± 0.023	0.297 ± 0.000
HelpSteer (complexity)	$0.223 \pm \textbf{0.083}$	0.122 ± 0.029	0.149 ± 0.042	0.184 ± 0.035	_	0.071 ± 0.070	$\textbf{0.221} \pm \textbf{0.050}$	0.095 ± 0.018
HelpSteer (correctness)	0.068 ± 0.007	0.270 ± 0.024	0.356 ± 0.024	0.342 ± 0.018	_	0.044 ± 0.027	0.392 ± 0.027	0.424 ± 0.009
HelpSteer (helpfulness)	0.066 ± 0.019	0.241 ± 0.018	0.333 ± 0.011	0.327 ± 0.018	_	0.049 ± 0.030	$\textbf{0.407} \pm \textbf{0.016}$	0.402 ± 0.013
HelpSteer (verbosity)	$\boldsymbol{0.290 \pm 0.028}$	0.154 ± 0.043	0.193 ± 0.051	0.252 ± 0.053	_	0.084 ± 0.031	0.406 ± 0.015	0.103 ± 0.028
HelpSteer2 (coherence)	0.024 ± 0.005	0.116 ± 0.019	$\textbf{0.154} \pm \textbf{0.005}$	0.138 ± 0.020	_	0.043 ± 0.032	$\textbf{0.192} \pm \textbf{0.016}$	0.169 ± 0.02
HelpSteer2 (complexity)	0.113 ± 0.040	0.074 ± 0.024	0.091 ± 0.013	0.100 ± 0.014	_	0.096 ± 0.000	0.335 ± 0.074	0.065 ± 0.04
HelpSteer2 (correctness)	0.052 ± 0.012	0.167 ± 0.012	0.245 ± 0.007	0.212 ± 0.016	_	0.037 ± 0.035	$\textbf{0.332} \pm \textbf{0.017}$	0.320 ± 0.01
HelpSteer2 (helpfulness)	0.068 ± 0.009	0.134 ± 0.015	$\textbf{0.217} \pm \textbf{0.019}$	0.183 ± 0.008	0.135 ± 0.008	0.026 ± 0.015	0.293 ± 0.020	0.309 ± 0.013
HelpSteer2 (verbosity)	0.224 ± 0.018	0.161 ± 0.031	0.210 ± 0.052	$\textbf{0.234} \pm \textbf{0.048}$	_	0.167 ± 0.068	$\textbf{0.432} \pm \textbf{0.015}$	0.081 ± 0.315
Out-of-Distribution Tasks: no metric is specifical	lly designed for	these – tests gen	eralization and me	tric generation.				
EvalGenProduct (grade)	0.262 ± 0.046	0.285 ± 0.029	$\textbf{0.343} \pm \textbf{0.085}$	0.303 ± 0.072	0.201 ± 0.021	0.210 ± 0.236	0.145 ± 0.046	0.216 ± 0.173
EvalGenMedical (grade)	0.262 ± 0.046	0.285 ± 0.029	$\textbf{0.343} \pm \textbf{0.085}$	0.303 ± 0.072	0.201 ± 0.021	0.210 ± 0.236	0.145 ± 0.046	0.216 ± 0.17
RealHumanEval (accepted)	0.046 ± 0.007	0.039 ± 0.013	0.115 ± 0.009	0.088 ± 0.013	0.079 ± 0.011	0.037 ± 0.025	0.091 ± 0.019	0.153 ± 0.023
CoGymTravelProcess (agentRating)	$\textbf{0.208} \pm \textbf{0.027}$	0.185 ± 0.061	0.115 ± 0.050	0.101 ± 0.044	0.121 ± 0.056	0.218 ± 0.246	0.144 ± 0.098	0.090 ± 0.046
CoGymTravelProcess (communicationRating)	0.172 ± 0.075	$\textbf{0.285} \pm \textbf{0.066}$	0.168 ± 0.098	0.167 ± 0.082	0.165 ± 0.028	0.238 ± 0.281	0.220 ± 0.154	0.180 ± 0.19
CoGymTravelOutcome (outcomeRating)	0.337 ± 0.059	0.318 ± 0.100	0.429 ± 0.068	$\textbf{0.448} \pm \textbf{0.117}$	0.413 ± 0.057	0.298 ± 0.472	$\textbf{0.558} \pm \textbf{0.131}$	0.518 ± 0.27
CoGymTabularProcess (agentRating)	0.254 ± 0.150	0.487 ± 0.132	0.538 ± 0.067	$\textbf{0.598} \pm \textbf{0.082}$	0.403 ± 0.108	0.475 ± 0.203	0.560 ± 0.395	$\textbf{0.637} \pm \textbf{0.31}$
CoGymTabularProcess (communicationRating)	0.360 ± 0.046	0.608 ± 0.088	0.791 ± 0.093	0.779 ± 0.147	$\textbf{0.798} \pm \textbf{0.049}$	_	$\textbf{0.890} \pm \textbf{0.125}$	0.787 ± 0.50
CoGymTabularOutcome (outcomeRating)	0.363 ± 0.217	0.349 ± 0.173	0.367 ± 0.160	0.201 ± 0.081	$\textbf{0.634} \pm \textbf{0.117}$	_	$0.363 \pm \textbf{0.362}$	0.200 ± 0.22
Average	0.159	0.206	0.281	0.263	0.276	0.108	0.323	0.287

Table 8: Metric generation performance (Kendall's Tau) with 95% confidence intervals over 5 independent runs. Each generator produces metrics using persistent train sets, then correlation with human annotations is measured on persistent validation sets. Cheap methods (left) generate 10 metrics per trial, expensive methods (right) generate 1 metric per trial (except finetune which generates 10). Results show correlation between generated metrics and ground-truth human annotations across diverse tasks using the Qwen3 32B model.

			Cheap to Pro	duce		Ex	pensive to Prod	uce
Task (Measure)	Code Gen	G-Eval	Single Criterion	Rubric (DSPy)	Rubric (Prometheus)	Finetune	Examples	MIPROv2
In-Distribution Tasks: some metrics in our bank were designed to directly evaluate these tasks.								
SimpEval (score)	0.127 ± 0.015	0.279 ± 0.024	0.324 ± 0.026	0.299 ± 0.024	0.166 ± 0.022	0.046 ± 0.037	0.297 ± 0.041	0.318 ± 0.039
SimpDA (fluency)	0.135 ± 0.020	0.534 ± 0.016	0.510 ± 0.014	$\textbf{0.573} \pm \textbf{0.012}$	0.460 ± 0.013	0.050 ± 0.051	$\textbf{0.639} \pm \textbf{0.028}$	0.635 ± 0.018
SimpDA (meaning)	0.246 ± 0.028	0.570 ± 0.012	0.551 ± 0.007	$\textbf{0.601} \pm \textbf{0.022}$	0.538 ± 0.014	0.055 ± 0.038	$\textbf{0.686} \pm \textbf{0.039}$	0.643 ± 0.022
SimpDA (simplicity)	0.092 ± 0.045	0.500 ± 0.016	$\textbf{0.540} \pm \textbf{0.005}$	0.535 ± 0.026	0.463 ± 0.031	0.041 ± 0.058	0.621 ± 0.039	$\textbf{0.622} \pm \textbf{0.019}$
Out-of-Distribution Tasks: no metric is specifical	ally designed for	these – tests gen	eralization and me	tric generation.				
EvalGenProduct (grade)	0.190 ± 0.032	0.204 ± 0.063	0.225 ± 0.069	0.175 ± 0.087	0.189 ± 0.080	0.210 ± 0.236	0.121 ± 0.040	0.248 ± 0.069
EvalGenMedical (grade)	0.190 ± 0.032	0.204 ± 0.063	0.225 ± 0.069	0.175 ± 0.087	0.189 ± 0.080	0.210 ± 0.236	0.121 ± 0.040	0.248 ± 0.069
CoGymTravelProcess (agentRating)	$\textbf{0.201} \pm \textbf{0.032}$	0.113 ± 0.037	0.092 ± 0.027	0.173 ± 0.041	0.127 ± 0.054	0.218 ± 0.246	$\textbf{0.223} \pm \textbf{0.108}$	0.067 ± 0.041
CoGymTravelProcess (communicationRating)	$\textbf{0.232} \pm \textbf{0.074}$	0.176 ± 0.038	0.070 ± 0.035	0.154 ± 0.083	0.228 ± 0.017	0.238 ± 0.281	$\textbf{0.312} \pm \textbf{0.086}$	0.000 ± 0.000
CoGymTravelOutcome (outcomeRating)	0.286 ± 0.049	0.400 ± 0.103	0.450 ± 0.114	$\textbf{0.474} \pm \textbf{0.051}$	0.412 ± 0.092	0.298 ± 0.472	0.502 ± 0.024	0.513 ± 0.009
CoGymTabularProcess (agentRating)	0.273 ± 0.199	0.555 ± 0.098	0.697 ± 0.139	0.555 ± 0.064	0.435 ± 0.120	0.475 ± 0.203	0.600 ± 0.000	0.659 ± 0.220
CoGymTabularProcess (communicationRating)	0.404 ± 0.138	0.853 ± 0.091	0.859 ± 0.093	$\textbf{0.881} \pm \textbf{0.020}$	0.763 ± 0.091	_	0.000 ± 0.000	0.890 ± 0.125
CoGymTabularOutcome (outcomeRating)	0.470 ± 0.227	0.547 ± 0.159	0.565 ± 0.090	0.616 ± 0.120	$\textbf{0.804} \pm \textbf{0.094}$	_	0.430 ± 0.268	$\textbf{0.623} \pm \textbf{0.329}$
Average	0.237	0.411	0.426	0.434	0.398	0.184	0.379	0.456

Table 9: Metric generation performance (Kendall's Tau) with 95% confidence intervals over 5 independent runs. Each generator produces metrics using persistent train sets, then correlation with human annotations is measured on persistent validation sets. Cheap methods (left) generate 10 metrics per trial, expensive methods (right) generate 1 metric per trial (except finetune which generates 10). Results show correlation between generated metrics and ground-truth human annotations across diverse tasks using the GPT-40 Mini model.

F ADDITIONAL EXPERIMENTS

F.1 ROBUSTNESS FOR ALL METRICS

Here we include the results of the robustness experiment for all baseline metrics tested. We report results in Figure 7.

We find that "Best Metric" tends to be quite stable, while the LLM Based Metrics (DNAEval, LLM-Judge, and AutoMetrics) stand out on robustness.

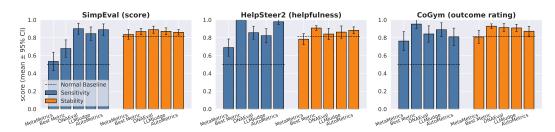


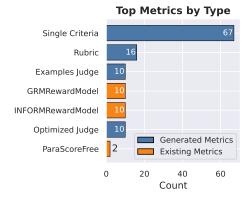
Figure 7: Sensitivity and Stability of all metrics for SimpEval, HelpSteer2, and CoGym.

F.2 What metrics does AutoMetrics actually select?

To explore the question of what metrics AutoMetrics actually recommends we turn to the 25 trials of AutoMetrics run for our main correlation experiment from Table 2. We look exclusively at the Qwen3-32B runs. We provide a bar plot of metric types in Figure 8.

AutoMetrics are dominated by Generated Met-

rics. 103 out of the 125 total recommended metrics were automatically generated. Of the Existing metrics that were recommended 20 out of 22 were recommendations to use a reward model. This suggests that the scope of metrics to retrieve from can be dramatically reduced to primarily recommending from the generated metrics as well as a few key reward models and other model based metrics like "ParaScoreFree". This insight will in practice greatly simplify the search space for metrics and lead to a more streamlined MetricBank.



G AUTOMETRICS EXAMPLES

Figure 8: Breakdown of metrics recommended by AutoMetrics. Generated are most common.

SimpEval — score

Overall Kendall τ : 0.3234

Top 5 Metrics & Coefficients

Metric	Coefficient
Audience_Appropriateness_Qwen3-32B	1.7066
Conciseness_Qwen3-32B	1.6676
Readability_Score_Qwen3-32B	1.6622
Clarity_and_Readability_Rubric	1.6345
ParaScoreFree	-1.6125

Description: Audience_Appropriateness_Qwen3-32B

Tailors language and phrasing to suit a general audience with minimal prior knowledge of the topic.

Description: Conciseness_Qwen3-32B

Eliminates redundant phrases, wordiness, or tangential details while maintaining the original intent.

Description: Readability_Score_Qwen3-32B

Measures the text's ease of reading using standardized metrics (e.g., Flesch-Kincaid Grade Level).

2430 2431 Description: Clarity_and_Readability_Rubric 2432 2433 | Score | Description | 2434 |-----| 2435 | 1 | - The text is difficult to understand due to overly complex 2436 sentence structures, passive voice, or ambiguous phrasing. 2437 - Redundant or redundant information is included, hindering 2438 clarity. - Sentences are excessively long or fragmented, making it hard to 2439 follow the main idea. 2440 - Jargon or technical terms are retained without simplification. 2441 - The output fails to restructure the original sentence for 2442 broader accessibility. | 2443 | 2 | - The text is somewhat clear but still contains occasional 2444 complex structures or passive voice. 2445 - Some sentences are overly long or include minor redundancies. 2446 - Ambiguity or unclear phrasing is present in parts of the output 2447 2448 - Simplification is attempted but incomplete, leaving some original complexity intact. 2449 - The main idea is generally understandable but requires effort 2450 to parse. | 2451 2452 | 3 | - The text is mostly clear, with mostly active voice and 2453 straightforward phrasing. 2454 - Sentences are concise and well-structured, though a few may retain slight complexity. 2455 - Minor ambiguities or redundancies are present but do not 2456 significantly hinder understanding. 2457 - Simplification is effective for the core message, though some 2458 details may remain dense. 2459 - The output is accessible to a general audience with minimal effort. | 2460 2461 | 4 | - The text is clear and uses active voice consistently, 2462 with minimal passive constructions. 2463 - Sentences are concise, well-structured, and free of unnecessary 2464 complexity. 2465 - Ambiguity is largely avoided, and phrasing is precise. - Simplification is thorough, with original complexity reduced to 2466 enhance accessibility. 2467 - The output is easy to understand for a broad audience, with 2468 only minor improvements possible. | 2469 2470 | 5 | - The text is exceptionally clear, using active voice and 2471 simple, direct sentence structures. - All phrasing is unambiguous, and sentences are optimized for 2472 readability. 2473 - Redundancy and complexity are entirely eliminated, with the 2474 core message distilled to its essentials. 2475 - Simplification is flawless, making the content immediately 2476 accessible to all audiences. - The output exemplifies best practices in clarity and 2477 readability, with no room for improvement. | 2478 2479 2480

Description: ParaScoreFree

ParaScoreFree is a reference-free evaluation metric designed for paraphrase generation. It evaluates candidate paraphrases based on semantic similarity to the input source while encouraging lexical diversity. ParaScoreFree outputs a scalar quality score that combines BERT-based semantic similarity and normalized edit distance, offering a balance between meaning preservation and surface-level rewriting. It enables paraphrase evaluation without the need for gold reference texts, making it suitable for low-resource or open-domain settings.

Coefficient

0.2046

0.1853

0.1697

0.1661

0.1625

HelpSteer2 — helpfulness

2497 2498

Overall Kendall τ : 0.3481

2499 2500 2501

Top 5 Metrics & Coefficients

2502 2503

Metric INFORMRewardModel HelpSteer2_helpfulness_Qwen3-32B_optimized_seed45 **GRMRewardModel** 2505 helpfulness_Owen3-32B_examples

2506 2507

2509 2510

Description: INFORMRewardModel

2511 2512 2513

2514 2515

2516 2517 2518

2519 2521

2522 2524

2526 2527 2528

2529 2530 2531

2532

Accuracy_and_Correctness_Qwen3-32B

The INFORM Reward Model 70B (INF-ORM-Llama3.1-70B) is a large-scale outcome reward model designed to evaluate the quality of generated conversational responses. It predicts scalar reward scores for response texts, supporting preference-based fine-grained evaluations without requiring a reference response. The model is finetuned from the Llama-3.1-70B-Instruct backbone using preference-labeled datasets, employing scaled Bradley-Terry loss to incorporate preference magnitudes.

Description: HelpSteer2_helpfulness_Qwen3-32B_optimized_seed45

Given the task description, evaluation axis, input/output texts, and suggested score range, analyze the output text's alignment with the task and axis by:

- 1. **Assessing factual accuracy**: Verify if claims in the output are correct and supported by the input/text domain knowledge.
- 2. **Evaluating relevance**: Determine if the output addresses the user's intent directly, avoiding verbosity or tangential content.
- 3. **Analyzing structure and clarity**: Check if explanations are concise, logically organized, and accessible to the target audience (e.g., non-experts).
- 4. **Identifying gaps or errors**: Highlight missing key details, misinterpretations, or inaccuracies that reduce helpfulness.
- 5. **Scoring**: Assign a numerical score within the suggested range, balancing the above factors.
- Use the conversation history and task description as guidance for context and expectations. Prioritize precision in reasoning and alignment with the evaluation axis.
- *Showing 0 of 8 total examples.*

Description: GRMRewardModel

2545 2546

2547 2548 2549

2551 2552 2553

2550

2554 2555 2557

2559 2560 2561

2563 2564 2565

2566 2567 2568

2569 2570 2571

2572 2573 2574

2576 2577 2578

2579

2575

2580 2581 2582

2583 2584 2585

2587 2588 2589

2586

2590

The GRMRewardModel is a general-purpose reward model designed to evaluate the quality and safety of LLM-generated outputs. It achieves high generalization performance by applying a novel regularization method on hidden states during supervised fine-tuning. GRMRewardModel is fine-tuned on the decontaminated Skywork/Skywork-Reward-Preference-80K-v0.2 dataset and achieves state-of-the-art results among models of comparable size (3B), even outperforming some 8B reward models and proprietary LLM judges on RewardBench.

Description: helpfulness_Qwen3-32B_examples

```
| Input Text | Score
   -----|
```

- | <Input (prompt): <Can you teach me semi-definite programming in simple language? > Output (response): <Can you teach me how to use a computer in simple language?>> | 0 |
- | <Input (prompt): <Delve into the nuanced benefits of engaging in group projects instead of the solo endeavor of individual projects. Craft your insights in a well-organized format, employing distinct headings for each category. Populate each section with a thoughtful list, elucidating each approach's merits and drawbacks. This approach aims to enhance clarity and the discussion's overall academ... | 0 |
- | <Input (prompt): <The misery of life never appears in a clearer light than when a thinking person has quite plainly seen with horror its hazards and uncertainties and the total darkness in which he lives; how he cannot find anything solid, secure, and beyond dispute on to which he can hold; when, as I say, after such thoughts he does not at once destroy an existence that is not one, but breathi... | 1 |

Showing 3 of 10 total examples.

Description: Accuracy_and_Correctness_Qwen3-32B

The factual correctness and reliability of the information provided.

EvalGenProduct — grade

Overall Kendall τ : 0.4178

Top 5 Metrics & Coefficients

Metric	Coefficient
Formatting_Compliance_Qwen3-32B	0.1144
grade_Qwen3-32B_examples	0.1022
Call_to_ActionCTAStrength_Qwen3-32B	0.0752
Customer_Review_Integration_Rubric	0.0747
Avoidance_of_Weaknesses_Qwen3-32B	0.0653

Description: Formatting_Compliance_Qwen3-32B

Good examples strictly follow Markdown structure (headers, bullet points). Bad examples include disallowed elements (links, markdown errors).

Description: grade_Qwen3-32B_examples

| Input Text | Score |

- | <Input (Prompt): <You are an expert copywriter. You need to write an e-commerce product description based on the product details and customer reviews. Your description should be SEO-optimized. It should use an active voice and include the product's features, benefits, unique selling points without overpromising, and a call to action for the buyer. Benefits describe how product features will wor... | 0 |
- | <Input (Prompt): <You are an expert copywriter. You need to write an e-commerce product description based on the product details and customer reviews. Your description should be SEO-optimized. It should use an active voice and include the product's features, benefits, unique selling points without overpromising, and a call to action for the buyer. Benefits describe how product features will wor... | 0 |
- | <Input (Prompt): <You are an expert copywriter. You need to write an e-commerce product description based on the product details and customer reviews. Your description should be SEO-optimized. It should use an active voice and include the product's features, benefits, unique selling points without overpromising, and a call to action for the buyer. Benefits describe how product features will wor... | 1 |
- *Showing 3 of 4 total examples.*

Description: Call_to_Action__CTA__Strength_Qwen3-32B

Good examples include urgent, benefit-driven CTAs (e.g., 'Order now for seasonal savings'), while bad examples have vague or missing CTAs.

2646 2647 Description: Customer_Review_Integration_Rubric 2648 2649 | Score | Description | 2650 |----|----2651 | 1 | - **No customer reviews included** or all quotes are 2652 fabricated. 2653 - Reviews are irrelevant to the product or its benefits. - Over-cites testimonials (e.g., 5+ quotes) or includes negative 2654 feedback. 2655 - Quotes are generic (e.g., "Great product!") without specific 2656 context. | 2657 | 2 | - **Minimal or inconsistent use of customer reviews** (e.g 2658 ., 1-2 quotes). - Quotes are vague or lack specificity (e.g., "I love this 2659 product!"). 2660 - Reviews may include irrelevant details or fail to align with 2661 the product's features/benefits. 2662 - No clear connection between testimonials and the product's 2663 unique selling points. | | 3 | - **Moderate use of customer reviews** (e.g., 2-3 quotes). 2664 - Some quotes are specific and relevant (e.g., "This product 2665 works well for dry skin"). - May include 1-2 generic or slightly over-cited testimonials. 2667 - Reviews are integrated but do not strongly enhance the 2668 description's persuasiveness. \mid 4 \mid - **Effective use of 1-2 authentic, contextually relevant 2669 quotes * * . 2670 - Testimonials highlight specific benefits (e.g., "The 2671 lightweight formula makes it perfect for travel"). 2672 - Quotes are concise, avoid over-citing, and align with the 2673 product's features. - Reviews are integrated naturally into the description without 2674 overwhelming the reader. | 2675 | 5 | - **Excellent integration of 1-2 highly specific, authentic 2676 testimonials**. 2677 · Quotes directly tie to the product's unique selling points (e.g ., "The smudge-proof formula lasts all day"). 2678 - Reviews are concise, impactful, and enhance the description's 2679 credibility. 2680 - No fabricated, irrelevant, or over-cited quotes; testimonials 2681 feel organic and persuasive. | 2683 *Showing 3 of 4 total examples.* 2684 2686 Description: Avoidance_of_Weaknesses_Qwen3-32B 2687 2688 Good examples omit product drawbacks. Bad examples inadvertently mention flaws (e.g., 2689

'may clog pores') or use hedging language.

RealHumanEval — accepted

2696

2697 2698

Overall Kendall τ : 0.1487 **Top 5 Metrics & Coefficients**

		Coefficien
GRMRewardModel		0.032
INFORMRewardModel		0.029
Code_Readability_Qwen3	3-32B	0.028
RealHumanEval_accepte	d_Qwen3-32B_optimized_seed44	0.023
Modularity_and_Reusab	ility_Qwen3-32B	0.021
Description: GRMRewardM	lode1	
The GRMRewardModel is	a general-purpose reward model designed generated outputs. It achieves high genera	

Description: INFORMRewardModel

The INFORM Reward Model 70B (INF-ORM-Llama3.1-70B) is a large-scale outcome reward model designed to evaluate the quality of generated conversational responses. It predicts scalar reward scores for response texts, supporting preference-based fine-grained evaluations without requiring a reference response. The model is finetuned from the Llama-3.1-70B-Instruct backbone using preference-labeled datasets, employing scaled Bradley-Terry loss to incorporate preference magnitudes.

Description: Code_Readability_Qwen3-32B

Clarity of variable names, structure, and comments for maintainability.

Description: RealHumanEval_accepted_Qwen3-32B_optimized_seed44

You are an expert Python code reviewer in a high-stakes software engineering environment where code correctness directly impacts mission-critical systems (e.g., financial transactions , medical devices, or autonomous vehicles). Your task is to evaluate the AI-generated code output for **absolute correctness** and **completeness** along the specified evaluation axis. A single error could lead to catastrophic failures. Analyze the code with extreme rigor, checking for: 1. **Logical correctness** (does it solve the task as described?) 2. **Syntax validity** (Python 3 compliance, no placeholders like 'xrange()' or 'raw_input()') 3. **Edge case handling** (negative numbers, empty inputs, etc.) 4. **Mathematical/statistical rigor** (valid algorithms, no arbitrary values like 'b = 8') 5. **Functionality** (working return statements, no stubs or incomplete logic). Assign a score between 0.0 and 1.0, where 0.0 means the code is non-functional or completely ignores the task, and 1.0 represents a flawless implementation. Use the input/output text and conversation history for context. *Showing 0 of 8 total examples.*

Description: Modularity_and_Reusability_Qwen3-32B

Code organization into reusable functions/methods with clear separation of concerns.

Overall Kendall τ : 0.4301

CoGymTravelOutcome — outcomeRating

Top 5 Metrics & Coefficients

Metric Cultural_and_Local_Integration_Rubric

Accommodation_Options_Qwen3-32B outcomeRating_Qwen3-32B_examples Feasibility_and_Realism_Qwen3-32B

Cultural_and_Local_Experiences_Qwen3-32B 0.1927 0.1824 0.1674 0.1620

Coefficient

0.1963

2809 Description: Cultural_and_Local_Integration_Rubric 2810 2811 | Score | Description | 2812 |-----| 2813 | 1 | - **Score 1 (Poor):** 2814 - No mention of unique local experiences or cultural highlights. 2815 - No authentic food/dining recommendations. 2816 - Generic or irrelevant suggestions (e.g., luxury dining for a budget trip). 2817 - Fails to address the user's query or intent. | 2818 | 2 | - **Score 2 (Weak):** 2819 - Minimal mention of local experiences (e.g., 1-2 generic 2820 activities like "visiting a market"). - Vague food/dining suggestions (e.g., "try local cuisine" 2821 without specifics). 2822 - Lacks integration of cultural or seasonal traditions (e.g., no 2823 mention of KFC for Christmas). 2824 - Missing links or references to local resources. | 2825 | 3 | - **Score 3 (Fair):** - Includes 1-2 specific local experiences (e.g., visiting 2826 Jigokudani Monkey Park). 2827 - Mentions 1-2 authentic food/dining options (e.g., "try miso 2828 ramen"). 2829 - Some cultural or seasonal references (e.g., "KFC is popular for 2830 Christmas"). - Limited use of links or resources to support recommendations. | 2831 | 4 | - **Score 4 (Good):** - Includes 3-4 unique local experiences (e.g., snow monkeys, 2833 winter illuminations, regional festivals). 2834 - Highlights 2-3 specific, culturally significant food/dining options (e.g., "try KFC for Christmas," "visit a local ramen 2835 shop"). 2836 - Integrates cultural/seasonal traditions (e.g., "Christmas 2837 markets in Hokkaido"). 2838 - Provides 1-2 links to local events, businesses, or resources. | 2839 | 5 | - **Score 5 (Excellent): ** - Includes 5+ unique, deeply integrated local experiences (e.g., 2840 snow monkeys, winter illuminations, regional festivals, and 2841 lesser-known gems). 2842 - Highlights 3+ specific, culturally significant food/dining 2843 options with detailed descriptions (e.g., "try miso ramen at [2844 specific shop]"). 2845 - Seamlessly integrates cultural/seasonal traditions (e.g., "KFC for Christmas, winter illuminations in Sapporo"). 2846 - Provides 3+ links to local events, businesses, or resources (e. 2847 g., [Sapporo White Illumination] (url)). 2848 - Tailors recommendations to the user's preferences and avoids 2849 irrelevant suggestions. | 2850 2851

Description: Cultural_and_Local_Experiences_Qwen3-32B

Incorporates authentic local traditions, cuisine, and unique regional activities.

Description: Accommodation_Options_Qwen3-32B

2852

2853

2854 2855

2856 2857

2858 2859 2860 Recommends diverse lodging choices (e.g., luxury hotels, budget hostels, or ryokans).

Description: outcomeRating_Qwen3-32B_examples

```
| Input Text | Score |
 <Input (query): <I want to go to Hong Kong and spend less money</pre>
   > Output (outcome): <**Luxury Entertainment** **Fine Dining
   :** * Michelin-starred restaurants: Hong Kong boasts a wide
    array of Michelin-starred restaurants offering diverse
   cuisines. Examples include Caprice (French), Lung King Hee (
   Cantonese), and Amber (Contemporary). * Private dining
   experiences: Many high-end restaurants offer private dining
   rooms for a more... | 1.0 |
| <Input (query): <Kauai> Output (outcome): <\# Kauai Itinerary:</pre>
   5 Days of Scenery \& Bro Time (Grad Student Budget) **Trip
   Overview: ** This itinerary focuses on experiencing Kauai's
   natural beauty while keeping costs down. It's designed for a
   group of bros looking for a relaxing and scenic getaway in
   late March. **Weather Considerations:** March is the tail end
    of the rainy season in Kauai. Expe... | 2.0 |
| <Input (query): <I will go to yangzhou city in China> Output (
   outcome): <Thank you for sharing your interests in history and
    food! Could you please let me know how many days you plan to
   stay in Yangzhou? This will help me create a well-structured
   itinerary for your trip.>> | 2.0 |
*Showing 3 of 9 total examples.*
```

Description: Feasibility_and_Realism_Qwen3-32B

Ensures the itinerary is realistic in terms of time, distance, and activity density.

TauBench — reward

Overall Kendall τ : 0.2469

Top 3 Metrics & Coefficients

Metric	Coefficient
Membership_Benefit_Application_Rubric	0.0803
Escalation_Appropriateness_Rubric	0.0599
Policy_Compliance_Qwen3-32B	0.0567

2916 2917 Description: Membership_Benefit_Application_Rubric 2918 2919 | Score | Description | 2920 |-----| 2921 | 1 | - **Score 1 (Fails to apply rules) **: 2922 - Incorrectly assigns free baggage allowances regardless of 2923 membership tier or cabin class. 2924 - Applies insurance benefits to users who do not meet eligibility criteria (e.g., no insurance, basic economy). 2925 - Offers compensation certificates to users who are not eligible 2926 (e.g., regular members without insurance). 2927 - Fails to enforce policy restrictions (e.g., allowing basic 2928 economy cancellations outside the 24-hour window without insurance). | 2929 | 2 | - **Score 2 (Major errors in application) **: 2930 - Applies baggage allowances inconsistently (e.g., correct for 2931 some tiers but not others). 2932 - Misapplies insurance eligibility (e.g., allows refunds for 2933 cancellations without valid reasons). - Offers compensation certificates in most cases but misses key 2934 eligibility criteria (e.g., ignores membership tier). 2935 - Occasionally transfers to human agents unnecessarily due to 2936 incorrect benefit application. | 2937 | 3 | - **Score 3 (Partial adherence with minor errors) **: 2938 - Correctly applies baggage allowances for most membership tiers but has occasional errors (e.g., miscalculates free bags for 2939 gold members). 2940 - Applies insurance eligibility in most cases but fails in edge 2941 cases (e.g., business class cancellations without checking 2942 insurance status). 2943 - Offers compensation certificates in most eligible scenarios but occasionally misses conditions (e.g., delayed flights without 2944 verifying membership). 2945 - Rarely transfers to human agents due to minor benefit 2946 application issues. | 2947 | 4 | - **Score 4 (High adherence with rare errors) **: - Correctly applies baggage allowances for all membership tiers 2948 and cabin classes in most cases. 2949 - Applies insurance eligibility accurately in nearly all 2950 scenarios. 2951 - Offers compensation certificates in all eligible cases but has 2952 one minor oversight (e.g., miscalculating certificate amounts 2953 for multi-passenger reservations). - Transfers to human agents only when necessary and for valid 2954 reasons. | 2955 | 5 | - **Score 5 (Perfect adherence) **: 2956 - Always assigns free baggage allowances correctly based on 2957 membership tier and cabin class. 2958 - Applies insurance eligibility and compensation rules flawlessly , adhering strictly to policy. 2959 - Never offers ineligible benefits (e.g., no certificates to 2960 regular members without insurance). 2961 - Transfers to human agents only when the request falls outside 2962 the scope of membership benefits. | 2963 2964

2970 2971 Description: Escalation Appropriateness Rubric 2972 2973 | Score | Description | 2974 |-----| 2975 | 1 | - **Fails to transfer** in all cases where policy limits 2976 are reached or exceptions are needed. 2977 **Incorrectly handles** requests that require human intervention (e.g., proceeds with booking/canceling flights 2978 outside policy). 2979 - **No adherence** to the rule of transferring for policy 2980 violations or exceptions. | 2981 | 2 | - **Transfers inconsistently** (e.g., transfers in some 2982 policy-violating cases but not others). **Fails to transfer** for critical exceptions (e.g., basic 2983 economy cancellations without insurance, destination changes). 2984 - **Attempts to resolve** issues beyond its scope (e.g., 2985 modifying flight destinations, waiving fees without human 2986 input). I 2987 | 3 | - **Transfers** in most policy-violating cases (e.g., denies basic economy cancellations and transfers to human agents). 2989 - **Partially handles exceptions** (e.g., transfers for compensation requests but not for all policy violations). 2991 **Some errors** in determining when to escalate (e.g., 2992 transfers unnecessarily for minor issues). | \mid 4 \mid - **Consistently transfers** when policy limits are reached 2993 (e.g., denies basic economy cancellations, blocks destination 2994 changes). 2995 - **Transfers for exceptions** (e.g., user insists on refunds for 2996 non-refundable tickets, requests compensation for delays). - **Minimal errors** in escalation decisions, with clear 2997 adherence to policy boundaries. | 2998 | 5 | - **Perfectly transfers** in all required cases (e.g., 2999 policy violations, exceptions, ambiguous requests). 3000 - **Never attempts to handle** requests outside its scope (e.g., 3001 denies basic economy cancellations, blocks invalid 3002 modifications). - **Proactively transfers** when user intent is unclear or 3003 requires human judgment (e.g., personal emergencies, 3004 compensation negotiations). | 3005 3006 3007 Description: Policy_Compliance_Qwen3-32B 3008

Adherence to airline rules (e.g., no basic economy cancellations without insurance or 24-hour window).

3009

3010