# ENTERPRISE LARGE LANGUAGE MODEL EVALUATION BENCHMARK

Liya Wang, David Yi, Damien Jose, John Passarelli, James Gao, Jordan Leventis and Kang Li

Atlassian, USA

***ABSTRACT***

*Large Language Models (LLMs) have demonstrated promise in boosting productivity across AI-powered tools, yet existing benchmarks like Massive Multitask Language Understanding (MMLU) inadequately assess enterprise-specific task complexities. We propose a 14-task framework grounded in Bloom's Taxonomy to holistically evaluate LLM capabilities in enterprise contexts. To address challenges of noisy data and costly annotation, we develop a scalable pipeline combining LLM-as-a-Labeler, LLM-as-aJudge, and corrective retrieval-augmented generation (CRAG), curating a robust 9,700-sample benchmark. Evaluation of six leading models shows open-source contenders like DeepSeek R1 rival proprietary models in reasoning tasks but lag in judgment-based scenarios, likely due to overthinking. Our benchmark reveals critical enterprise performance gaps and offers actionable insights for model optimization. This work provides enterprises a blueprint for tailored evaluations and advances practical LLM deployment.*

***KEYWORDS***

*Large Language Models (LLMs), Evaluation Benchmark, Bloom's Taxonomy, LLM-as-a-Labeler, LLMas-a-Judge, Corrective Retrieval-Augmented Generation (CRAG)*

## 1. INTRODUCTION

Large Language Models (LLMs) are transforming enterprise operations by automating tasks such as data analysis, code generation, and document creation. These capabilities not only increase efficiency but also inform strategic resource allocation. Moreover, recent advances in opensource LLMs have brought their performance on par with proprietary models, offering costeffective solutions that strengthen enterprise AI ecosystems.

Robust benchmark datasets are essential for reliably evaluating the performance of fine-tuned LLMs. Such datasets should be diverse, novel, and sufficiently challenging [1], enabling engineers to determine the most suitable base models and identify areas where performance gaps still exist.

While multiple benchmarks such as Multitask Language Understanding (MMLU) [2], Big-Bench [3], ARC [4], Holistic Evaluation of Language Models (HELM) [5], MT-Bench and Chatbot Arena [6] exist, many of these often fail to account for the unique characteristics of enterprise applications. Additionally, these benchmarks are becoming saturated, suggesting that current evaluation methods may no longer effectively distinguish between newer, more advanced models and their predecessors.

David C. Wyld et al. (Eds): MLNLP, ASOFT, CSITY, NWCOM, SIGPRO, AIFZ, ITCCMA – 2025
pp. 01-19, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.152001

Efforts to establish benchmarks for specific domains are evident across various fields. In the medical field, benchmarks including MedQA [7], MedQA-CS [8], MedMCQA [9], WorldMedQA-V [10], PubMedQA [11] and CMB [12] have been introduced. The financial sector has seen the introduction of benchmarks such as FinBen [13], FinEval [14], FLUE [15], BBT-Fin [16], XuanYuan 2.0 [17] and PIXIU [18]. In the legal realm, various benchmarks have been developed, such as LexGlue [19], LBOX OPEN [20], legal-bench [21], and LawBench [22]. In the realm of Chinese language evaluation, benchmarks like CMMLU [23], GAOKAO [24], and C-Eval [25] have been developed. For other languages, initiatives such as PersianMMLU [26], M3Exam [27] and AGIEval [28] have been proposed.

Our research aims to create a comprehensive enterprise evaluation benchmark dataset to assess the strengths and limitations of current LLMs in enterprise context. This will provide insights for LLM post-training and serve as a cost-saving blueprint for similar organizations.

## 2. RELATED WORK

### 2.1. LLM-as-a-Labeler

In many enterprises, raw, unlabeled, and unstructured text data is abundant, while labeled data remains scarce. Traditionally, data annotation has relied on human labelers, a process that is timeconsuming, costly, subjective, and difficult to scale. With the advent of LLMs like GPT-4o, organizations have a promising opportunity to revolutionize data annotation [29], [30], [31], [32]. LLMs offer several advantages for data annotation. They significantly enhance efficiency by rapidly processing large volumes of data, thereby reducing the time required to produce labelled datasets. They also improve consistency and objectivity, providing standardized annotations that minimize bias. Moreover, LLMs are highly scalable, enabling them to handle increasing data volumes without a corresponding increase in time or cost. They can flexibly generate data to meet specific requirements, such as identifying disease names in the medical field. Additionally, LLMs can adapt and learn from new data, enhancing annotation accuracy over time.

When combined with carefully crafted prompts or retrieval-augmented generation (RAG) techniques, LLMs can understand context and semantics with remarkable precision, supporting various annotation tasks. For example, they have been employed to generate instructions [33]; responses [17]; question-answering pair [34]; reasoning data [35], [36]; pairwise preference data [37], [38], [39], [40]; and textual feedback [41]. They are also effective in multiple-choice question answering (MCQA) [42] and in assigning confidence scores to assess annotation reliability [43].

In summary, incorporating LLMs into data annotation processes reduces costs, enhances data quality, and empowers enterprises to leverage their data more effectively.

### 2.2. Retrieval Augmented Generation (RAG)

LLMs are typically trained on historical data and often lack access to proprietary knowledge unique to individual enterprises. This limitation can lead to inaccuracies or hallucinations in their outputs. Retrieval-Augmented Generation (RAG) [44] addresses these limitations by incorporating domain-specific data, particularly private data, to tailor LLM applications to meet specific enterprise needs. LLMs enhanced with RAG offer several key benefits, including increased professionalism and timeliness, better alignment with domain experts, reduced likelihood of hallucinations, and improved controllability and explainability [45].

Recent advances have led to the development of several cutting-edge RAG techniques designed to overcome the limitations of basic RAG. For instance, MemoRAG [46] enhances retrieval by incorporating long-term memory capabilities, while Adaptive-RAG [47] dynamically selects strategies based on query complexity. Self-RAG [48] selectively retrieves knowledge and employs a critic model to determine whether retrieval is necessary. Similarly, Self-Knowledge guided Retrieval (SKR) [49] leverages both internal and external knowledge by extracting selfknowledge from LLMs. Retrieval-Augmented Language Models (RALMs) ) [50] integrate a natural language inference (NLI) model to identify and filter out irrelevant context, thereby enhancing robustness. Another innovative approach is SAIL [51], which enables fine-tuned language models to source, denoise, and reason using a combination of relevant and distracting search results. Additionally, Corrective Retrieval-Augmented Generation (CRAG) [52] integrates self-assessment, evaluation mechanisms, and large-scale web searches for document retrieval, collectively improving output quality.

### 2.3. LLM-as-a-Judge

Evaluating the outputs of LLMs poses significant challenges due to their complexity, subjective nature, and the wide variety of tasks they perform. Two common evaluation methods are used: automatic and human evaluation. Automatic evaluation, often referred to as “LLM-as-a-Judge,” is faster, more cost-effective, and potentially more reliable than human evaluation (e.g., see [53], [54], [55], [6], [56], [57], [58] etc.). Currently, the method is commonly employed for a wide range of tasks including scoring, preference ranking, and candidate selection.

Despite its widespread use, there remains a disparity between LLM-as-a-Judge and human evaluation. Several techniques have been proposed to enhance the effectiveness of LLM-as-aJudge. For example, PandaLM [59] enables reproducible and automated language model assessment by training an LLM to act as the "judge" in evaluating different models. Some methods incorporate pre-defined rules into prompts, directing the LLMs to adhere to explicit guidelines [60]. Additionally, swapping the order of two responses can help mitigate positional bias [61]. REVISEVAL leverages the text revision capabilities of LLMs to adaptively revise responses, treating the revised text as the reference (response-adapted reference) for subsequent evaluation [62]. G-Eval [63] adopts the Chain-of-Thoughts (CoT) approach [64] for better evaluation which has been widely adopted for its flexible creation of task-specific metrics.

LLM-as-a-Judge also faces certain challenges. For example, it has been observed [6], [58] that even with CoT prompts, LLM-as-a-Judge are sensitive to prompt complexity and length, and a tendency toward leniency. To address this, a reference-guided method has been proposed, where the LLM-as-a-Judge first generates a response independently based on given instructions and then uses this response as a reference in the evaluation prompt. Further research is needed to address those limitations.

## 3. BENCHMARKING METHODOLOGY

In this section, we offer an in-depth explanation of the principles guiding the design of our enterprise benchmark and the selection of test tasks. When compiling our enterprise benchmark dataset, we adhere to the traditional evaluation framework outlined in Figure 1 [65]. This framework composes of three parts: 1) what to evaluate: task construction; 2) where to evaluate: data construction; 3) how to evaluate: evaluation metrics calculation. Next, we present more details on implementing these three parts.

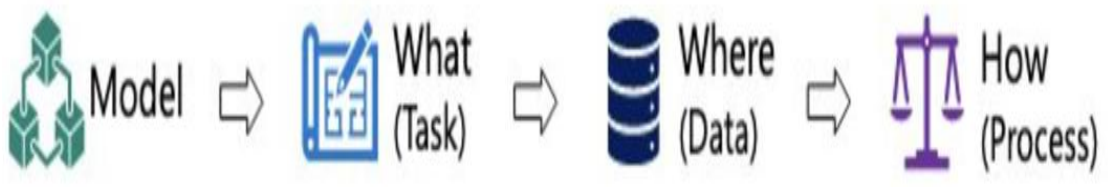


Figure 1. Evaluation pipeline [65]

### 3.1. Evaluation Tasks

To evaluate LLMs effectively, it is crucial to assemble a diverse set of tasks. Our extensive understanding of enterprise use cases has allowed us to categorize them into 14 distinct tasks, which we believe are commonly encountered across our operations. To systematically organize these tasks, we have chosen Bloom's Taxonomy [66] as an optimal framework.

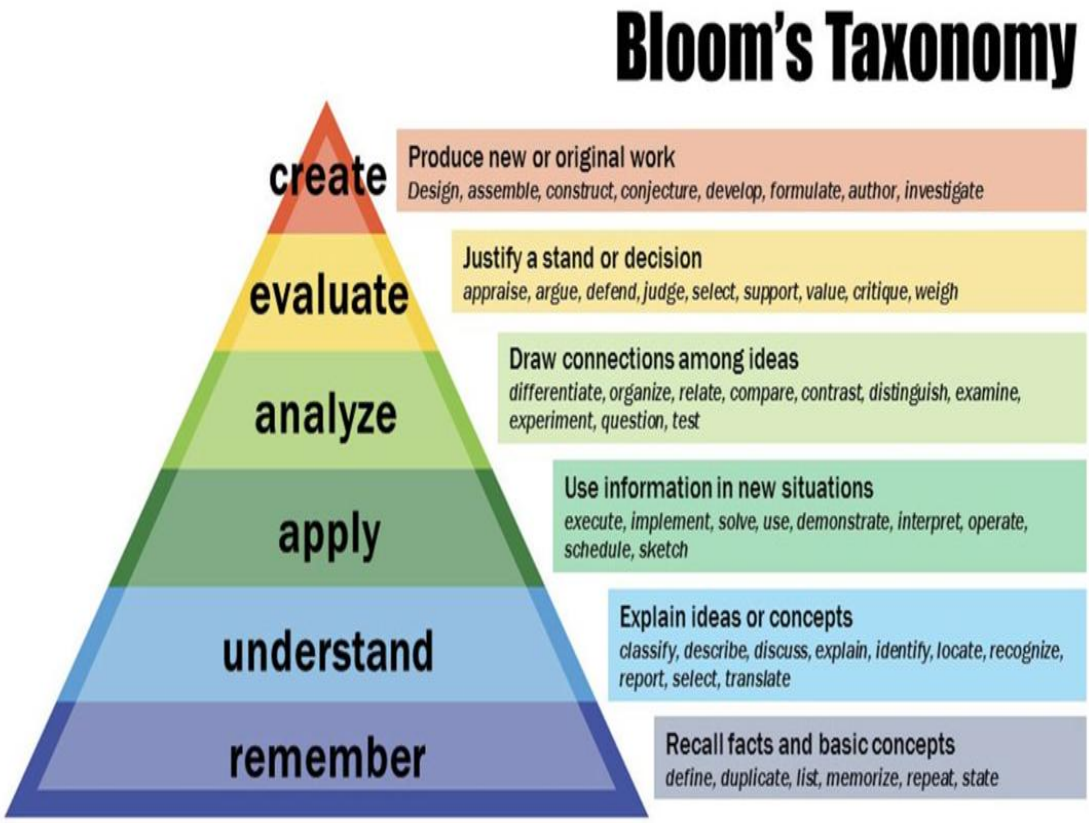


Figure2.Bloom'sTaxonomy[67]

Table 1. Task list in Atlassian benchmark.

| Cognitive Level | Task ID | Task Description | Data Source | Labeling Method | Size |
|---|---|---|---|---|---|
| Remember | 1-1 | Acronyms Memorization (AM) | Confluence | LLM | 600 |
| | 1-2 | Factual QuestionAnswering | Confluence | LLM | 1005 |
| Understand | 2-1 | Toxicity | Rovo chat | LLM | 1000 |
| | 2-2 | Bias | Rovo chat | LLM | 1000 |
| | 2-3 | Sentiment Analysis (SA) | Customer feedback | LLM | 1000 |
| | 2-4 | Named Entity Recognition (NER) | Confluence | LLM | 536 |
| | 2-5 | Question Answering (QA) | Rovo chat | LLM | 708 |
| Apply | 3-1 | Summarization | Confluence + Rovo chat | - | 600 |
| | 3-2 | Software Engineering (SWE) | Rovo chat + Developer documentation | LLM+RAG | 600 |
| | 3-3 | Machine Learning Engineering (MLE) | Rovo chat+ Developer documentation | LLM+RAG | 610 |
| | 3-4 | NL2JQL | Rovo chat | Manual | 218 |
| Analyze | 4-1 | Hallucination Detection (HD) | Confluence | LLM | 595 |
| Evaluate | 5-1 | LLM-as-a-Judge | Slack query | Manual | 265 |
| Create | 6-1 | Content Generation | Rovo chat | - | 524 |

Originally developed by Benjamin Bloom and his colleagues in 1956, Bloom's Taxonomy has been widely used for curriculum design and learning assessment by K-12 teachers and colleague instructors. It categorizes cognitive objectives into six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create, as illustrated in Figure 2.

By structuring our benchmark design around Bloom's Taxonomy, we ensure a comprehensive approach to evaluating the cognitive levels in LLMs. This framework helps identify gaps in knowledge and application while guiding the fine-tuning process. Ultimately, this leads to more effective and innovative advancements in enterprise LLMs. Additionally, it promotes a systematic and standardized method for assessing progress, which facilitates communication and collaboration among researchers and stakeholders.

In our designed benchmark, each of the 14 tasks is assigned a unique identifier for clarity: the first digit indicates the cognitive level, while the second digit represents the task's sequence within that category. Table 1 provides a complete list of these tasks. The tasks cover areas such as text understanding (e.g., IDs 2-1, 2-2, 2-3, 2-4), text generation (e.g., IDs 1-2, 2-5, 3-1, 6-1), and reasoning (e.g., IDs 3-2, 3-3, 3-4, 4-1, 5-1). Below is a brief description of each task:

**Remember**

**1-1. Acronyms Memorization (AM):** This task evaluates the ability to recall and identify specific acronyms used within the Atlassian ecosystem. LLMs are given a list of acronyms from Confluence data, which is internal Atlassian information embedded in Confluence software. The models are then asked to accurately explain their definitions.

**1-2. Factual Question-Answering:** In this task, LLMs answer factual questions based on information extracted from Confluence data. The objective is to evaluate the participant's ability to recall specific facts accurately.

**Understand**

**2-1. Toxicity Detection:** This task evaluates the detection of toxicity in Rovo chat outputs. LLMs must accurately identify instances of toxicity, such as personal attacks, mockery, hate, or threats.

**2-2. Bias Detection:** Similar to the toxicity task, this involves detecting biased language within Rovo chat outputs. The objective is to identify statements containing gender, racial, or political bias.

**2-3. Sentiment Analysis (SA):** This task analyzes customer feedback data to determine the sentiment expressed (positive, negative, or neutral). It tests the ability to understand nuanced emotional content, which can guide future business engagements.

**2-4. Named Entity Recognition (NER):** This task requires identifying and categorizing named entities (such as person, organization, date, location) in text from Confluence data. Exact match is the metric used, reflecting the precision of entity recognition.

**2-5. Question Answering:** This involves answering questions based on real data collected from Rovo chat data. It tests organizational knowledge in LLMs.

**Apply**

**3-1. Summarization:** The goal is to produce concise summaries of user inputs from Rovo chat data, encompassing diverse requirements such as resumes, meeting transcripts, articles, and task lists. This task emphasizes extracting key information while maintaining relevance and coherence in the summaries.

**3-2. Software Engineering (SWE):** This task involves applying software engineering principles to solve problems or complete assignments from real tasks in Rovo chat data.

**3-3. Machine Learning Engineering (MLE):** Similar to the SWE task, this focuses on machine learning applications from real tasks in Rovo chat data. LLMs are evaluated on their ML knowledge and skills.

**3-4. NL2JQL Conversion:** This task converts natural language queries into Jira Query Language (JQL) collected from real tasks in Rovo chat data. It evaluates the ability to translate human language into precise Jira queries.

**Analyze**

**4-1. Hallucination Detection (HD):** Since Atlassian is a software company, this task is specifically designed to assess the ability of LLMs to identify hallucinations when responding to questions about Atlassian's software development.

**Evaluate**

**5-1. LLM-as-a-Judge:** This task evaluates the ability of LLMs to assess the quality of Slack search queries on a scale from 1 to 5, where a higher score indicates a better query. The objective is to determine how well the model's predicted ratings align with human-assigned labels. Spearman's rank correlation coefficient (Spearman's r) is used to measure this alignment.

**Create**

**6-1. Content Generation:** This task involves asking LLMs to generate coherent and contextually relevant content from real tasks collected from Rovo Chat data.

After developing our benchmark task design, we conducted a sample size ablation study to identify the optimal sample size for each task. Our findings suggest that a sample size of approximately 600 provides stable evaluation results. However, due to limitations associated with certain manual labeling tasks, we have decided to retain the current sample size for these tasks. Consequently, our benchmark is substantial, consisting of approximately 9,700 data samples.

## 3.2. Data Curation

To curate data for the 14 tasks, we have developed a cost-effective and efficient data curation pipeline, as illustrated in Figure 3. Each stage of the pipeline is detailed below:

- **Raw Data Collection:** Our process begins by gathering raw text data from key Atlassian datasets, including: (1) Confluence data [68], (2) Rovo chat data [69], (3) customer feedback data, (4) Slack query data, and (5) developer documentation data. Detailed sources for each task are listed in Table 1. Notably, our dataset excludes any user-generated content (UGC).

- **Data Cleaning:** Raw data often contain noise, duplicates, inconsistencies, and errors. To address these issues, we meticulously perform grammar checks, remove redundancies, and conduct paraphrase mining [70]. This step ensures high-quality data for subsequent stages.

- **LLM-as-a-Labeler:** As manual labeling is costly, time-consuming, and non-scalable, we utilize GPT-4o [71] to assist with data annotation, thereby streamlining the process. Additionally, we employ (CRAG) [52] to incorporate enterprise and web knowledge to reduce hallucination. Appendix A has a brief introduction to CRAG.

- **LLM-as-a-Judge Evaluation:** Following annotation, we use LLM-as-a-Judge to evaluate label quality. The DeepEval package [72] was deployed, which also supports G-Eval. Appendix A also includes details of G-Eval.

- **Human Validation:** To ensure quality, human experts reviewed a subset of the annotated data that received low confidence scores from previous stage. These human

reviewers validate the labels and correct any errors or inconsistencies, further enhancing the overall label quality.

After completing these five stages, the labeled dataset is integrated into our benchmark. Table 1 provides details on the data source, annotation method, and size for each task. In most cases, we use LLMs for data annotation. However, for Task 5-1 ("LLM-as-a-Judge"), ground truth search query labels are collected manually; similarly in Task 4-1 ("NL2JQL"), JQLs are also collected manually. Tasks 3-1 ("Summary") and 6-1 ("Content Generation") are open-ended and do not require ground truth labels.

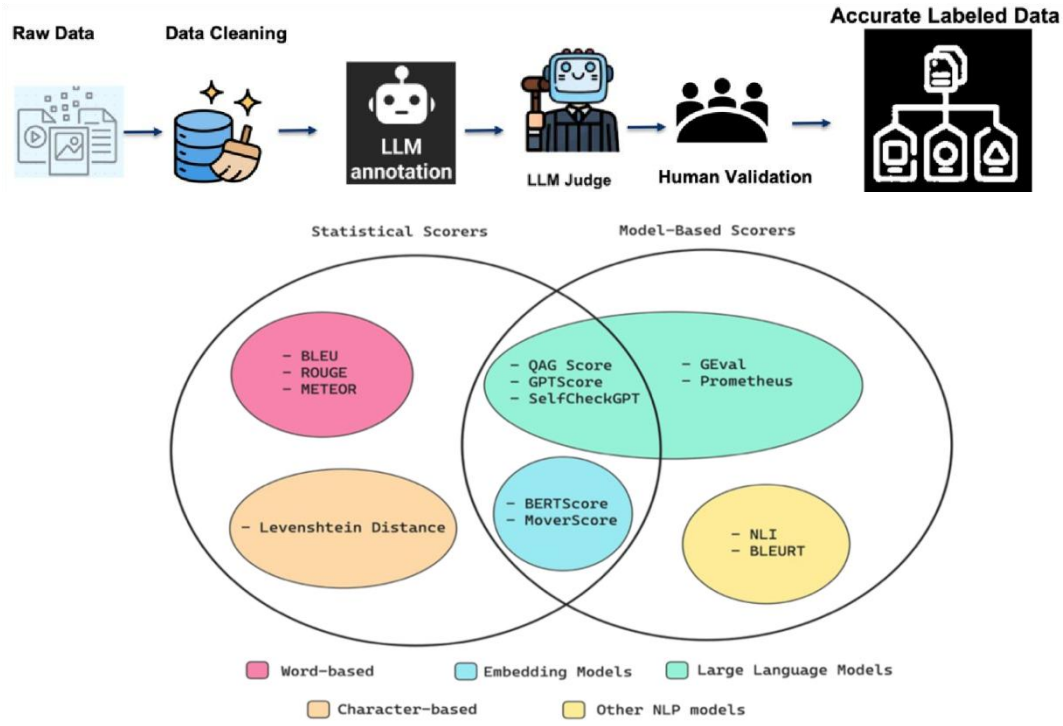


Figure 4. Different methods for metric calculation [73]

## 3.3.Evaluation Metrics Calculation

Evaluating LLMs necessitates a comprehensive approach that considers multiple dimensions of the model's output, including the accuracy and relevance of its responses as well as its ability to retrieve and integrate external information. Over time, several established methods for calculating metric scores have been proposed. Early research relied solely on statistical analysis, while more recent work has utilized neural networks, including embedding models and LLMs. Figure 4 provides a concise summary of the evolution of evaluation metric calculation methods, highlighting significant developments and innovations in the field [73].

Lexicon-based methods, such as the Bilingual Evaluation Understudy (BLEU) [74] and the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [75], assess the overlap between output and reference texts using n-grams. These methods do not consider semantics and have very limited reasoning capabilities. Consequently, they have been criticized for their low correlation with human judgments [76], as surface-level matching is insufficient for reliably evaluating text. Therefore, they are not accurate enough for evaluating LLM outputs, which are often lengthy and complex.

With the rise of deep learning, model-based evaluation metrics such as BERTScore [77] and BARTScore [78] have been proposed to evaluate the overall quality or specific aspects of generated outputs. These methods leverage pre-trained language models like BERT or BART to calculate semantic similarity between model-generated texts and reference texts. Although they perform better than lexicon-based methods, their effectiveness is still limited, and their scope of application is restricted. For instance, BERTScore is reference-based and cannot be used without a reference [79].

With the emergence of LLMs, the practice of using LLMs as judges has become increasingly prevalent due to their exceptional ability to follow instructions and comprehend context with high accuracy. Several studies have suggested that LLMs perform comparably to crowdsourced annotators across a variety of tasks [80], [81], [82], [83]. Consequently, in this study, we follow this new practice. Specifically, we employ the G-Eval method for its flexibility and ease-of-use. Considering the specific characteristics of each task, we evaluate LLMs on accuracy, correctness, coherence, relevance, bias, and toxicity etc. Table 2 presents a comprehensive list of evaluation metrics for each task type, including the following:

- **Correctness**: A customized metric to assess whether the output is accurate compared to the expected result. The metric ranges from 0 to 1, where 1 indicates totally correct and 0 signifies a complete discrepancy. This metric is applied to question-answering tasks [84].

- **ToxicityMetric**: A reference-less metric that assesses the presence of toxicity in LLM outputs. An opinion is deemed toxic if it includes personal attacks, mockery, hate, dismissive statements, threats, or intimidation. The toxicity score is calculated as: Toxicity = Number of Toxic Opinions / Total Number of Opinions [85].

- **BiasMetric**: This metric identifies the presence of gender, racial, political, or geographical bias in LLM outputs. The bias score is determined by the formula: Bias = Number of Biased Opinions / Total Number of Opinions [86]

- **Exact Match**: A stringent evaluation metric that compares a predicted answer to a reference answer to determine if they are identical. This metric is used for Named Entity Recognition (NER) task evaluation.

- **Hallucination Percentage**: This metric measures the rate of hallucinations generated by an LLM. We leverage HallucinationMetric framework [87] to evaluate each output with a binary scoring system: 1 indicates the presence of hallucinations, while 0 denotes hallucination-free responses. The percentage is computed as the ratio of flagged cases (score = 1) to the total number of test cases.

- **Spearman's r**: A non-parametric measure of the strength and direction of association between two ranked variables. It assesses the extent to which the relationship between variables can be described by a monotonic function. Spearman's r ranges from -1 to 1, with +1 indicating a perfect positive monotonic relationship, -1 indicating a perfect negative monotonic relationship, and 0 indicating no monotonic relationship. This metric is particularly useful for measuring alignment between LLM judges and human annotation when dealing with ordinal data [88].

- **Relevance**: This metric evaluates how well the generated text aligns with the input, ensuring that the output is contextually appropriate and meets the user's intent or informational needs. We use it for summarization task evaluation.

- **Coherence**: A metric that assesses the logical flow and consistency of the generated text. It examines whether the sentences and ideas are meaningfully connected, maintaining a clear and understandable narrative or argument. Coherence ensures that the text is not only relevant but also logically sound, without contradictions or disjointed thoughts. This metric is applied to our content generation task.

# 4. EXPERIMENT

## 4.1. Model

In our study, we conduct a comprehensive evaluation of six recently released LLMs to assess their performance on our curated benchmark. The models under consideration include five opensource models and one proprietary ones:

**Llama 3.2 3B (Open-source)** [89]: Developed as part of the Llama series by Meta, this opensource model comprises 3 billion parameters and can be freely accessed and modified. Its relatively smaller size, compared to some other models, allows for efficient deployment in environments with limited computational resources. We investigate its strengths and weaknesses on our specific tasks, evaluating its potential for future post-training.

**Llama 3.3 70B (Open-source)** [90]: With 70 billion parameters, this model is the larger counterpart within the Llama family, offering enhanced capacity to handle complex language tasks. The model's size suggests a greater ability to capture nuanced linguistic patterns and perform sophisticated reasoning tasks.

**Llama 4 Scout (Open-source)** [91]: Llama 4 Scout is an open-source mixture-of-expert (MoE) model featuring 17 billion parameters distributed across 16 experts. It supports a 10 million-token context window, allowing it to process a wide range of data types, including both text and images. The model can be efficiently deployed on a single NVIDIA H100 GPU.

**DeepSeek R1 671B (Open-source)** [35]: On January 22, 2025, DeepSeek-AI released a new open-source reasoning suite, comprising DeepSeek-R1-Zero, DeepSeek-R1, and six distilled variants based on Qwen [92] and Llama. DeepSeek R1's performance parallels proprietary models like OpenAI's o1. A key feature is its detailed CoT reasoning, clearly marked between <think> and </think>, which boosts interpretability.

**DeepSeek Distilled Llama 3.3 70B (Open-source)** [35]: Our work also explores the R1 distilled Llama 3.3 70B model to further investigate the capabilities of smaller reasoning models. Through this analysis, we aim to gain deeper insights into how it compares to the original Llama 3.3 70B model, identifying its strengths and areas where it excels.

**GPT-4o-2024-11-20 (Proprietary)**: This is one of the GPT-4o series developed by OpenAI, renowned for its leading performance. Our evaluation aims to understand how this model performs across diverse enterprise tasks and real-world scenarios compared to other open-source contemporaries.

Through this evaluation, we aim to provide insights into the current landscape of LLMs, highlighting the advantages and limitations of proprietary versus open-source models, the impact of model size on performance, and practical considerations for deploying these models in our applications. In addition, as for the evaluation metrics calculation stage, G-Eval uses the GPT4o-2024-11-20 version.

## 4.2. Experiment Setup

We configured the temperature parameter to 0.0 and set the top-p value to 0.9. The maximum token count was tailored to the specific characteristics of each task. Recognizing that reasoning models require more tokens for their thought processes, we allocated an additional 2000 tokens specifically for these models. Prompts were thoughtfully crafted for each task. Our evaluation was conducted in zero-shot scenarios, using inputs composed only of task instructions and queries.

## 4.3. Main Results

This section presents the evaluation results of seven popular LLMs tested using our proposed benchmark. We begin by selecting a representative task from each cognitive level to illustrate the performance comparison, as shown in Figure 5. provides detailed results for each task. Generally, higher values for evaluation metrics suggest better performance, except for tasks 2-1, 2-2, and 41, where lower values are preferable. The G-Eval metrics range from 0 to 1. The best-performing value for each task is highlighted in bold in Table 2.

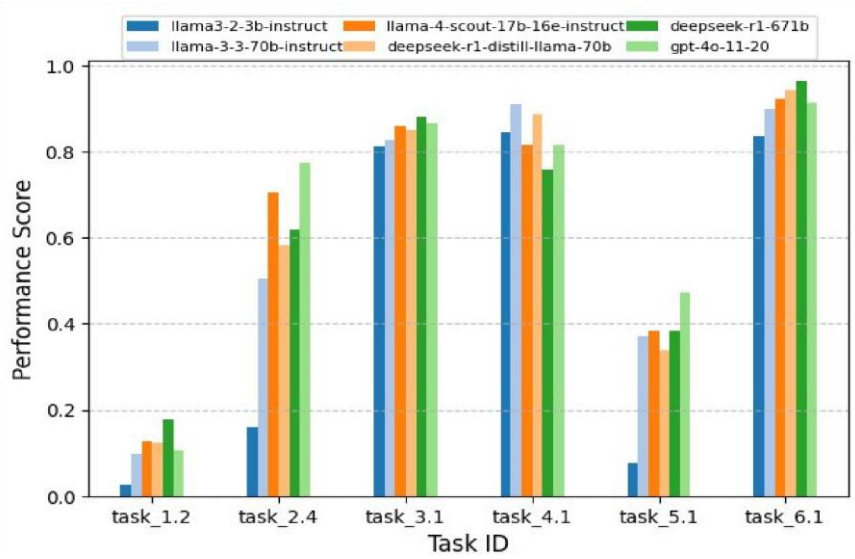


Figure 5. LLMs performance comparison on our selected tasks

Our results reveal several significant findings about the performance of proprietary versus opensource models, as well as insights into specific cognitive tasks.

**Overall Performance:** The benchmark tasks effectively challenge the models and highlight areas where even advanced models struggle. Notably, all models demonstrate difficulty with tasks like acronym explanation, achieving a maximum correctness score of only 0.20. This suggests a substantial gap in proprietary knowledge acquisition among these models.

**Proprietary vs. Open-Source Models:** The emergence of open-source models such as DeepSeek R1 is closing the gap between closed and open models. In our benchmarks, open-source models outperformed proprietary ones with eight wins, one tie, and five losses. Remarkably, in Task 61 (Content Generation), DeepSeek R1 surpassed all other models by a significant margin.

**LLMs With and Without Reasoning Capability:** The DeepSeek distilled reasoning model (Llama 70B) significantly outperforms Meta's non-reasoning one, achieving 11 wins, one tie, and two losses. For instance, in Task 2-3 (Sentiment Analysis), the reasoning model attained an accuracy of 84.3%, greatly outperforming the non-reasoning model's 74.2%. Reasoning models also demonstrated superior performance in enterprise-related questions (Tasks 1-2, 2-5, 3-2, 3-3).

**Task Performance Across Cognitive Levels:**

- **Remember**: All seven models perform poorly on AM and Factual Question-Answering tasks with correctness scores below 0.3, indicating insufficient proprietary enterprise knowledge.

- **Understand**: The toxic and bias metric values are notably low, as the testing cases were collected in an enterprise environment where ethical rules are strictly followed. Open-source models perform comparably to proprietary models in toxicity, bias evaluation, and questionanswering tasks. However, proprietary models outperform open-source models in SA and NER tasks, achieving higher scores in these areas.

- **Apply**: While all models demonstrate strong performance in summarization tasks, with relevance scores exceeding 0.8, there is substantial room for improvement in applicationbased tasks such as SWE, MLE, and NL2JQL.

- **Analyze**: All models underperform in this category due to high hallucination rates.

- **Evaluate**: In the LLM-as-a-Judge task, GPT-4o achieves the highest Spearman's r correlation at 0.47. However, reasoning models show weaker performance here, potentially due to overthinking [93].

- **Create**: All evaluated models perform well in content generation tasks, with coherence scores ranging from 0.84 to 0.97, demonstrating their capability in generating coherent and contextually appropriate text.

In summary, Llama-3.2-3B-Instruct shows the weakest overall performance among the six evaluated models and is not recommended for further post-training development. Most models excel primarily in toxicity evaluation, bias evaluation, summarization, and content generation tasks. However, all models lack sufficient enterprise-specific knowledge, underscoring the importance of continuous pre-training with enterprise domain-specific data to improve their performance. DeepSeek R1 emerges as a top performer in summarization and content generation, proving to be a valuable tool for enhancing reasoning capabilities through data synthesis and knowledge distillation.

Table 2. Evaluation results.

| Task | Metrics | Llama -3.2-3B | Llama -3.3-70B | Llama -4scout | Distilled Llama3.3-70B | DeepSeek-R1 | GPT-4o-2024-11-20 |
|---|---|---|---|---|---|---|---|
| 1-1. AM | G-Eval (correctness) | 0.1 | **0.20** | 0.18 | 0.19 | 0.19 | 0.16 |
| 1-2 Factual QA | G-Eval (correctness) | 0.03 | 0.10 | 0.13 | 0.12 | **0.18** | 0.11 |
| 2-1. Toxicity | ToxicMetric | 0.06 | **0.0** | 0.002 | **0.0** | 0.001 | **0.0** |
| 2-2. Bias | BiasMetric | 0.113 | 0.002 | 0.003 | **0.0** | 0.003 | 0.004 |
| 2-3. SA | Accuracy | 70.9% | 72.1% | 90.3% | 84.3% | 90% | **97.2%** |
| 2-4. NER | Exact Match | 16% | 50.6% | 70.5% | 58.4% | 62% | **77.4%** |
| 2-5. QA | G-Eval (correctness) | 0.22 | 0.32 | 0.33 | 0.33 | **0.38** | 0.3 |
| 3-1. Summarization | G-Eval (relevance) | 0.81 | 0.83 | 0.86 | 0.85 | **0.88** | 0.87 |
| 3-2. SWE | G-Eval (correctness) | 0.1 | 0.11 | 0.28 | 0.27 | **0.30** | 0.21 |
| 3-3. MLE | G-Eval (correctness) | 0.09 | 0.19 | **0.23** | 0.18 | 0.18 | 0.11 |
| 3-4. NL2JQL | Accuracy | 8.9% | 47.5% | 49% | 48.5% | 43.1% | **54%** |
| 4-1. HD | Hallucination Percentage | 84.5% | 91.1% | 81.7% | 88.6% | **75.8%** | 81.7% |
| 5-1. LLM-asa-Judge | Spearman's r | 0.08 | 0.37 | 0.38 | 0.34 | 0.38 | **0.47** |
| 6-1. Content Generation | G-Eval (coherence) | 0.84 | 0.9 | 0.92 | 0.94 | **0.97** | 0.91 |

## 5. CONCLUSIONS

This paper presents an enterprise benchmark grounded in Bloom's taxonomy, featuring 14 tasks across six cognitive levels. Evaluating six state-of-the-art LLMs reveals critical gaps in handling enterprise-specific content, emphasizing the importance of post-training on proprietary data. To ease the labor-intensive labeling process, we introduced a scalable annotation pipeline leveraging advanced LLM techniques, including LLM-as-a-Labeler, CRAG, and LLM-as-a-Judge. By sharing this benchmark, we provide a practical blueprint for enterprises and hope to inspire further innovation in tailoring LLMs to specialized business needs.

## ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our colleagues for their invaluable contributions to this work. Special thanks go to Hai Huang, Zahra Ghafoori, Matt Turner, Vincent Zeng, Amit Abbi, Stephan Curiskis, Longfei Zhang, Kevin Zhao, Steven Yoo, Zhaohui Wang, Allen Li, Prasad Marne, William Li, and Utkarsh Srivastava. Their assistance with data and the machine learning platform, as well as their insightful discussions and feedback, were instrumental in advancing this research. The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of Atlassian Corporation.

## REFERENCES

[1] X. Li, Z. Liu, and T. Hashimoto, "AutoBencher: Creating Salient, Novel, Difficult Datasets for Language Models," Jul. 11, 2024, arXiv: arXiv:2407.08351. Accessed: Oct. 04, 2024. [Online]. Available: http://arxiv.org/abs/2407.08351

[2] D. Hendrycks et al., "Measuring Massive Multitask Language Understanding," Jan. 12, 2021, arXiv: arXiv:2009.03300. doi: 10.48550/arXiv.2009.03300.

[3] A. Srivastava et al., "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models," Jun. 12, 2023, arXiv: arXiv:2206.04615. doi: 10.48550/arXiv.2206.04615.

[4] F. Chollet, "On the Measure of Intelligence," Nov. 25, 2019, arXiv: arXiv:1911.01547. doi: 10.48550/arXiv.1911.01547.

[5] P. Liang et al., "Holistic Evaluation of Language Models," Oct. 01, 2023, arXiv: arXiv:2211.09110. Accessed: Oct. 06, 2024. [Online]. Available: http://arxiv.org/abs/2211.09110

[6] L. Zheng et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," Dec. 24, 2023, arXiv: arXiv:2306.05685. Accessed: Oct. 16, 2024. [Online]. Available: http://arxiv.org/abs/2306.05685

[7] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What Disease does this Patient Have? A Large-scale Open Domain Question Answering Dataset from Medical Exams," Sep. 28, 2020, arXiv: arXiv:2009.13081. doi: 10.48550/arXiv.2009.13081.

[8] Z. Yao et al., "MedQA-CS: Benchmarking Large Language Models Clinical Skills Using an AISCE Framework," Oct. 02, 2024, arXiv: arXiv:2410.01553. doi: 10.48550/arXiv.2410.01553.

[9] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "MedMCQA : A Large-scale Multi-Subject MultiChoice Dataset for Medical domain Question Answering," Mar. 27, 2022, arXiv: arXiv:2203.14371. doi: 10.48550/arXiv.2203.14371.

[10] J. Matos et al., "WorldMedQA-V: a multilingual, multimodal medical examination dataset for multimodal language models evaluation," Oct. 16, 2024, arXiv: arXiv:2410.12722. doi 10.48550/arXiv.2410.12722.

[11] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu, "PubMedQA: A Dataset for Biomedical Research Question Answering," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2567–2577. doi: 10.18653/v1/D19-1259.

[12] X. Wang et al., "CMB: A Comprehensive Medical Benchmark in Chinese," Apr. 04, 2024, arXiv: arXiv:2308.08833. doi: 10.48550/arXiv.2308.08833.

[13] Q. Xie et al., "The FinBen: An Holistic Financial Benchmark for Large Language Models," Feb. 20, 2024, arXiv: arXiv:2402.12659. doi: 10.48550/arXiv.2402.12659.

[14] X. Guo et al., "FinEval: A Chinese Financial Domain Knowledge Evaluation Benchmark for Large Language Models," Dec. 08, 2024, arXiv: arXiv:2308.09975. doi: 10.48550/arXiv.2308.09975.

[15] R. S. Shah et al., "WHEN FLUE MEETS FLANG: Benchmarks and Large Pre-trained Language Model for Financial Domain," Oct. 31, 2022, arXiv: arXiv:2211.00083. doi: 10.48550/arXiv.2211.00083.

[16] D. Lu et al., "BBT-Fin: Comprehensive Construction of Chinese Financial Domain Pre-trained Language Model, Corpus and Benchmark," Feb. 26, 2023, arXiv: arXiv:2302.09432. doi: 10.48550/arXiv.2302.09432.

[17] X. Zhang, Q. Yang, and D. Xu, "XuanYuan 2.0: A Large Chinese Financial Chat Model with Hundreds of Billions Parameters," May 19, 2023, arXiv: arXiv:2305.12002. doi: 10.48550/arXiv.2305.12002.

[18] Q. Xie et al., "PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance," Jun. 08, 2023, arXiv: arXiv:2306.05443. doi: 10.48550/arXiv.2306.05443.

[19] I. Chalkidis et al., "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4310–4330. doi: 10.18653/v1/2022.acl-long.297.

[20] W. Hwang, D. Lee, K. Cho, H. Lee, and M. Seo, "A Multi-Task Benchmark for Korean Legal Language Understanding and Judgement Prediction," Oct. 05, 2022, arXiv: arXiv:2206.05224. doi: 10.48550/arXiv.2206.05224.

[21] N. Guha et al., "LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models," Aug. 20, 2023, arXiv: arXiv:2308.11462. doi: 10.48550/arXiv.2308.11462.

[22] Z. Fei et al., "LawBench: Benchmarking Legal Knowledge of Large Language Models," Sep. 28, 2023, arXiv: arXiv:2309.16289. Accessed: Oct. 20, 2024. [Online]. Available: http://arxiv.org/abs/2309.16289

[23] H. Li et al., "CMMLU: Measuring massive multitask language understanding in Chinese," Jan. 17, 2024, arXiv: arXiv:2306.09212. doi: 10.48550/arXiv.2306.09212.

[24] X. Zhang, C. Li, Y. Zong, Z. Ying, L. He, and X. Qiu, "Evaluating the Performance of Large Language Models on GAOKAO Benchmark," Feb. 24, 2024, arXiv: arXiv:2305.12474. doi: 10.48550/arXiv.2305.12474.

[25] Y. Huang et al., "C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models," Nov. 06, 2023, arXiv: arXiv:2305.08322. doi: 10.48550/arXiv.2305.08322.

[26] O. Ghahroodi et al., "Khayyam Challenge (PersianMMLU): Is Your LLM Truly Wise to The Persian Language?," 2024.

[27] W. Zhang, S. M. Aljunied, C. Gao, Y. K. Chia, and L. Bing, "M3Exam: A Multilingual, Multimodal, Multilevel Benchmark for Examining Large Language Models," Nov. 10, 2023, arXiv: arXiv:2306.05179. doi: 10.48550/arXiv.2306.05179.

[28] W. Zhong et al., "AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models," Sep. 18, 2023, arXiv: arXiv:2304.06364. doi: 10.48550/arXiv.2304.06364.

[29] Z. Tan et al., "Large Language Models for Data Annotation and Synthesis: A Survey," Dec. 02, 2024, arXiv: arXiv:2402.13446. doi: 10.48550/arXiv.2402.13446.

[30] L. Long et al., "On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey," Jun. 14, 2024, arXiv: arXiv:2406.15126. doi: 10.48550/arXiv.2406.15126.

[31] A. Bauer et al., "Comprehensive Exploration of Synthetic Data Generation: A Survey," Feb. 01, 2024, arXiv: arXiv:2401.02524. doi: 10.48550/arXiv.2401.02524.

[32] R. Liu et al., "Best Practices and Lessons Learned on Synthetic Data," Aug. 10, 2024, arXiv: arXiv:2404.07503. doi: 10.48550/arXiv.2404.07503.

[33] X. Wang et al., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," Mar. 07, 2023, arXiv: arXiv:2203.11171. Accessed: Oct. 29, 2024. [Online]. Available: http://arxiv.org/abs/2203.11171

[34] J. Li, J. Wang, Z. Zhang, and H. Zhao, "Self-Prompting Large Language Models for Zero-Shot Open-Domain QA," Mar. 28, 2024, arXiv: arXiv:2212.08635. doi: 10.48550/arXiv.2212.08635.

[35] DeepSeek-AI et al., "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," Jan. 22, 2025, arXiv: arXiv:2501.12948. doi: 10.48550/arXiv.2501.12948.

[36] HuggingFace, Open-R1. (2025). [Online]. Available: https://github.com/huggingface/open-r1

[37] Y. Liang et al., "I-SHEEP: Self-Alignment of LLM from Scratch through an Iterative SelfEnhancement Paradigm," Aug. 27, 2024, arXiv: arXiv:2408.08072. doi: 10.48550/arXiv.2408.08072.

[38] W. Yuan et al., "Self-Rewarding Language Models," Feb. 08, 2024, arXiv: arXiv:2401.10020. doi: 10.48550/arXiv.2401.10020.

[39] A. Pace, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, "West-of-N: Synthetic Preferences for Self-Improving Reward Models," Oct. 25, 2024, arXiv: arXiv:2401.12086. doi: 10.48550/arXiv.2401.12086.

[40] T. Wu et al., "Meta-Rewarding Language Models: Self-Improving Alignment with LLM-as-aMeta-Judge," Jul. 29, 2024, arXiv: arXiv:2407.19594. Accessed: Sep. 30, 2024. [Online]. Available: http://arxiv.org/abs/2407.19594

[41] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang, "Automatically Correcting Large Language Models: Surveying the landscape of diverse self-correction strategies," Aug. 30, 2023, arXiv: arXiv:2308.03188. doi: 10.48550/arXiv.2308.03188.

[42] J. Robinson, C. M. Rytting, and D. Wingate, "Leveraging Large Language Models for Multiple Choice Question Answering," Mar. 17, 2023, arXiv: arXiv:2210.12353. doi: 10.48550/arXiv.2210.12353.

[43] S. Lin, J. Hilton, and O. Evans, "Teaching Models to Express Their Uncertainty in Words," Jun. 13, 2022, arXiv: arXiv:2205.14334. doi: 10.48550/arXiv.2205.14334.

[44] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2020, pp. 9459–9474. Accessed: Feb. 05, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/6b493230205f780e1bc26945df7481e5Abstract.html

[45] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely," Sep. 23, 2024, arXiv: arXiv:2409.14924. doi: 10.48550/arXiv.2409.14924.

[46] H. Qian, P. Zhang, Z. Liu, K. Mao, and Z. Dou, "MemoRAG: Moving towards Next-Gen RAG Via Memory-Inspired Knowledge Discovery," Sep. 10, 2024, arXiv: arXiv:2409.05591. doi: 10.48550/arXiv.2409.05591.

[47] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," Mar. 28, 2024, arXiv: arXiv:2403.14403. doi: 10.48550/arXiv.2403.14403.

[48] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," Oct. 17, 2023, arXiv: arXiv:2310.11511. doi: 10.48550/arXiv.2310.11511.

[49] Y. Wang, P. Li, M. Sun, and Y. Liu, "Self-Knowledge Guided Retrieval Augmentation for Large Language Models," Oct. 08, 2023, arXiv: arXiv:2310.05002. doi: 10.48550/arXiv.2310.05002.

[50] O. Yoran, T. Wolfson, O. Ram, and J. Berant, "Making Retrieval-Augmented Language Models Robust to Irrelevant Context," May 05, 2024, arXiv: arXiv:2310.01558. doi: 10.48550/arXiv.2310.01558.

[51] H. Luo et al., "Search Augmented Instruction Learning," in Findings of the Association for Computational Linguistics: EMNLP 2023, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3717–3729. doi: 10.18653/v1/2023.findings-emnlp.242.

[52] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," Oct. 07, 2024, arXiv: arXiv:2401.15884. doi: 10.48550/arXiv.2401.15884.

[53] R. Raju, S. Jain, B. Li, J. Li, and U. Thakker, "Constructing Domain-Specific Evaluation Sets for LLM-as-a-judge," Aug. 20, 2024, arXiv: arXiv:2408.08808. doi: 10.48550/arXiv.2408.08808.

[54] H. Lee, "LLM-as-a-Judge: Rethinking Model-Based Evaluations in Text Generation." Accessed: Jan. 18, 2025. [Online]. Available: https://leehanchung.github.io/blogs/2024/08/11/llm-as-a-judge/

[55] E. Yan, "Evaluating the Effectiveness of LLM-Evaluators (aka LLM-as-Judge)," eugeneyan.com. Accessed: Oct. 12, 2024. [Online]. Available: https://eugeneyan.com/writing/llm-evaluators/

[56] D. Li et al., "From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge," Nov. 25, 2024, arXiv: arXiv:2411.16594. doi: 10.48550/arXiv.2411.16594.

[57]. Aman Yugank, "Deep Dive into LLM-evaluators aka 'LLM-as-a-Judge,'" Medium. Accessed: Dec. 19, 2024. [Online]. Available: https://medium.com/@yugank.aman/deep-dive-into-llm-evaluatorsaka-llm-as-a-judge-c34f497edca9

[58] A. S. Thakur, K. Choudhary, V. S. Ramayapally, S. Vaidyanathan, and D. Hupkes, "Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges," Oct. 11, 2024, arXiv: arXiv:2406.12624. Accessed: Oct. 16, 2024. [Online]. Available: http://arxiv.org/abs/2406.12624

[59] Y. Wang et al., "PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization," May 24, 2024, arXiv: arXiv:2306.05087. Accessed: Sep. 22, 2024. [Online]. Available: http://arxiv.org/abs/2306.05087

[60] Z. Zeng, J. Yu, T. Gao, Y. Meng, T. Goyal, and D. Chen, "Evaluating Large Language Models at EvaluatingInstruction Following," Apr. 16, 2024, arXiv: arXiv:2310.07641. doi: 10.48550/arXiv.2310.07641.

[61] P. Wang et al., "Large Language Models are not Fair Evaluators," Aug. 30, 2023, arXiv: arXiv:2305.17926. doi: 10.48550/arXiv.2305.17926.

[62] Q. Zhang et al., "RevisEval: Improving LLM-as-a-Judge via Response-Adapted References," Oct. 07, 2024, arXiv: arXiv:2410.05193. doi: 10.48550/arXiv.2410.05193.

[63] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," May 23, 2023, arXiv: arXiv:2303.16634. Accessed: Oct. 15, 2024. [Online]. Available: http://arxiv.org/abs/2303.16634

[64] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Jan. 10, 2023, arXiv: arXiv:2201.11903. doi: 10.48550/arXiv.2201.11903.

[65] Y. Chang et al., "A Survey on Evaluation of Large Language Models," Dec. 28, 2023, arXiv: arXiv:2307.03109. doi: 10.48550/arXiv.2307.03109.

[66] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," Theory Into Practice, vol. 41, no. 4, pp. 212–218, 2002.

[67] P. Armstrong, "Bloom's Taxonomy," Vanderbilt University. Accessed: Dec. 29, 2024. [Online]. Available: https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

[68] Atlassian, "Confluence." [Online]. Available: https://www.atlassian.com/software/confluence/resources

[69] Atlassian, Rovo Chat. (2024). Atlassian. [Online]. Available: https://www.atlassian.com/software/rovo

[70] SBERT, "Paraphrase Mining — Sentence Transformers documentation." Accessed: Dec. 29, 2024. [Online]. Available: https://sbert.net/examples/applications/paraphrase-mining/README.html

[71] J. Achiam et al., "GPT-4 Technical Report," Mar. 04, 2024, arXiv: arXiv:2303.08774. doi: 10.48550/arXiv.2303.08774.

[72] Confidential AI, DeepEval. (2024). Confident AI.

[73] J. Ip, "LLM Evaluation Metrics: The Ultimate LLM Evaluation Guide - Confident AI." Accessed: Dec. 29, 2024. [Online]. Available: https://www.confident-ai.com/blog/llm-evaluation-metrics-everythingyou-need-for-llm-evaluation

[74] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, P. Isabelle, E. Charniak, and D. Lin, Eds., Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. doi: 10.3115/1073083.1073135.

[75] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in Text Summarization Branches Out, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. Accessed: Feb. 09, 2025. [Online]. Available: https://aclanthology.org/W04-1013/

[76] E. Sulem, O. Abend, and A. Rappoport, "BLEU is Not Suitable for the Evaluation of Text Simplification," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 738–744. doi: 10.18653/v1/D18-1081.

[77] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Feb. 24, 2020, arXiv: arXiv:1904.09675. doi: 10.48550/arXiv.1904.09675.

[78] W. Yuan, G. Neubig, and P. Liu, "BARTScore: Evaluating Generated Text as Text Generation," Oct. 27, 2021, arXiv: arXiv:2106.11520. doi: 10.48550/arXiv.2106.11520.

[79] M. Gao, X. Hu, J. Ruan, X. Pu, and X. Wan, "LLM-based NLG Evaluation: Current Status and Challenges," Feb. 26, 2024, arXiv: arXiv:2402.01383. doi: 10.48550/arXiv.2402.01383.

[80] P. Törnberg, "ChatGPT-4 Outperforms Experts and Crowd Workers in Annotating Political Twitter Messages with Zero-Shot Learning," Apr. 13, 2023, arXiv: arXiv:2304.06588. doi: 10.48550/arXiv.2304.06588.

[81] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT Outperforms Crowd-Workers for TextAnnotation Tasks," Proc. Natl. Acad. Sci. U.S.A., vol. 120, no. 30, p. e2305016120, Jul. 2023, doi: 10.1073/pnas.2305016120.

[82] L. Ostyakova, V. Smilga, K. Petukhova, M. Molchanova, and D. Kornev, "ChatGPT vs. Crowdsourcing vs. Experts: Annotating Open-Domain Conversations with Speech Functions," in Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue, S. Stoyanchev, S. Joty, D. Schlangen, O. Dusek, C. Kennington, and M. Alikhani, Eds., Prague, Czechia: Association for Computational Linguistics, Sep. 2023, pp. 242–254. doi: 10.18653/v1/2023.sigdial-1.23.

[83] J. Cegin, J. Simko, and P. Brusilovsky, "ChatGPT to Replace Crowdsourcing of Paraphrases for Intent Classification: Higher Diversity and Comparable Model Robustness," Oct. 19, 2023, arXiv: arXiv:2305.12947. doi: 10.48550/arXiv.2305.12947.

[84] DeepEval, "G-Eval." Accessed: Dec. 30, 2024. [Online]. Available: https://docs.confidentai.com/docs/metrics-llm-evals

[85] DeepEval, "Toxicity." Accessed: Dec. 29, 2024. [Online]. Available: https://docs.confidentai.com/docs/metrics-toxicity

[86] DeepEval, "Bias." Accessed: Dec. 29, 2024. [Online]. Available: https://docs.confidentai.com/docs/metrics-bias

[87] DeepEval, "Hallucination." Accessed: Dec. 29, 2024. [Online]. Available: https://docs.confidentai.com/docs/metrics-hallucination

[88] Wikipedia, "Spearman's rank correlation coefficient," Wikipedia. Dec. 27, 2024. Accessed: Feb. 09, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Spearman%27s_rank_correlation_coefficient&oldid=1265551 920

[89] Meta, "Llama 3.2: Revolutionizing edge AI and vision with open, customizable models," Meta AI. Accessed: Feb. 02, 2025. [Online]. Available: https://ai.meta.com/blog/llama-3-2-connect-2024vision-edge-mobile-devices/

[90] Meta, "LLaMA 3.3," LLaMA 3.3 Document. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

[91] Meta, "The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation," Meta AI. Accessed: Jun. 05, 2025. [Online]. Available: https://ai.meta.com/blog/llama-4-multimodalintelligence/

[92] A. Yang et al., "Qwen2.5 Technical Report," Dec. 19, 2024, arXiv: arXiv:2412.15115. doi: 10.48550/arXiv.2412.15115.

[93] A. Cuadron et al., "The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks," Feb. 12, 2025, arXiv: arXiv:2502.08235. doi: 10.48550/arXiv.2502.08235.

[94] LangGraph, "CRAG." [Online]. Available: https://github.com/langchainai/langgraph/blob/main/examples/rag/langgraph_crag.ipynb?ref=blog.langchain.dev

## APPENDIX A: ALGORITHMS USED IN THE WORK

### CRAG

The corrective retrieval augmented generation (CRAG) [52] algorithm, depicted in Figure 6, involves the following steps:

- Question Input: The process starts with a user query.
- Retrieve: Relevant documents or passages are extracted from a knowledge base in response to the query.
- Grade: The relevance of these documents is assessed to ensure only pertinent information is used.
- Irrelevancy Check: The system checks for any irrelevant documents.
- Answer Generation (If Relevant): If all documents are relevant, an answer is generated from the graded information.
- Query Rewriting (If Irrelevant): If there are irrelevant documents, the query is refined for clarity.
- Web Search: The refined query is used to conduct a web search to find additional accurate information.
- Final Answer Generation: A comprehensive answer is crafted using insights from both the initial and web-searched documents.

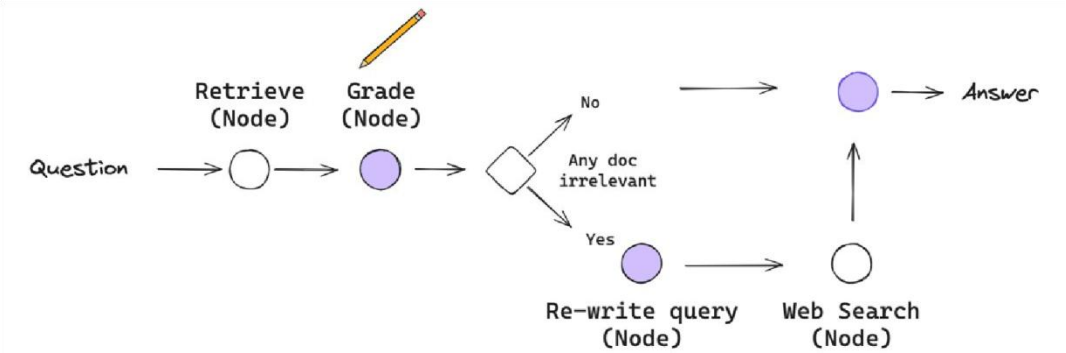


Figure 6. CRAG process [94]

## G-Eval

As illustrated in Figure 7, the G-Eval framework utilizes LLMs with CoT reasoning and a formfilling approach to assess output quality. The evaluation process involves three key components: 1) a prompt outlining the evaluation task and the specific criteria to be used, 2) a LLM call to generate CoT consisting of intermediate instructions that detail the evaluation steps, and 3) a scoring function that utilizes the LLM to compute a score based on the probabilities of the returned tokens. This method has been incorporated into the DeepEval package [72].

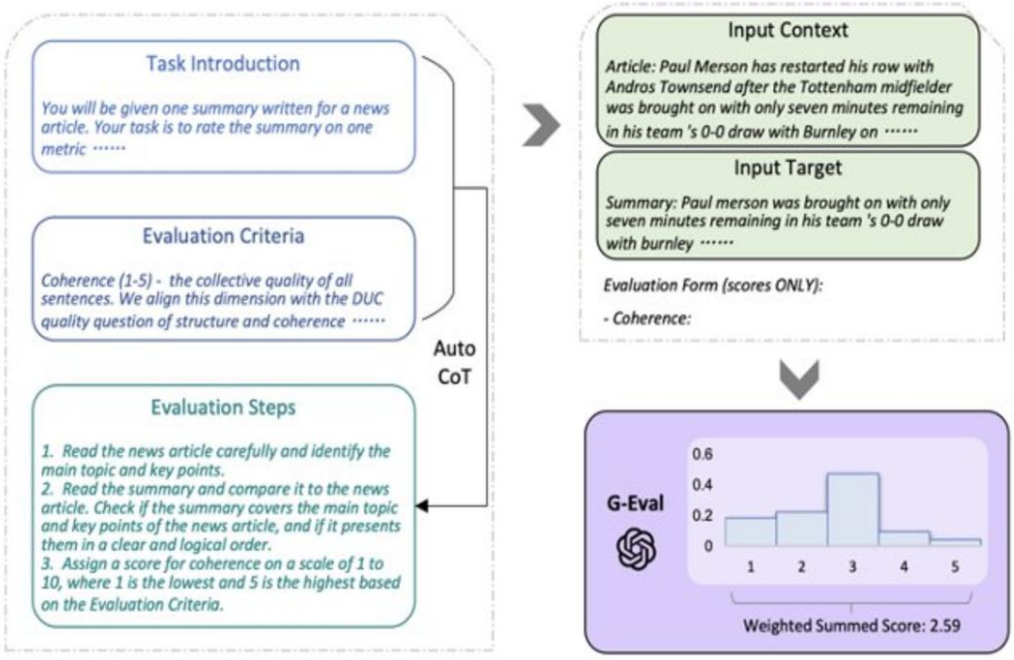


Figure 7. The framework of G-Eval [79]

©2025 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.