EvalSense: A Framework for Domain-Specific LLM (Meta-)Evaluation

Adam Dejl
Imperial College London*
Department of Computing
ad5518@ic.ac.uk

Jonathan Pearson
NHS England
Transformation Directorate
jonathanpearson@nhs.net

Abstract

Robust and comprehensive evaluation of large language models (LLMs) is essential for identifying effective LLM system configurations and mitigating risks associated with deploying LLMs in sensitive domains. However, traditional statistical metrics are poorly suited to open-ended generation tasks, leading to growing reliance on LLM-based evaluation methods. These methods, while often more flexible, introduce additional complexity: they depend on carefully chosen models, prompts, parameters, and evaluation strategies, making the evaluation process prone to misconfiguration and bias. In this work, we present EvalSense, a flexible, extensible framework for constructing domain-specific evaluation suites for LLMs. EvalSense provides out-of-the-box support for a broad range of model providers and evaluation strategies, and assists users in selecting and deploying suitable evaluation methods for their specific use-cases. This is achieved through two unique components: (1) an *interactive guide* aiding users in evaluation method selection and (2) *automated meta-evaluation tools* that assess the reliability of different evaluation approaches using perturbed data. We demonstrate the effectiveness of EvalSense in a case study involving the generation of clinical notes from unstructured doctor-patient dialogues, using a popular open dataset. All code, documentation, and assets associated with EvalSense are open-source and publicly available at https://github.com/nhsengland/evalsense.

1 Introduction

Backed by training on unprecedentedly large quantities of data, large language models (LLMs) have radically advanced the field of machine learning and demonstrated a wide range of impressive capabilities across diverse domains (Bubeck et al., 2023; Van Veen et al., 2024; Luo et al., 2025; McDuff et al., 2025). While these results suggest that LLMs have the potential to deliver substantial benefits, their use also entails significant risks, including hallucinations (Huang et al., 2025), omissions of crucial information (Busch et al., 2025), unintended disclosure of sensitive personal data (Das et al., 2025), and vulnerability to harmful instructions (Das et al., 2025). Rigorous evaluation of LLMs has been proposed as a key strategy for mitigating these risks and ensuring that LLM-based systems perform reliably on their assigned tasks (WHO, 2023; Ong et al., 2024).

However, reliable evaluation of open-ended texts produced by LLMs remains challenging as a result of the unstructured and complex nature of these texts. Due to the inadequacy of standard statistical metrics, the community has increasingly adopted LLM-as-a-judge approaches (Liu et al., 2023; Fu et al., 2024; Kim et al., 2024a,b), which use LLMs themselves to assess model outputs. These methods tend to be more effective at capturing content-related nuances and generally achieve higher correlations with human judgements (Zheng et al., 2023). Yet, the reliability of LLMs as evaluators may vary depending on the considered task, LLM judge and the used evaluation strategy (Murugadoss et al., 2025; Tan et al., 2025; Han et al., 2025). This motivates the need to carefully choose the evaluation approach suitable for the specific domain and to rigorously *meta-evaluate* its effectiveness (i.e., to evaluate the evaluator), steps that are often neglected in the existing evaluation pipelines.

Several open-source toolkits and frameworks for evaluating LLMs have been introduced, such as `lm-evaluation-harness` (Gao et al., 2024), OpenCompass (Contributors, 2023), LightEval (Habib et al., 2023), Inspect (UK AI Security Institute, 2024) and `Unitxt` (Bandel et al., 2024). However, while these tools provide useful infrastructure for running standardised benchmarks or implementing specific evaluation workflows, they typically do not aid users in selecting appropriate

*Work done while at NHS England.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 480–491

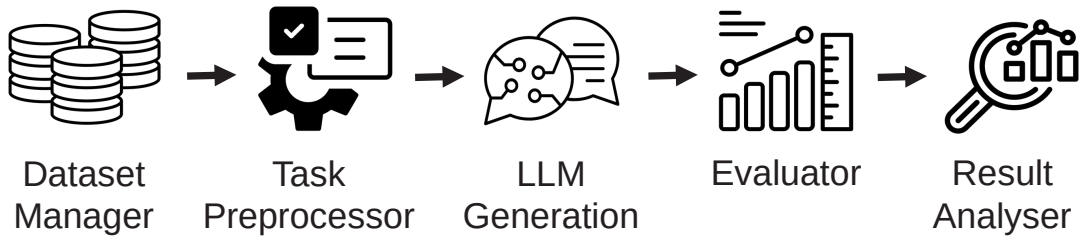


Figure 1: Overview of the LLM evaluation pipeline implemented in EvalSense[1]. After data loading and task-specific preprocessing, model outputs are generated and scored using different evaluators. Result analysers summarise outcomes across experiments, identify higher-level patterns, and support meta-evaluation.

methods or in quantitatively measuring the effectiveness of these methods for a specific domain and task through meta-evaluation.

In response to these gaps, we introduce EvalSense, a highly flexible software framework that enables users to systematically evaluate LLMs on custom datasets. EvalSense offers two key features to help users navigate the spectrum of available evaluation methods:

1. It includes an *interactive evaluation guide*[2], which prompts users to specify their tasks along with the associated risks and requirements, and then suggests appropriate evaluation strategies. After a subset of methods is selected, the guide generates a coverage report indicating whether the chosen methods comprehensively cover the specified risks and requirements (Figure 2a).

2. EvalSense incorporates *automated meta-evaluation tools* that leverage controlled perturbations to validate evaluator reliability on the user's own dataset. These tools systematically degrade specific aspects of the output texts, verifying the degree to which these changes are reflected in the scores produced by the different evaluation techniques.

In addition to these features, EvalSense also supports systematic experimentation, a broad range of local and API model providers, configuring evaluations through a graphical user interface (Figure 2b), high-level result analysis and complex generation workflows.

[1]Icons by Noun Project, authors Srinivas Agra, Iconiqu, Gonza Monta, Keyy Creative and suhaiba, CC BY 3.0.

[2]Available on the EvalSense website at https://nhsengland.github.io/evalsense/.

To demonstrate and assess the capabilities of EvalSense, we apply it to a realistic evaluation task using ACI-Bench (Yim et al., 2023), which involves generation of structured clinical notes from doctor-patient dialogues. Using EvalSense's meta-evaluation tools, we demonstrate a non-trivial disparity in the quality of the scores produced by different evaluation methods. This demonstrates the importance of careful method selection and configuration, a process our framework is specifically designed to support.

Overall, we hope that EvalSense contributes to advancing best practices in LLM evaluation by systemizing the process of choosing between different evaluation strategies, both through the interactive guidance provided by the EvalSense guide and the quantitative meta-evaluation supported by the associated open-source library.

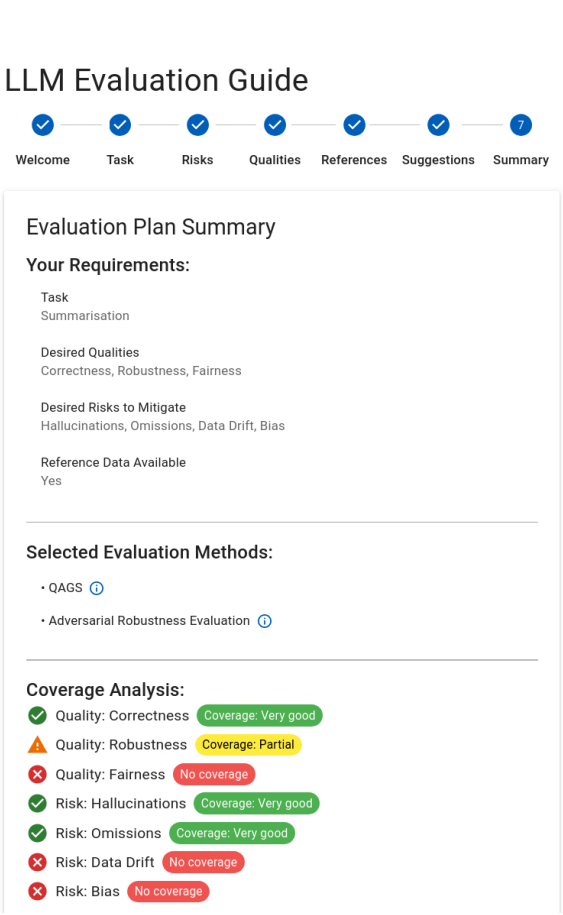
2 Background and Related Work

2.1 Evaluation Metrics

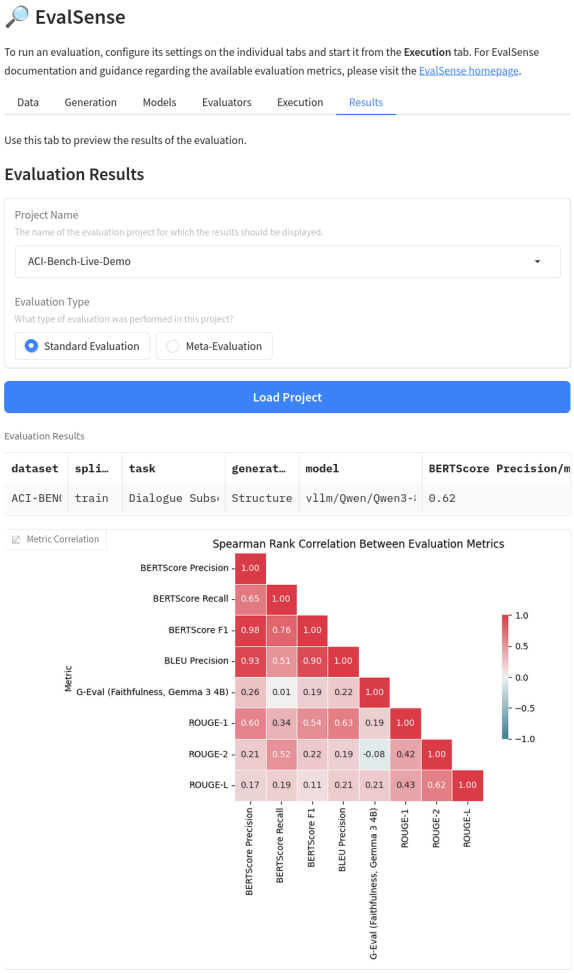
The growing use of machine learning models for text generation has led to the development of a wide range of evaluation techniques. Broadly, these can be categorised into three groups: traditional statistical metrics, LLM-as-a-judge methods and hybrid approaches.

Statistical metrics rely on direct, deterministic comparison of text units extracted from the evaluated text to the ground-truth reference. While still in use, these approaches are often overly simple to reliably assess the quality of open-ended texts. Examples of such metrics include the BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) scores.

LLM-as-a-judge methods leverage the general capabilities of LLMs to assess generated texts, mitigating many of the drawbacks associated with sta-



(a) LLM Evaluation Guide



(b) EvalSense user interface

Figure 2: **(a)** EvalSense's LLM Evaluation Guide assists users in selecting suitable evaluation methods based on task-specific risks and requirements. The final evaluation plan summary highlights any risks and requirements not fully covered by the selected methods. The guide is available at `https://nhsengland.github.io/evalsense/guide`. **(b)** The web-based user interface provided by the EvalSense library can be used to configure and execute evaluations, as well as to view their results. Alternatively, this can be done through code after importing the library.

tistical metrics (Zheng et al., 2023). However, the effectiveness of these methods is highly sensitive to the choice of model, prompt formulation, and general evaluation protocol. Notable examples include G-Eval (Liu et al., 2023) and GPTScore (Fu et al., 2024).

Hybrid methods also make use of pre-trained models, but only use these models for targeted subtasks as part of a structured evaluation framework. For instance, BERTScore (Zhang et al., 2020) compares texts based on contextual embeddings, while QAGS (Wang et al., 2020) assesses factual consistency using question answering.

2.2 LLM Evaluation Toolkits

A number of open-source toolkits have been introduced to support the evaluation of LLMs. Frameworks such as such as `lm-evaluation-harness` (Gao et al., 2024), OpenCompass (Contributors, 2023), and LightEval (Habib et al., 2023) primarily focus on benchmarking models against standardised tasks and datasets. While some of these tools also support evaluation on custom data, they generally lack dedicated mechanisms for guiding evaluation design or assessing the suitability of selected methods in specific domains. The FreeEval framework (Yu et al., 2024) extends beyond benchmarking by incorporating human judgements, bias detection, contamination analysis, and case-by-case inspection. However, it does not support automated meta-evaluation or provide interactive tools for evaluation strategy selection.

Among existing tools, Inspect (UK AI Security Institute, 2024) is especially relevant to our

work. EvalSense uses Inspect as its basis, inheriting its support for multiple model providers, tool use, agentic workflows and detailed logging infrastructure. Nevertheless, EvalSense significantly expands on this foundation through a more versatile and extensible pipeline tailored to custom datasets, improved resource management, support for advanced evaluation methods (including sophisticated LLM-as-a-judge and hybrid approaches), and its unique focus on meta-evaluation and domain-specific guidance. The bespoke components of EvalSense are described in the following section.

3 EvalSense Pipeline

EvalSense implements a robust and customisable pipeline that manages the key steps of the evaluation process, from data management and preprocessing, through LLM generation, to final evaluation and result analysis (as illustrated in Figure 1). Uniquely, the generation and result analysis modules provide built-in support for meta-evaluating the reliability of the used metrics in addition to simply returning their scores. The overall design of the pipeline supports reusability and extensibility by making individual components easily replaceable, enabling the use of custom datasets, LLMs, or evaluation methods.

3.1 Dataset Manager

Dataset managers are responsible for loading and generic preprocessing of the data on which the LLM is to be evaluated. For open-source datasets, this may involve downloading the data files from a publicly available repositories, while for internal datasets, the data will typically be loaded from a local file system or secure cloud storage. To simplify data management for custom datasets, EvalSense provides a base `DatasetManager` class that defines a general interface for data managers and implements helper methods for retrieving associated files based on paths specified in a dataset configuration file. These methods can be overridden to support more complex data loading and preprocessing workflows.

3.2 Task Preprocessor

Task preprocessors implement any additional preprocessing steps that may be required to prepare the data for a specific task, as a single dataset may potentially support multiple such tasks. In many simple cases where no additional preprocessing is needed, users can rely on the `DefaultTaskPreprocessor`, which acts as an identify function. For more complex scenarios, users can define a custom preprocessing function following the `TaskPreprocessingFunction` protocol, which can then be used with the standard `TaskPreprocessor`.

3.3 LLM Generation Steps

After preparing the data for the task, the pipeline generates LLM outputs for evaluation using predefined generation steps. These typically involve prompting the model with specific system and user prompts. Optionally, the generation steps can incorporate more advanced strategies, such as enabling access to external tools (e.g., via the Model Context Protocol[3]), incorporating model self-critiques (Madaan et al., 2023), or using agentic workflows like ReAct (Yao et al., 2023). For the purposes of meta-evaluation, the LLM generation steps can also apply targetted perturbations degrading the quality of the output texts in predictable ways. In EvalSense, generation steps are defined via the `GenerationSteps` class, and the model configuration is specified using the `ModelConfig`.

3.4 Evaluator

Evaluators implement automated methods for scoring model outputs based on predefined quality criteria. EvalSense includes several out-of-the-box score calculators, including:

- BLEU (`BleuPrecisionScoreCalculator`)
- ROUGE (`RougeScoreCalculator`)
- BERTScore (`BertScoreCalculator`)
- G-Eval (`GEvalScoreCalculator`)
- QAGS (`QagsScoreCalculator`)

These calculators can either be used independently or wrapped in an `Evaluator` class to be used as part of an evaluation pipeline. For convenience, EvalSense provides helper functions to easily initialise these evaluators (e.g., `get_bleu_evaluator` for BLEU). All key aspects of the evaluator configurations, such as the used prompts and models for LLM-as-a-judge approaches are fully customisable. Users may also implement new evaluators to be used as part of the pipeline.

[3] https://modelcontextprotocol.io/introduction

3.5 Result Analyser

While evaluators produce fine-grained scores for the individual samples and summary metrics for each configuration evaluated by the pipeline, result analysers can be used to summarise the results from multiple such configurations and surface higher-level trends. EvalSense currently includes three main analysers: `TabularResultAnalyser` (for tabular summaries), `MetricCorrelationAnalyser` (for inter-metric correlation analysis), and `MetaResultAnalyser`. The last-mentioned analyser is crucial for the meta-evaluation capabilities of EvalSense, and can assess the consistency of metric scores either with different levels of automated perturbations (increasingly degrading a specific aspect of the evaluated texts to obtain the ground-truth output rankings) or human annotations. Meanwhile, the correlation analysis provided by the `MetricCorrelationAnalyser` can be particularly helpful for identifying similarities between different evaluation methods.

3.6 Pipeline

All components are integrated through the `Pipeline`, which schedules and executes the planned experiments (i.e., different configurations to be evaluated). These experiments can be declared individually or in batches using the `ExperimentConfig` and `ExperimentBatchConfig` data classes, enabling systematic evaluation sweeps. By default, the pipeline attempts to schedule the experiments in an optimal order to minimise the number of necessary model loads for local models, while also enabling users to resume any failed model generation tasks. As outlined above, the pipeline also supports automated meta-evaluation, performing controlled perturbations during the generation stage and assessing the reliability of different evaluation metrics during result analysis.

3.7 Project

All outputs, results, and metadata from pipeline execution are tracked through the `Project` class, which maintains a record of all experiments associated with a given project, their status, and links to the relevant logs. This class also provides a high-level interface through which the pipeline and result analysers can access and update these logs.

4 Evaluation Case Study

Task Setup To demonstrate EvalSense's effectiveness, we apply it to LLM evaluation on the task of dialogue summarisation using the ACI-Bench dataset (Yim et al., 2023). In this setting, the LLM is tasked with generating a structured clinical note based on a doctor-patient dialogue transcript. Given that correctness and comprehensiveness of the generated notes are the most crucial qualities in this context, our evaluation primarily focuses on these aspects.

We used the 120 samples from the test partitions of the ACI-Bench dataset. Since we are using the original, unchanged dialogues from the dataset, the dataset manager and task preprocessor stages of our pipeline are mostly focused on loading the relevant samples without significant additional preprocessing.

For the LLM generation steps, we used the system and user prompts from (Kanithi et al., 2024). The user prompt specified the intended note structure and section headings, as more general instructions would make the output format ambiguous. We experimented with six different open-weight models: Llama 3.1 8B (Dubey et al., 2024), Phi 4 (Abdin et al., 2024), Qwen3 8B and Qwen3 14B (Yang et al., 2025), and Gemma 3 12B and Gemma 3 27B (Kamath et al., 2025). All models were run locally using vLLM (Kwon et al., 2023) in their default precisions, with the generation temperature set to 0.7, top-p sampling value of 0.95 and a seed of 42.

Evaluation Setup The case study involved a total of 13 variants of five major evaluators implemented in EvalSense: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020), G-Eval (Liu et al., 2023) and QAGS (Wang et al., 2020). For ROUGE, we considered ROUGE-1, ROUGE-2 and ROUGE-L. The G-Eval metric was used with two different prompt variations: a detailed prompt providing thorough instructions on how to evaluate a note and a brief prompt asking the model to evaluate general faithfulness and accuracy. We also experimented with different G-Eval judge models: Llama 3.1 8B, Qwen3 14B and Gemma 3 27B. For the QAGS metric, we considered two different versions: ternary QAGS that generates questions requiring ternary responses and judge QAGS using more open-ended questions with an LLM judge comparing the responses. Both considered variants of the QAGS score used Llama

Table 1: Results from LLM evaluation on the ACI-Bench case study task using statistical and hybrid evaluation methods. Best results are bolded, second-best results are underlined.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore F1	Ternary QAGS	Judge QAGS
Gemma 3 12B	**0.128**	0.514	0.207	**0.300**	0.666	0.842	0.817
Gemma 3 27B	0.120	0.502	0.198	0.291	0.668	**0.846**	**0.821**
Llama 3.1 8B	0.127	**0.534**	**0.221**	0.294	0.662	0.806	0.789
Phi 4 14B	0.120	0.504	0.207	0.290	**0.670**	0.832	0.811
Qwen3 8B	0.091	0.468	0.174	0.259	0.648	0.818	0.784
Qwen3 14B	0.100	0.451	0.170	0.259	0.640	0.810	0.793

Table 2: Results from LLM evaluation on the ACI-Bench case study task using different variants of G-Eval. Best results are bolded, second-best results are underlined

Model	Brief Gemma 3	Det. Gemma 3	Brief Llama 3.1	Det. Llama 3.1	Brief Qwen3	Det. Qwen3
Gemma 3 12B	**0.929**	0.904	0.847	0.834	0.777	0.665
Gemma 3 27B	0.926	**0.916**	**0.848**	0.840	0.798	0.640
Llama 3.1 8B	0.835	0.823	0.788	0.801	0.683	0.598
Phi 4 14B	0.906	0.892	0.826	0.836	0.763	0.662
Qwen3 8B	0.864	0.876	0.845	**0.854**	0.775	0.630
Qwen3 14B	0.885	0.899	0.843	0.849	**0.814**	**0.682**

Table 3: Results from perturbation-based meta-evaluation of the different evaluation methods. Methods are ordered from best to worst.

Method Name	Avg. Correlation
G-Eval (Detailed, Gemma 3 27B)	0.999
G-Eval (Brief, Gemma 3 27B)	0.998
G-Eval (Detailed, Llama 3.1 8B)	0.995
G-Eval (Brief, Llama 3.1 8B)	0.992
Ternary QAGS (Llama 3.1 8B)	0.982
Judge QAGS (Llama 3.1 8B)	0.969
G-Eval (Brief, Qwen 3 14B)	0.967
G-Eval (Detailed, Qwen 3 14B)	0.924
BERTScore F1	0.431
ROUGE-1	0.323
BLEU Precision	0.296
ROUGE-L	0.232
ROUGE-2	0.049

3.1 8B as the underlying model.

For our meta-evaluation, we used a set of three prompts instructing the model to apply different levels of perturbations to the note: one rephrasing the note without changing its meaning, one introducing minor content changes and one significantly changing the meaning of the note. The used prompts are given in Appendix A.

Results The results of our evaluation are summarised in Tables 1 and 2. We can observe that there is substantial disagreement among the different methods, with no universally best-performing model. Without further information on each method's reliability for this task, drawing definitive conclusions would be difficult.

However, based on the meta-evaluation results in Table 3, we can assign greater weight to G-Eval variants using Gemma 3 and Llama 3.1, as well as both QAGS versions. These methods consistently rank the Gemma 3 models highest, except for G-eval with Llama 3.1 using the detailed prompt. Notably, statistical metrics and BERTScore underperform compared to LLM-based methods.

5 Conclusion

In this paper, we introduced EvalSense, a novel framework for systematic evaluation of LLMs on custom tasks. Unlike other toolkits, which mainly focus on direct application of evaluation methods without providing principled ways to assess their suitability, EvalSense guides users in selecting evaluation approaches tailored to their specific domains and provides quantitative insights about the effectiveness of these approaches through meta-evaluation. We demonstrated its capabilities through a case study on structured clinical note generation from doctor-patient dialogues, showing that it supports robust evaluation even when different evaluation methods yield disagreeing results.

Acknowledgments

We thank the UK AI Security Institute and the wider development team for their work on the Inspect framework, which serves as a basis for EvalSense.

References

Marah I Abdin, Jyoti Aneja, Harkirat S. Behl, Sébastien Bubeck, Ronen Eldan, and 1 others. 2024. Phi-4 technical report. *CoRR*, abs/2412.08905.

Elron Bandel, Yotam Perlitz, Elad Venezian, Roni Friedman, Ofir Arviv, Matan Orbach, Shachar Don-Yehiya, Dafna Sheinwald, Ariel Gera, Leshem Choshen, Michal Shmueli-Scheuer, and Yoav Katz. 2024. Unitxt: Flexible, shareable and reusable data preparation and evaluation for generative AI. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 207–215, Mexico City, Mexico. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712.

Felix Busch, Lena Hoffmann, Christopher Rueger, Elon HC van Dijk, Rawen Kader, Esteban Ortiz-Prado, Marcus R. Makowski, Luca Saba, Martin Hadamitzky, Jakob Nikolas Kather, Daniel Truhn, Renato Cuocolo, Lisa C. Adams, and Keno K. Bressem. 2025. Current applications and challenges in large language models for patient care: a systematic review. *Communications Medicine*, 5(1):26.

OpenCompass Contributors. 2023. OpenCompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`.

Badhan Chandra Das, M. Hadi Amini, and Yanzhao Wu. 2025. Security and privacy challenges of large language models: A survey. *ACM Comput. Surv.*, 57(6).

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and 1 others. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for LLM evaluation.

Steve Han, Gilberto Titericz Junior, Tom Balough, and Wenfei Zhou. 2025. Judge's verdict: A comprehensive analysis of LLM judge capability through human agreement. *CoRR*, abs/2510.09738.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, and 1 others. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Praveen K. Kanithi, Clément Christophe, Marco AF Pimentel, Tathagata Raha, Nada Saadi, Hamza Javed, Svetlana Maslenkova, Nasir Hayat, Ronnie Rajan, and Shadab Khan. 2024. MEDIC: Towards a comprehensive framework for evaluating llms in clinical applications. *CoRR*, abs/2409.07314.

Seungone Kim, Jamin Shin, Yejin Choi, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K. Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O. Cohen, Valentina Borghesani, Anton Pashkov, Daniele Marinazzo, Jonathan Nicholas, Alessandro Salatiello, Ilia Sucholutsky, Pasquale Minervini, Sepehr Razavi, Roberta Rocca, Elkhan Yusifov, Tereza Okalova, and 20 others. 2025. Large language models surpass human experts in predicting neuroscience results. *Nature Human Behaviour*, 9(2):305–315.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, Le Hou, Yong Cheng, Yun Liu, S. Sara Mahdavi, Sushant Prakash, Anupam Pathak, Christopher Semturs, Shwetak Patel, Dale R. Webster, and 9 others. 2025. Towards accurate differential diagnosis with large language models. *Nature*.

Bhuvanashree Murugadoss, Christian Pölitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. 2025. Evaluating the evaluator: Measuring LLMs' adherence to task evaluation instructions. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 19589–19597. AAAI Press.

Jasmine Chiat Ling Ong, Shelley Yin-Hsi Chang, Wasswa William, Atul J. Butte, Nigam H. Shah, Lita Sui Tjien Chew, Nan Liu, Finale Doshi-Velez, Wei Lu, Julian Savulescu, and Daniel Shu Wei Ting. 2024. Ethical and regulatory challenges of large language models in medicine. *The Lancet Digital Health*, 6(6):e428–e432.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca A. Popa, and Ion Stoica. 2025. Judgebench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

UK AI Security Institute. 2024. Inspect AI: Framework for Large Language Model Evaluations.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, Nidhi Rohatgi, Poonam Hosamani, William Collins, Neera Ahuja, Curtis P. Langlotz, Jason Hom, Sergios Gatidis, John Pauly, and Akshay S. Chaudhari. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*, 30(4):1134–1142.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

WHO. 2023. WHO calls for safe and ethical AI for health.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific Data*, 10(1):586.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Zhengran Zeng, Wei Ye, Jindong Wang, Yue Zhang, and Shikun Zhang. 2024. FreeEval: A modular framework for trustworthy and efficient evaluation of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 1–13, Miami, Florida, USA. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Used prompts

A.1 Note Generation Prompt

The prompt used for the ACI-Bench note genration, adapted from (Kanithi et al., 2024), is shown in Listing 1.

A.2 Perturbation Prompt 1

The prompt used for rephrasing the output notes without changing their meaning is shown in Listing 2.

A.3 Perturbation Prompt 2

The prompt used for introducing minor content changes is shown in Listing 3.

A.4 Perturbation Prompt 3

The prompt used for significantly changing the meaning of the generated notes is given in Listing 4.

```
Your task is to generate a clinical note based on a conversation between a doctor
    \ and a patient. Use the following format for the clinical note:

1. **CHIEF COMPLAINT**: [Brief description of the main reason for the visit]
2. **HISTORY OF PRESENT ILLNESS**: [Summary of the patient's current health status
    \ and any changes since the last visit]
3. **REVIEW OF SYSTEMS**: [List of symptoms reported by the patient]
4. **PHYSICAL EXAMINATION**: [Findings from the physical examination]
5. **RESULTS**: [Relevant test results]
6. **ASSESSMENT AND PLAN**: [Doctor's assessment and plan for treatment or further
    \ testing]

**Conversation:**
{prompt}

**Note:**
```

Listing 1: Note generation prompt

```
Your task is to generate a clinically plausible variation of the provided clinical
    \ note.

You should maintain the original note's structure and formatting, but modify its
    \ content according to the specified types of perturbation below. Try to
    \ maintain internal consistency and general medical plausibility when applying
    \ any changes.

**Perturbation Instructions**
Apply the following types of perturbations:
- Rephrase sentences while preserving the exact medical meaning. You may use
    \ synonyms, vary sentence structure, or change sentence length, but all
    \ clinical facts and measurements must remain unchanged.
- Slightly alter the writing style, such as using different terminology or
    \ presenting findings differently, while ensuring the factual content remains
    \ identical.

Respond only with the perturbed clinical note, do not include any commentary,
    \ reasoning or explanation.

**Original Clinical Note**
{prompt}

**Perturbed Clinical Note**
```

Listing 2: Perturbation prompt 1

```
Your task is to generate a clinically plausible variation of the provided clinical
    \ note.

You should maintain the original note's structure and formatting, but modify its
    \ content according to the specified types of perturbation below. Try to
    \ maintain internal consistency and general medical plausibility when applying
    \ any changes.

**Perturbation Instructions**
Apply the following types of perturbations:
- Make small changes to test results and quantitative measurements, ensuring they
    \ remain clinically plausible and consistent with the original context.
- Introduce minor modifications to the patient's reported symptoms, making sure
    \ they are still consistent with the assessment, diagnosis, and treatment plan
    \ (e.g., adding or substituting symptoms that commonly co-occur).
- Slightly adjust the patient's clinical history, ensuring consistency with the
    \ assessment, diagnosis, and treatment plan.
- Make minor modifications to the treatment plan, but ensure it remains appropriate
    \  for the assessment and diagnosis.

Respond only with the perturbed clinical note, do not include any commentary,
    \ reasoning or explanation.

**Original Clinical Note**
{prompt}

**Perturbed Clinical Note**
```

Listing 3: Perturbation prompt 2

```
Your task is to generate a clinically plausible variation of the provided clinical
    \ note.

You should maintain the original note's structure and formatting, but modify its
    \ content according to the specified types of perturbation below. Try to
    \ maintain internal consistency and general medical plausibility when applying
    \ any changes.

**Perturbation Instructions**
Apply the following types of perturbations:
- Significantly alter test results and quantitative measurements, in a way that may
    \  change the clinical interpretation or implications of the note.
- Make substantial changes to the patient's reported symptoms, potentially
    \ affecting the clinical interpretation of the note.
- Make substantial changes to the patient's clinical history, potentially affecting
    \  the clinical interpretation.
- Significantly modify the treatment plan, such that it may lead to a different
    \ clinical outcome than the original plan.

Respond only with the perturbed clinical note, do not include any commentary,
    \ reasoning or explanation.

**Original Clinical Note**
{prompt}

**Perturbed Clinical Note**
```

Listing 4: Perturbation prompt 3