EvaluLLM: LLM Assisted Evaluation of Generative Outputs

Michael Desmond
Zahra Ashktorab
Qian Pan
James M. Johnson
Casey Dugan
mdesmond@us.ibm.com
IBM Research
Yorktown Heights, New York, USA

ABSTRACT

With the rapid improvement in large language model (LLM) capabilities, its becoming more difficult to measure the quality of outputs generated by natural language generation (NLG) systems. Conventional metrics such as BLEU and ROUGE are bound to reference data, and are generally unsuitable for tasks that require creative or diverse outputs. Human evaluation is an option, but manually evaluating generated text is difficult to do well, and expensive to scale and repeat as requirements and quality criteria change. Recent work has focused on the use of LLMs as customize-able NLG evaluators, and initial results are promising. In this demonstration we present EvaluLLM, an application designed to help practitioners setup, run and review evaluation over sets of NLG outputs, using an LLM as a custom evaluator. Evaluation is formulated as a series of choices between pairs of generated outputs conditioned on a user provided evaluation criteria. This approach simplifies the evaluation task and obviates the need for complex scoring algorithms. The system can be applied to general evaluation, human assisted evaluation, and model selection problems.

ACM Reference Format:
Michael Desmond, Zahra Ashktorab, Qian Pan, James M. Johnson, and Casey Dugan. 2024. EvaluLLM: LLM Assisted Evaluation of Generative Outputs. In *29th International Conference on Intelligent User Interfaces - Companion (IUI Companion '24), March 18–21, 2024, Greenville, SC, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3640544.3645216

1 LLMS AS EVALUATORS

LLMs adept at instruction following are capable of producing outputs that exceed the quality of human produced reference data [5]. As such, conventional reference-based metrics such as ROUGE [6] and BLEU [8] may not be reliable indicators of NLG output quality. Additionally, conventional metrics are not customize-able or adaptable to specific user needs or concerns. If we extrapolate the performance of current LLM output quality, its likely that future evaluation will need to measure adherence to complex values and preferences expressed by individuals and organizations, rather than relying on similarity to fixed reference examples. Human evaluation is a potential solution, however reviewing and scoring generated text is not a trivial task, and the approach does not scale well in an iterative development process, where data, models and evaluation criteria evolve over time.

Recent studies suggest that LLMs can be used as customize-able NLG output evaluators [7, 9]. This general approach can be applied to tasks that do not have human provided references, or are unsuitable for reference based evaluation in general. Wang et al. 2023 [9] evaluated ChatGPT as an aspect specific NLG evaluator in both reference-based and reference-free trials. They found that a ChatGPT evaluator has a high correlation with human preference in most cases, especially for creative NLG tasks, but evaluation is sensitive to prompt design and for different tasks and aspects the evaluation prompt needs careful consideration. G-Eval [7] applies LLMs using a chain-of-thought (CoT) prompting approach and a form-filling paradigm to assess the quality of NLG outputs. The framework is applied to two NLG tasks: text summarization and dialogue generation, and with GPT-4 as the evaluator model achieves a correlation of 0.514 with human judgments on a summarization task. The authors do however raise concern regarding potential for bias towards self generated outputs during evaluation.

When applying an LLM as an NLG evaluator, an important consideration is how to reliably generate a quality score or value. G-Eval [7] prompts an LLM to directly produce a score along a fixed scale, but granularity is coarse and LLMs are known to have issues with numbers [1]. GPTScore [2] measures NLG quality as the conditional probability of a generated output, conditioned on a task description, an aspect definition, and a task context. This approach expands scoring granularity and opens up the library of models that can serve as evaluators. However the approach relies on access to conditional token probabilities which may not be available from hosted or closed LLMs. An alternative approach to scoring, applied by AlpacaEval [4], is to measure the fraction of times a powerful LLM (e.g. GPT 4) prefers the outputs from a particular model over outputs from a reference model. The approach leads to an intuitive *win rate* metric, which can then be used to sort by comparative performance.

More recently, LLMs specifically finetuned for evaluation have begun to emerge. Prometheus [3], a 13B evaluator LLM, is trained to evaluate long-form generative outputs provided a custom score rubric. Experimental results indicate that Prometheus scores a Pearson correlation of 0.897 with human evaluators on 45 customized score rubrics.

IUI Companion '24, March 18–21, 2024, Greenville, SC, USA

ACM ISBN 979-8-4007-0509-0/24/03
https://doi.org/10.1145/3640544.3645216

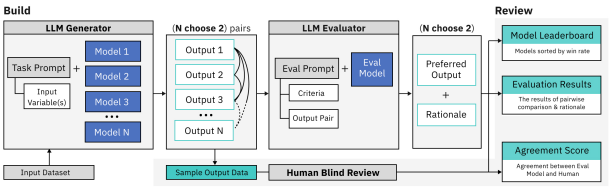

Figure 1: Overview of the NLG evaluation process. $\binom{n}{2}$ evaluations are applied to each input data instance, where n is the number of generated outputs to evaluate.

2 EvaluLLM

EvaluLLM is a web based NLG evaluator designed to determine and compare the performance of a set of LLMs on a user specified NLG task. The NLG task is defined by an input dataset D and a task prompt P. Variables exposed by the schema of D may be embedded into P as parameters using conventional curly bracket notation, alongside static task instructions. When executing the prompt over datums from D, variable references in the prompt are replaced with corresponding values from the datum. The NLG task is evaluated over a set of LLMs M, using an evaluator LLM E, and an evaluation criteria C. The criteria C is a user defined quality description expressed in natural language.

Evaluation is formulated as a set of choices between pairs of model outputs, similar in nature to AlpacaEval [4], except that generated model outputs are compared against each other, rather than to a fixed reference model output. Decomposing evaluation into a set of pairwise decisions reduces the complexity of the evaluation task and the size of the input context, at the cost of additional inference operations.

Each datum $d_i \in D$ from the dataset is applied to the task prompt P, and instance $P(d_i)$ is then inferenced on each LLM $m_i \in M$ resulting in a set of $|M|$ generated outputs O_i. Evaluation of generated outputs involves application of a custom evaluation prompt incorporating an *instruction* (P populated with variables from datum d_i), a pair of selected outputs $o_1, o_2 \in O_i$, and the evaluation criteria C, see figure 1. The evaluation prompt is engineered to induce a choice from the evaluator LLM conditioned on the evaluation criteria i.e. the prompt ends with the statement *Based on the evaluation criteria, the best output is.* The first non-whitespace character generated by the evaluator LLM indicates the preferred output (1 or 2). The evaluator is also prompted to produce an explanation which can be useful for human review purposes and debugging of evaluator reasoning.

Evaluation of each instance $P(d_i)$ consists of $\binom{n}{2}$ trials, where n is equal to the number of generated outputs. Local (per instance) and global (per model) scoring is calculated via win rate. The win rate represents the proportion of wins (the model output is preferred by the evaluator LLM) to the total number of evaluation trials that an output has participated in, and is a metric that is easily understood by end users. At the example level win rate is used to sort outputs for human review, and at the model level win rate is aggregated and used to sort models into an overall performance leaderboard.

3 INTERFACE DESIGN

The EvaluLLM interface is designed with two user experiences in mind: the build experience, focused on setting up and monitoring an LLM assisted evaluation, and the review experience, intended for exploring and evaluating results, see 1. The build experience comprises two configurable sections: the generator and the evaluator. The generator section allows the user to choose a dataset and to edit their task prompt. Data variables from the dataset schema can be embedded in the task prompt using standard curly bracket notation. The system also presents a selection of LLMs that the user can choose to evaluate. The evaluator section is where the user selects the LLM that will perform the automatic evaluation of the model outputs (from the generator), and enters the custom evaluation criteria. The evaluation criteria is a natural language description of quality by which model outputs should be assessed by the evaluator LLM.

Once an evaluation has completed running the user moves to the review experience. A *model leaderboard* displays a list of LLMs sorted by win rate aggregated over all evaluated data instances. To assist in assessing the reliability of the LLM evaluator preference, the user can choose to manually review and rate a sample of model outputs. These ratings, which are gathered in a blind manner to avoid bias, are then compared to LLM based evaluations to produce an *agreement score*. The agreement score indicates how well the user and the evaluator LLM agree on output quality, and can be helpful in understanding if the evaluation criteria is being interpreted and applied correctly. Finally the user can also browse a list of all evaluated instances and preference rationales.

4 PERSPECTIVES ON NLG EVALUATION

In addition to the design and development of EvaluLLM, we also carried out interviews with six individuals whose role is to assist clients at a large multinational technology company to select the most suitable model based on user-defined criteria and requirements. These interviews delved into the nuances of the NLG evaluation and model selection processes, which typically involve direct human evaluation, but can inform the design of LLM based evaluation approaches. Table 1 highlights themes that emerged from these interviews that we plan to further study, and potentially integrate into EvaluLLM in future iterations.

Participants expressed preferences regarding scoring systems: there was a notable demand for the option to assign negative scores to particularly poor outputs, and for defining clear extremes on evaluation dimensions, such as distinguishing between 'factual' and 'non-factual' outputs. The need to prioritize certain evaluation dimensions over others was emphasized. For instance, one user, highlighted a greater concern for the "usefulness" of an output over its "faithfulness" with respect to a given source material, suggesting a need for flexibility to accommodate complex scoring and user-defined evaluation criteria. Another theme that emerged during the interviews was the diversity in how practitioners approached the evaluation process. Some interviewees favored involving a larger number of human evaluators with minimal guidance, while others preferred a smaller, more guided approach. It was also clear that scoring rubrics evolved and expanded over the course of the evaluation effort, indicating the need for an iterative system.

Theme	Example
Scoring	
Ability to assign negative scores for egregious outputs	-2 → Answer completely incoherent. Does not answer the question at all or answers the wrong question (Usefulness)
Ability to define opposite ends of a evaluation dimension	Factual (1) : answer is fully grounded in source content or "overflows" the current prompt Factual (-1): Answer not fully grounded in source content and does not overflow the prompt
Ability to rank the importance of evaluation dimensions	Usefulness is more important than Faithfulness
Workflow	
Collaborative and Iterative	Rating started out as binary, then expanded to multiple dimensions
Divide and Conquer	Evaluators rate the same content to establish baseline and shared understanding of evaluation criteria dimensions. This process is followed by a discussion and agreement on definition of dimensions. Remaining data then divided among evaluators.
Prioritize more evaluators + very little guidance over fewer evaluators + too much guidance	Customers don't have guidance that evaluators have and they have gut/immediate reactions – evaluators should be given few guidelines

Table 1: Themes & summary of interview findings on NLG evaluation

5 SIGNIFICANCE & LIMITATIONS

Evaluation of high quality NLG systems is challenging, and automated evaluation using LLMs is a promising solution. LLMs offer levels of customize-ability and repeatability that are difficult to achieve with conventional metrics or human evaluation processes. EvaluLLM allows practitioners to design and run custom natural language defined evaluations, and explore and validate results. While the approach and tool are preliminary, going forward we believe that LLM based evaluation will become a standard practice, and adequate tooling in this space will be important.

There are however some caveats and open issues to address. In general LLMs are known to suffer from hallucinations, fail to reason adequately, and are sensitive to prompt design [1, 5]. Existing work in LLM evaluation has identified issues with bias [4, 7] and prompt sensitivity [9]. Tools that apply LLM evaluation in a robust and transparent manner are necessary to help practitioners to understand the risks of LLM evaluation and set expectations appropriately. Integrating human oversight into the process is also an important consideration. Gathering and exposing metrics such as agreement and correlation with human judgements can help to ground and verify the results of automatic evaluations. Interactive definition and validation of evaluation criteria is another important aspect to make LLM evaluation scale. In general we believe that evaluation via LLM is not a replacement for human evaluation, but can serve to make the NLG evaluation process more robust and less laborious.

REFERENCES

[1] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109* (2023).

[2] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166* (2023).

[3] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491* (2023).

[4] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

[5] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).

[6] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[7] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment, May 2023. *arXiv preprint arXiv:2303.16634* (2023).

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[9] Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048* (2023).