A Field Guide to Automatic Evaluation of LLM-Generated Summaries

Tempest A. van Schaik
Microsoft
Redmond Washington USA
tempest.van@microsoft.com

Brittany Pugh
Microsoft
Redmond Washington USA
brittanypugh@microsoft.com

ABSTRACT

Large Language models (LLMs) are rapidly being adopted for tasks such as text summarization, in a wide range of industries. This has driven the need for scalable, automatic, reliable, and cost-effective methods to evaluate the quality of LLM-generated text. What is meant by evaluating an LLM is not yet well defined and there are widely different expectations about what kind of information evaluation will produce. Evaluation methods that were developed for traditional Natural Language Processing (NLP) tasks (before the rise of LLMs) remain applicable but are not sufficient for capturing high-level semantic qualities of summaries. Emerging evaluation methods that use LLMs to evaluate LLM-output, appear to be powerful but lacking in reliability. New elements of LLM generated text that were not an element of previous NLP tasks, such as the artifacts of hallucination, need to be considered. We outline the different types of LLM evaluation currently used in the literature but focus on offline, system-level evaluation of the text generated by LLMs. Evaluating LLM-generated summaries is a complex and fast-evolving area, and we propose strategies for applying evaluation methods to avoid common pitfalls. Despite having promising strategies for evaluating LLM summaries, we highlight some open challenges that remain.

CCS CONCEPTS

• Information Systems → Information retrieval → Evaluation of retrieval results

KEYWORDS

Evaluation metrics; LLMs; summarization; offline evaluation

ACM Reference format:
Tempest A. van Schaik and Brittany Pugh. 2024. A Field Guide to Automatic Evaluation of LLM-Generated Summaries. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'24). July 14-18, 2024, Washington, D.C., USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3626772.3661346

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07. https://doi.org/10.1145/3626772.3661346

1 INTRODUCTION

Large Language Models (LLMs), such as the GPT-3, show surprising, emergent abilities [1] compared to smaller pre-trained models (PTMs) like BERT [2]. ChatGPT is an application of LLMs that caused much excitement in the AI community and beyond [1]. The power and ease-of-use of LLMs has led to a surge in LLM-powered applications with developers harnessing Artificial Intelligence (AI) to address real world challenges. An emerging application of LLMs in industry is to summarize text, such as producing summaries of product reviews, or technical manuals.

LLMs have known issues of hallucination, knowledge recency, reasoning inconsistency, difficulty in computational reasoning and more [1], which result in unpredictable errors. It is important to evaluate how well LLM applications perform before they are released [3], and when running in production [3]. Lack of proper evaluation can lead to undetected errors that can have a range of harms, from brand damage and revenue loss to consequential impact on life opportunities like access to healthcare or employment.

For software developers who are new to AI, the landscape of evaluation can be difficult to get started in. For data scientists experienced in NLP and evaluation, it can feel challenging to keep-up with advancements in LLMs, let alone how best to evaluate them. We present a guide to LLM evaluation that is geared towards the common situations that AI practitioners (software developers and data scientists) face when building LLM applications in industry.

We begin by surveying what is meant by LLM evaluation in section 2, and what evaluation methods exist in section 3. We present recommendations for best practices in section 4 and conclude with open issues in section 5. Although we focus on summarization, much of what we discuss is relevant to additional language tasks.

2 WHAT IT MEANS TO EVALUATE AN LLM

What is meant by evaluation of an LLM varies between different AI communities. We present these different definitions to help practitioners navigate the landscape of LLM evaluation and find the most appropriate definition for them.

2.1 Security and Responsible AI. It is important to evaluate how well LLM systems align with social norms, values, and regulations such as fairness, privacy and copyright [4]. They should also be evaluated for robustness against producing harmful content and jailbreaking [5].

2.2 Computing Performance. LLMs can be costly and have high latency [6] so they should be evaluated in terms of cost, CPU and GPU usage, latency and memory.

2.3 Retrieval vs Generator Evaluation. Retrieval-Augmented Generation (RAG) is the process of retrieving relevant data from outside the pre-trained model to enhance the input to improve the generated output [7]. We recommend breaking down the evaluation of RAGs into three parts: (i) the information retrieval part of such a system (e.g. with well-established search metrics like precision, recall, Discounted Cumulative Gain [8]), since the generator can only perform as well as the context that it is given; (ii) the generative AI component; and (iii) the entire RAG system end-to-end to see how well it meets end-user needs.

2.4 Offline vs Online Evaluation. Offline evaluation involves developing the system with a batch of test data that is often human-annotated ground truth. It guides practitioners while they build the system and helps them decide when the system performs well enough to release. Online evaluation monitors performance with live user data. In some applications like e-commerce, it provides an opportunity to evaluate the system based on user-interaction metrics like click-through rate during AB testing.

2.5 System Evaluation vs Model Evaluation. Model evaluation is often performed with benchmarks like HellaSwag, TruthfulQA and MMLU [9] Benchmarks are used to compare competing LLMs, using a fixed, standard dataset, and fixed, standard language tasks (like summarization). However, in many industry scenarios, the model is fixed. Practitioners need to evaluate how well their model performs on specific industry data and tasks [10]. As an example, with prompt engineering, a system evaluation would keep the LLM constant but change the prompts.

With a focus on offline, system-level evaluation of generative AI text, we outline methods for evaluating the quality of summaries.

3 EVALUATION METHODS

Evaluation methods measure how well our system is performing. Manual evaluation (human review) of each summary would be time-consuming, costly and would not be scalable, so it is usually complemented by automatic evaluation. Many automatic evaluation methods attempt to measure the same qualities of a summary that human evaluators would consider. Those qualities include fluency [11], coherence [10], [11], [12], relevance [11], factual consistency [11], and fairness [13]. Similarity in content or style to a reference text can also be an important quality of generated text. In the next sections, we will examine reference-based, reference-free (context-based), and LLM-based metrics.

3.1 Reference-based Metrics

Reference-based metrics are used to compare generated text to a reference: the human annotated ground truth text. Many of these metrics were developed for traditional NLP tasks before LLMs were developed but remain applicable to LLM summarization.

3.1.1 N-gram based metrics. Metrics **BLEU (Bilingual Evaluation Understudy)** [14], **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)** [15], and **JS divergence (JS2)** [16] are overlap-based metrics that measure the similarity of the output text and the reference text using n-grams.

3.1.2 Embedding-based metrics. **BERTScore** [17], **MoverScore** [18], and **Sentence Mover Similarity (SMS)** [19] metrics all rely on contextualized embeddings to measure the similarity between two texts.

While these metrics are relatively simple, fast, and inexpensive compared to LLM-based metrics, studies have shown that they can have poor correlation with human evaluators, lack of interpretability, inherent bias, poor adaptability to a wider variety of tasks and inability to capture subtle nuances in language [19].

3.2 Reference-free Metrics

Reference-free (context-based) metrics produce a score for the generated text and do not rely on ground truth. Evaluation is based on the context or source document. Many of these metrics were developed in response to the challenge of creating ground truth data. These methods tend to be newer than reference-based techniques, reflecting the growing demand for scalable text evaluation as PTMs became increasingly powerful.

3.2.1 Quality-based metrics. Metrics **SUPERT [21]** and **BLANC** [22] focus on the quality of the content of the generated summary and use BERT to generate a pseudo-reference. **ROUGE-C** is a modification of ROUGE without the need for references and uses the source text as the context for comparison [23].

3.2.2 Entailment-based metrics. Entailment-based metrics are based on the Natural Language Inference (NLI) task, where for a given text (premise), it determines whether the output text (hypothesis) entails, contradicts or undermines the premise [24]. The **SummaC (Summary Consistency)** benchmark [24], **FactCC** [25], and **DAE (Dependency Arc Entailment)** [26] metrics serve as an approach to detect factual inconsistencies with the source text. Entailment-based metrics are designed as a classification task with labels “consistent” or “inconsistent”.

3.2.3 Factuality, QA and QG-based metrics. Factuality-based metrics like **SRLScore (Semantic Role Labeling**) [27] and **QAFactEval** [27] evaluate whether generated text contains incorrect information that does not hold true to the source text [29]. QA-based, like **QuestEval** [30], and QG-based metrics are used as another approach to measure factual consistency and relevance [30] [31].

Reference-free metrics have shown improved correlations to human evaluators compared to reference-based metrics, but there are limitations to using them as the single measure of progress on a task [32], including bias towards their underlying models' outputs and bias against higher-quality text [32].

3.3 LLM-based Evaluators

LLM's remarkable abilities have led to their emerging use as not only summarizers of text, but also evaluators of summarized text. [33], [34], [35] evaluators offer scalability and explainability.

3.3.1 Prompt-based evaluators. LLM-based evaluators prompt an LLM to be the judge of some text. The judgement can be based on (i) the summary alone (reference-free), where the LLM is judging qualities like fluency, and coherence; (ii) the summary, the original text, and potentially a topic for summarization (reference-free), where the LLM is judging qualities like consistency, and relevancy (iii) a comparison between the summary and the ground truth summary (reference-based), where the LLM is judging quality, and similarity. Some frameworks for these evaluation prompts include **Reason-then-Score (RTS)** [36], **Multiple Choice Question Scoring (MCQ)** [36], **Head-to-head scoring (H2H)** [36], and **G-Eval** [37].

LLM-evaluation is an emerging area of research and has not yet been systematically studied. Already, researchers have identified issues with reliability in LLM evaluators [3] such as **positional bias** [38], [39], **verbosity bias** [40], **self-enhancement bias** [40], and **limited mathematical and reasoning capabilities** [40]. Strategies that have been proposed to mitigate positional bias include Multiple Evidence Calibration (MEC), Balanced Position Calibration (BPC), and Human In The Loop Calibration (HITLC) [38].

3.3.2 LLM embedding-base metrics. Recently, the embedding models from LLMs, such as GPT3's text-embedding-ada-002 [1] has also been used for embedding-based metrics (see section 3.1.2).

4 BEST PRACTICES TO AVOID EVALUATION PITFALLS

Given the extensive menu of evaluation metrics that are available for LLM-generated summaries, we provide some best practices for evaluation that could help to avoid common pitfalls.

4.1 Suite of metrics. Rather than searching for "one metric to rule them all", we recommend developing a suite of metrics. We make the analogy with software testing, where unit tests, smoke tests and functional tests all work together to test the software at different levels. The suite approach acknowledges that each metric has its strengths and weaknesses. For example, ROUGE benefits from being simple and deterministic but does not capture semantic meaning. An LLM-based evaluator benefits from capturing nuanced qualities of a summary but is not always reliable.

4.2 Standard and custom metrics. Several metrics have emerged as the standard for evaluating LLM text (such as factual consistency). While these are important basic qualities of text, there are often additional, unique qualities of text for a specific use-case that will require developing custom metrics (see section 4.6.3). An example would be in summarizing legal documents which have a distinctive writing style.

Even a standard measure like consistency may need custom implementation. For example, a summary of electronic product specification contains many numerical values, such as "16 GB". To detect a factual inconsistency such as "8 GB", we used a simple custom consistency metric to extract these facts from the summary and ground truth using a regular expression search, and measured their overlap with F1 score. Consistency metrics based on PTMs failed to distinguish the semantic difference between "8 GB" and "16 GB". This illustrates why custom metrics are important for each application which has unique data qualities.

4.3 LLM and non-LLM metrics. Given the limitations of LLM evaluators, we recommend using them along with evaluators based on smaller PTMs, as well as non-model metrics (like ROUGE). These different metrics can corroborate each other. Discrepancies between these metrics can detect changes in performance that may not be apparent from each metric on its own. For example, we have observed that even when LLM-based metrics saturate (scoring almost every summary as a perfect 5/5), non-LLM based metrics can still detect differences in the summaries' qualities. Cost is also a consideration, and it may not be economical to use the most powerful LLM to routinely evaluate a large dataset.

4.4 Validate the evaluators. Non-standard (custom) metrics, especially custom LLM-based evaluators, should be validated to see how they perform. One way to do this is to calibrate them against human-evaluated summaries, that are balanced and ranging from "good" to "bad". Another best practice is to determine the stability of each metric by reporting its average over several runs of the same inputs.

4.5 Visualize and analyze metrics. Simply calculating metrics is not sufficient, they need to be interpreted to make them actionable. Data visualization is far more effective than simply viewing data in tabular format. For example, using boxplots to visualize metrics distribution, skewness, and outliers, for the whole dataset and within certain sub-categories (such as the search query in a RAG system).

4.6 Involve the experts. Sufficient time and resources need to be budgeted to develop a high quality LLM solution which involves domain experts (who are often the end-users) in three areas:

1) Annotation: Even when relying on reference-free evaluation, some annotated ground truth data is always necessary to calibrate the metrics and to develop the LLM system with.

2) Evaluation: The most important judge of the quality of a summary is the domain expert. This is especially important when the text is from a technical field (such as summarizing clinical research or airplane manuals), where the AI practitioner may not have the domain expertise to judge or even understand the summary. Domain experts may provide feedback that is qualitative (describing what they like about the summary), or quantitative (a Likert score for each summary, or thumbs up/down).

3) Metric Design: Domain experts must define what unique qualities are important to them and inform the design of metrics. In one scenario, we observed that domain experts evaluate an AI summary by seeing if it contains the information in each

sentence from the ground truth summary. In response, we designed a custom metric to perform sentence-by-sentence matching. We considered a sentence in the ground truth summary to be "found" in the AI summary if the cosine similarities between a domain-relevant PTM (PubMedBERT [41]) embedding of those sentences was above a certain threshold. The experts highlighted that editing-down excessive information saved more time than writing missing content, so the metric (shown in figure 1) measured sentence recall.

4.7 Data-driven prompt engineering. Having metrics allows the practitioner to form a hypothesis, test it, and observe the metrics to see if the experiment was successful. For example, a practitioner might write a new LLM summarization prompt and observe the effect on the summaries. They may solicit feedback from experts, and/or get immediate feedback from automatically calculated metrics and iterate on the prompt. Note that it is likely that the metrics that are chosen at the start of the project will evolve as practitioners gain a deeper understanding of the goals of summarization.

4.8 Tracking metrics over time. If metrics are introduced early in development, the metric values can be tracked for this corpus of summaries over time. Having metrics from the start enables the practitioner to establish baseline performance for a simple, single prompt, before progressing to more complex prompting strategies, and measuring what value this complexity adds. Figure 1 shows the average values of various metrics over multiple iterations of prompt engineering.

4.9 Appropriate metric interpretation. Even standard, established metrics like F1 score need careful custom interpretation for each application. A heuristic such as, F1 score should be "high" or close to 1 for a "good" summary, can be misleading. For ROUGE, F1 score comparing an abstractive summary and the original text, the summary may be good and have few overlapping n-grams (low F1).

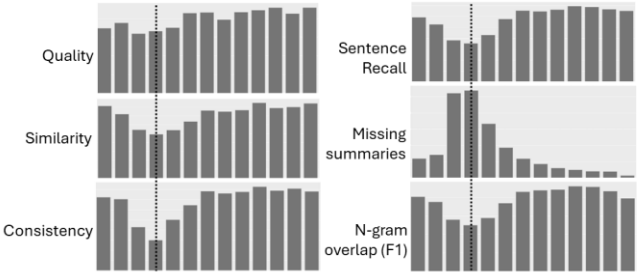

Figure 1: Six metrics that track performance of an LLM summarizer during prompt engineering: Quality, Similarity and Factualness (GPT-4 based); sentence recall (BERT-based); missing summaries; and F1 score for n-gram overlap. Y-axis is the average value for that metric across the corpus of 100 summaries. X-axis is the prompt variant number, with prompt 1 being the first (leftmost) and prompt 13 being the last (rightmost). Dashed line shows an unsuccessful prompt experiment (prompt 4), after which summaries began to improve.

5 OPEN CHALLENGES

5.1 Cold start problem. LLMs are so powerful that they enable entirely new services to be built, rather than simply enhancing existing solutions. This leads to the cold-start problem where little, or no data exists to evaluate the new solution. One approach is to synthesize data, though this should be used cautiously to enhance the productivity of human annotators rather than replace them [42]. Naively synthesized data may not represent the real-world distribution of underlying data [43]. Another approach is to explore re-purposing an existing dataset. For example, historic search logs can tell us what information an end-user found valuable.

5.2 Subjectivity in evaluating and annotating text. What makes a summary good is subjective and experts sometimes disagree. It is challenging for experts to consistently quantify how good a summary is. Annotators may also disagree on what information should be annotated, for example sufficient or exhaustive information [44]. Inter-annotator agreement varies by task and industry but was recently found in LLM-studies [40] to be 80%. Therefore, we cannot expect perfect agreement between automatic metrics and human evaluators. Subjectivity remains a challenge when evaluating text.

5.3 Challenge of good vs excellent. While we are confident that the approaches that we have set forth in this paper can discern good summaries from bad summaries, the challenge remains in discerning good summaries from excellent summaries. LLM's remarkable performance in text summarization has highlighted this challenge. This may be unsurprising, because even if automatic metrics can match human evaluators, the expertise required to evaluate summaries in some fields is very rare. Two domain-specific summaries with subtle quality differences (good and excellent) would have almost identical LLM embedding vectors. This can potentially be improved by fine-tuning with domain-specific text, though this is not always practical.

6 CONCLUSION

There is a clear need to evaluate how well LLMs perform, and a wide variety of ways to approach this. It can be challenging to decide what element of the LLM to evaluate, which established techniques are still applicable in the era of LLMs, and which emerging evaluation techniques to adopt. We outline what is meant by LLM evaluation in different AI communities, and we focus on system-level, offline evaluation of LLM-generated summaries. We survey established NLP metrics which are reference-based, more recent PTM-based reference-free metrics, and emerging LLM prompt-based metrics. We explore some best practices for selecting, combining, interpreting, and acting on metrics; and for involving human experts in evaluation. Even when using these best-practices, we find that several open-challenges still remain. Evaluating LLMs is far from being solved. We should not underestimate the complexity and investment needed to properly evaluate LLMs, which is essential for building LLM applications.

AUTHOR BIOGRAPHIES

Tempest van Schaik is a Principal Data Scientist in Microsoft's Industry Solutions Engineering team. She helps to build new products and services that use AI for Microsoft's health and life science enterprise customers. She received her PhD in Bioengineering from Imperial College London in 2014. Her research publications have been in the areas of medical devices, clinical research, and responsible AI.

Brittany Pugh is a Senior Data Scientist in Microsoft's Industry Solutions Engineering team. She helps customers solve their most challenging problems utilizing AI products and services for Microsoft's health and life science enterprise customers. Brittany received a master's degree in systems engineering from George Washington University after receiving a bachelor's in mathematics from the University of Virginia.

REFERENCES

[1] W. X. Zhao *et al.*, “A Survey of Large Language Models,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.18223

[2] J. Wei *et al.*, “Emergent Abilities of Large Language Models”, Accessed: Feb. 14, 2024. [Online]. Available: https://openreview.net/forum?id=yzkSU5zdwD

[3] X. Wang, R. Gao, A. Jain, G. Edge, and S. Ahuja, “How Well do Offline Metrics Predict Online Performance of Prod-uct Ranking Models,” no. 23, 2023, doi: 10.1145/3539618.3591865.

[4] Y. Liu *et al.*, “Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment,” Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.05374

[5] Y. Dong *et al.*, “Building Guardrails for Large Language Models,” Feb. 2024, [Online]. Available: http://arxiv.org/abs/2402.01822

[6] S. Minaee *et al.*, “Large Language Models: A Survey”.

[7] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”, Accessed: Feb. 11, 2024. [Online]. Available: https://github.com/huggingface/transformers/blob/master/

[8] O. Jeunen, I. Potapov, and A. Ustimenko, “On (Normalised) Discounted Cumulative Gain as an Off-Policy Evaluation Metric for Top-í µí±□ Recommendation,” *Proceedings of ACM Conference (Conference'17)*, vol. 1.

[9] P. Liang *et al.*, “Holistic Evaluation of Language Models,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2211.09110

[10] “LLM Evaluation: Everything You Need To Run, Benchmark Evals.” Accessed: Feb. 11, 2024. [Online]. Available: https://arize.com/blog-course/llm-evaluation-the-definitive-guide/#large-language-model-model-eval

[11] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, “Neural Text Summarization: A Critical Evaluation,” Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.08960

[12] R. Fabbri *et al.*, “SummEval: Re-evaluating Summarization Evaluation”, doi: 10.1162/tacl.

[13] O. Gallegos *et al.*, “Bias and Fairness in Large Language Models: A Survey,” Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.00770

[14] Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation.”

[15] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries.”

[16] M. Bhandari, P. Gour, A. Ashfaq, P. Liu, and G. Neubig, “Re-evaluating Evaluation in Text Summarization,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.07100

[17] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTSCORE: EVALUATING TEXT GENERATION WITH BERT”, Accessed: Feb. 11, 2024. [Online]. Available: https://github.com/Tiiiger/bert_score.

[18] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, “MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance”, Accessed: Feb. 11, 2024. [Online]. Available: http://tiny.cc/vsqtbz

[19] E. Clark, A. Celikyilmaz, N. A. Smith, and P. G. Allen, “Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts.” [Online]. Available: https://github.com/src-d/wmd-relax

[20] B. Sai and M. M. Khapra, “A Survey of Evaluation Metrics Used for NLG Systems,” 2020, doi: 10.1145/0000001.0000001.

[21] Y. Gao, W. Zhao, and S. Eger, “SUPERT: Towards New Frontiers in Unsupervised Evaluation Metrics for Multi-Document Summarization.” [Online]. Available: https://tac.nist.gov/

[22] O. Vasilyev, V. Dharnidharka, and J. Bohannon, “Fill in the BLANC: Human-free quality estimation of document summaries.”

[23] T. He *et al.*, “ROUGE-C: A Fully Automated Evaluation Method for Multi-document Summarization.” [Online]. Available: http://www.isi.edu/-cyl/SEE

[24] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst, “SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization,” Nov. 2021, [Online]. Available: http://arxiv.org/abs/2111.09525

[25] W. Kryściński, B. McCann, C. Xiong, and R. Socher, “Evaluating the Factual Consistency of Abstractive Text Summarization,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.12840

[26] T. Goyal and G. Durrett, “Findings of the Association for Computational Linguistics Evaluating Factuality in Generation with Dependency-level Entailment.” [Online]. Available: https://github.com/

[27] Fan, D. Aumiller, and M. Gertz, “Evaluating Factual Consistency of Texts with Semantic Role Labeling,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.13309

[28] R. Fabbri, C.-S. Wu, W. Liu, and C. Xiong, “QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization,” Dec. 2021, [Online]. Available: http://arxiv.org/abs/2112.08542

[29] Pagnoni, V. Balachandran, and Y. Tsvetkov, “Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.13346

[30] T. Scialom *et al.*, “QuestEval: Summarization Asks for Fact-based Evaluation,” Mar. 2021, [Online]. Available: http://arxiv.org/abs/2103.12693

[31] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano, “Answers Unite! Unsupervised Metrics for Reinforced Summarization Models,” Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.01610

[32] D. Deutsch, R. Dror, and D. Roth, “On the Limitations of Reference-Free Evaluations of Generated Text,” Oct. 2022, [Online]. Available: http://arxiv.org/abs/2210.12563

[33] Peng, C. Li, P. He, M. Galley, and J. Gao, “INSTRUCTION TUNING WITH GPT-4”, Accessed: Feb. 19, 2024. [Online]. Available: https://instruction-tuning-with-gpt-4.github.io/

[34] Z. Sun *et al.*, “Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision”, Accessed: Feb. 19, 2024. [Online]. Available: https://github.com/IBM/Dromedary

[35] Zhou *et al.*, “LIMA: Less Is More for Alignment”.

[36] Shen, L. Cheng, X.-P. Nguyen, Y. You, and L. Bing, “Large Language Models are Not Yet Human-Level Evaluators for Abstractive Summarization,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.13091

[37] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.16634

[38] P. Wang *et al.*, “Large Language Models are not Fair Evaluators”, Accessed: Feb. 11, 2024. [Online]. Available: https://github.com/i-Eval/

[39] C.-H. Chiang and H.-Y. Lee, “Can Large Language Models Be an Alternative to Human Evaluation?,” Long Papers.

[40] Zheng *et al.*, “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena”.

[41] Y. U. Gu *et al.*, “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing,” no. 1, p. 24, 2021, doi: 10.1145/3458754.

[42] V. Veselovsky, M. H. Ribeiro, A. Arora, M. Josifoski, A. Anderson, and R. West, “Generating Faithful Synthetic Data with Large Language Models: A Case Study in Computational Social Science,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.15041

[43] Josifoski, M. Sakota, M. Peyrard, and R. West, “Exploiting Asymmetry for Synthetic Training Data Generation: SynthIE and the Case of Information Extraction,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.04132

[44] H. Cheng *et al.*, “MDACE: MIMIC Documents Annotated with Code Evidence”, Accessed: Feb. 23, 2024. [Online]. Available: https://github.com/3mcloud/MDACE/.