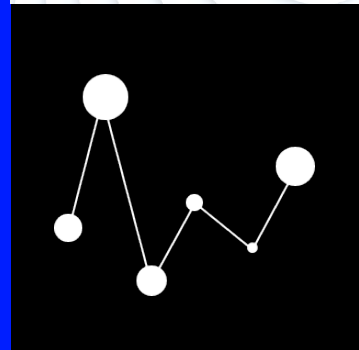
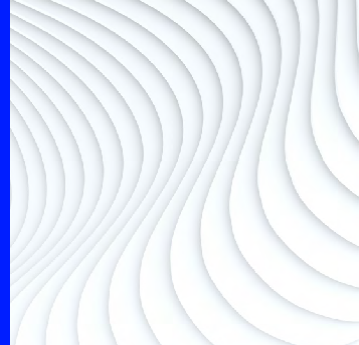


# Judging AI with AI

LLM-as-a-Judge: Using AI to Evaluate GenAI Output



**LSEG** DATA & ANALYTICS



---

LSEG Analytics is actively investigating novel approaches to enhance the next generation of LSEG AI products. This primer outlines LLM-as-a-Judge, a technique that leverages Large Language Models to evaluate AI-generated content. By repurposing Large Language Models as evaluators, this approach offers a scalable solution to one of the most pressing challenges in generative AI: maintaining quality at scale.

---

Large Language Models (LLMs) have surged in popularity with their ability to generate vast amounts of text, which, at least at first glance, appear coherent and factually accurate. However, with such an overwhelming volume of content being created, a critical challenge emerges: *how can we reliably assess the quality of these outputs?*

The idea of using LLMs to evaluate the content they generate is often met with scepticism, reflecting a broader concern around the reliability, autonomy, and transparency of AI systems.

To address these concerns this paper explores the concept of **LLM-as-a-Judge**, an approach that leverages LLMs to automatically assess the quality of generated AI text. Our goal is to demystify this technique and demonstrate its potential as a scalable, accessible, and effective tool for evaluation.

## Why we need to consider automated evaluation

While evaluating text is a well-established practice in Natural Language Processing (NLP), the scale and complexity introduced by generative AI have rendered traditional approaches increasingly inadequate:

- Manual annotation by Subject Matter Experts is resource-intensive and difficult to scale. In the financial industry, subject matter experts time is already stretched across critical functions. Automated evaluation via *LLM-as-a-Judge* offers an on-demand alternative that reduces dependency on human coordination.
- Traditional NLP evaluation metrics that are lexical based, such as BLEU and ROUGE, rely on word-level overlap that is poorly suited to the flexible, semantically rich outputs of modern LLMs. More advanced metrics like BERTScore can fall short when nuanced judgment is required [1]. Automated evaluation using LLMs enables more context-aware, meaning-driven assessments that better reflect human expectations of quality.

Furthermore, automated evaluation will provide significant advantages as AI systems scale in complexity, for example integration into CI/CD pipelines, supporting regression testing and providing a historical record of model behaviour tied to specific configurations. As AI systems become more modular and agent-based, this kind of traceable, automated evaluation will likely become essential for maintaining trust and accountability.

## AUTHORS

**Stanislav Chistyakov**  
Data Scientist

**Dinesh Kalamegam**  
AI Engineer, LSEG  
Analytics

**David Oliver**  
Director, Data Science

*To find out more about this paper or for other enquiries related to AI Research, please contact:*

**Helen Zhang**,  
Head of Quantitative  
Solutions Research  
[helen.zhang@lseg.com](mailto:helen.zhang@lseg.com)

## LLM-as-a-Judge

To assist with this large-scale automatic evaluation, it is natural to consider how to use AI to evaluate generative AI output. This paradigm is referred to as *LLM-as-a-Judge*<sup>1</sup>, being the use of an LLM to evaluate the output generated by another LLM.

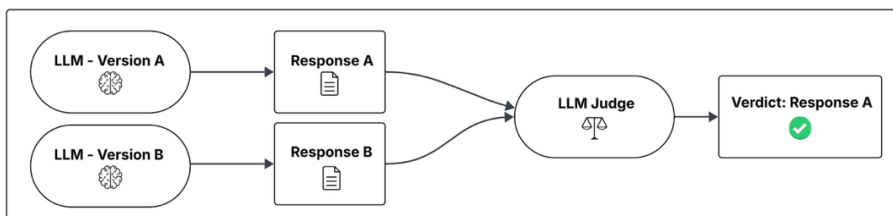
To help clarify the concept of *LLM-as-a-Judge*, it's important to note that the evaluator model need not be distinct from the one generating the content — it can be the same LLM. There's nothing inherently special about the model used for evaluation. Moreover, the techniques outlined in this paper are vendor-neutral, making them broadly applicable across different LLMs.

The implementation of *LLM-as-a-Judge* is very similar to most current AI applications; there is a prompt, or series of prompts, being a set of instructions for the LLM on the task to perform. In the case of LLM-of-a-Judge these are instructions on how to perform the evaluation required [2]. This prompt is specifically designed to evaluate the output of another LLM, with a focus on the expected output, this allows for domain knowledge and expertise to be included in the evaluation.

To illustrate the concept of *LLM-as-a-Judge*, we focus on two commonly used paradigms: **pairwise comparison** and **direct single-response scoring**. While there are many ways to implement LLM-based evaluation, these two approaches are among the most intuitive and align closely with how humans would typically do a similar assessment. **Pairwise comparison** involves evaluating two outputs side-by-side to determine which is better, while **direct single-response scoring** assigns a rating to an individual output, such as 1-to-5 stars.

### LLM-as-a-Judge: Pairwise Comparison

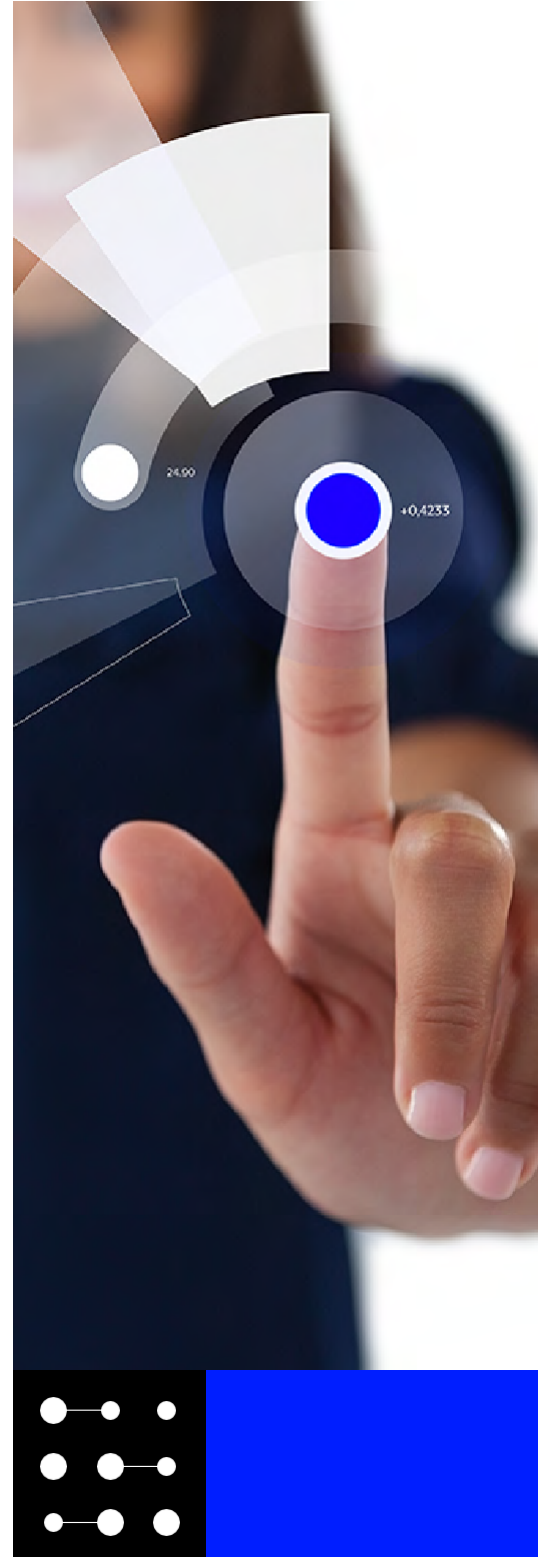
In **pairwise comparison** the *LLM-as-a-Judge* selects the better of the two responses generated by the application.



**Figure 1: Pairwise Comparison, two responses are generated, and the LLM-as-a-Judge selecting the one deemed superior based on predefined criteria.**

A good example of when pairwise comparison is suitable is in evaluating an LLM upgrade, to confirm a new LLM improves over the previous model; in this case the two responses come from two different LLMs versions, for example OpenAI's GPT-5 and GPT-4o.

Note that human pairwise comparison is already very popular with AI, most notably in the LM Arena, an open evaluation platform by UC Berkeley, that uses pairwise comparisons to rank large language models based on human preferences .



<sup>1</sup> The term *LLM-as-a-Judge* was coined by the Chatbot Arena (now known as the LM Arena) [2]



Figure 2 displays simple-but-effective prompt that can be used to create a pairwise comparison LLM evaluator; this prompt is written by the creators of the Chatbot Arena (now renamed the LM Arena) [2]. The LLM evaluator is asked to state which of the two responses is superior and in ambiguous cases, to simply confirm a tie. The language used in this prompt is simple to understand, demonstrating that complexity is not required for *LLM-as-a-Judge*.

```
[System]
Please act as an impartial judge and evaluate the quality of the responses provided by two
AI assistants to the user question displayed below. You should choose the assistant that
follows the user's instructions and answers the user's question better. Your evaluation
should consider factors such as the helpfulness, relevance, accuracy, depth, creativity,
and level of detail of their responses. Begin your evaluation by comparing the two
responses and provide a short explanation. Avoid any position biases and ensure that the
order in which the responses were presented does not influence your decision. Do not allow
the length of the responses to influence your evaluation. Do not favor certain names of
the assistants. Be as objective as possible. After providing your explanation, output your
final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]"
if assistant B is better, and "[[C]]" for a tie.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

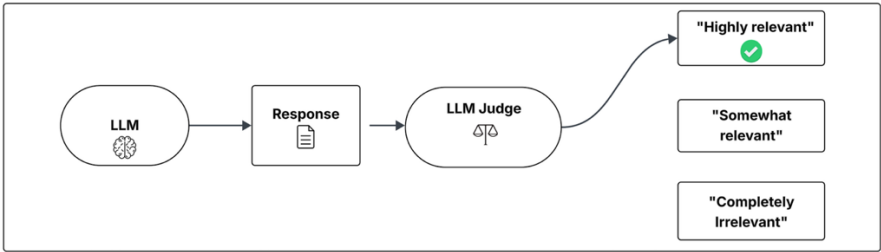
[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

**Figure 2: The default prompt for pairwise comparison proposed by the creators of Chatbot Arena, as shown in “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena” [2]. The instructions provided are clear and easy to understand, supporting consistent evaluation if they were to be also used with human annotators.**

The intuitive nature of pairwise comparison makes it easy to align *LLM-as-a-Judge* evaluations with human judgments. Because the instructions used in the prompt can be given to human annotators, near verbatim, this approach is particularly valuable when involving subject matter experts. Clear, familiar instructions significantly streamline the annotation process and improve consistency across evaluations.

**LLM-as-a-Judge: Direct single response scoring**

**Direct single-response scoring** refers to the use of LLM-as-a-Judge to evaluate a single generated output and produce a quantitative or qualitative metric for assessment. Unlike pairwise comparison, which requires two responses for relative evaluation, this method enables standalone judgment and represents a more advanced approach to automated evaluation.



**Figure 3: Direct single response scoring, where a single response is generated and the LLM-as-a-Judge produces a score align with a particular metric, in this case, relevance.**





Figure 4 provides an example prompt for direct single response scoring; the *LLM-as-a-Judge* is prompted to provide a numeric score from 1 to 3 to assess the “relevance” of the response to a user input question<sup>2</sup>.

You are an expert evaluator assessing how well a candidate response fits the context and intent of a given input.

Your task is to judge Relevance.

You will be given the following information:

1. A question
2. A model-generated answer

Your goal is to score how relevant, on-topic, and contextually appropriate the response is. Consider whether the response:

- Directly addresses the user's input
- Stays on-topic without introducing unrelated or unnecessary information
- Is contextually suitable (e.g., aligns with the tone, scope, and detail level of the input)
- A relevance score from 1 to 3:
  - 1 = completely off-topic
  - 2 = somewhat relevant response, but with certain parts of the response being irrelevant.
  - 3 = highly relevant and appropriate
- A brief justification explaining the score and specific points of alignment or mismatch with the input.

Today's date is {{ today\_date }}. Make sure you use today's date for reference when providing the score.

-----  
Question: {{ query }}

Model-generated answer: {{ model\_generated\_answer }}

-----  
Respond in JSON format.

Return relevance score and justification.

**Figure 4: Example LLM-as-a-Judge prompt for scoring the Relevance of the model response to the user query in a conversational chatbot scenario**

While the concept of **direct scoring** is intuitive, its implementation requires careful consideration. Evaluation criteria should be clearly defined and explicitly stated within the prompt. Moreover, the metrics used are likely needed to be tailored to the specific use case; given the variety of LLM-generated content, there are no universally accepted industry standards for evaluation. Three dimensions that serve as a decent practical starting point are; **relevance**, **faithfulness**, and **correctness**. **Relevance** assesses how well the response aligns with the user query; **faithfulness** evaluates whether the response is grounded in the provided context and free from hallucinations; and **correctness** measures factual accuracy.

Reducing ambiguity is essential for reliable evaluation. Generally, if a subject matter expert cannot confidently distinguish between scoring levels, such as assigning a “2” versus a “3” for relevance, then an LLM is unlikely to perform better. Additionally, providing contextual information significantly enhances the LLM’s ability to evaluate responses. This may include supplying the source material used to generate the response or offering an expected answer for comparison.

Finally, prompting the *LLM-as-a-Judge* to generate reasoning alongside its score can greatly improve transparency and explainability. Though simple, in practice this step adds valuable insight into the model’s decision-making process and strengthens trust in automated evaluation.

<sup>2</sup>The example prompt in Figure 4 is taken from a prototype developed by LSEG Analytics Research, an AI conversational chatbot.

## Evaluating LLM-as-a-Judge

A substantial body of literature already exists on evaluating *LLM-as-a-Judge*, our intention is not to replicate such a comprehensive review [4] [5]. Instead, we offer commentary on two pragmatic considerations: first, the importance of comparing human annotation with LLM-based evaluation, and second, how established techniques from machine learning can be adapted to assess the performance of LLM-as-a-Judge.

### Evaluating LLM-as-a-Judge: Comparison with Human Annotation

A central objective of the LLM-as-a-Judge paradigm is to achieve alignment with human judgment, and therefore **one of the most important evaluation approaches for LLM-as-a-Judge is comparison with human annotations.**

The creators of Chatbot Arena, who pioneered the use of LLM evaluators, reported higher than 80% agreement between human annotators and GPT-4, the state-of-the-art model at the time of their research [2]. Definitions of “agreement” can vary, but the researchers set it as *“the probability of randomly selected individuals (but not identical) of each type agreeing on a randomly selected question.”*

It is important to appreciate that achieving consistent human evaluation of LLM outputs can be inherently challenging. In human annotation, ambiguity may arise from the content itself, from differences in interpretation among annotators, or from unclear evaluation criteria. Multiple annotators are commonly used, and inter-rater consistency is an important metric. Quantitative measures such as Cohen’s kappa (for two raters) and *Fleiss’ kappa* (for multiple raters) are frequently used to evaluate this consistency.

These challenges are needed to be kept in mind when evaluating *LLM-as-a-Judge* approach. We stress that it is important to cross-reference at least a subset of *LLM-as-a-Judge* evaluations with human annotations. For instance, if human evaluators show low inter-rater agreement during a particular assessment, it may indicate that the evaluation criteria are unclear or insufficiently defined, suggesting the need for refinement before relying on LLM-as-a-Judge on a large scale.

### Evaluating LLM-as-a-Judge: Using common Machine Learning metrics

Existing evaluation techniques are well-suited for assessing *LLM-as-a-Judge*, and there’s no pressing need to develop entirely new methodologies. To demonstrate this, we apply well-established metrics from machine learning classification that are both familiar and effective in practice.

In the following example, the *LLM-as-a-Judge* was prompted to assess the relevance of one hundred chatbot responses to user queries, assigning scores of “1” (completely irrelevant), “2” (somewhat relevant), or “3” (highly relevant). A single subject matter expert provided the ground truth labels evaluation set. While relying on a single annotator does not account for potential bias, as discussed in the previous section, it offers a practical and efficient starting point for evaluation.

Insight into the limitations of the *LLM-as-a-Judge* performance can be observed via standard machine learning classification metrics, including **accuracy**, **precision**, **recall** and a **confusion matrix**, shown in Figure 5.

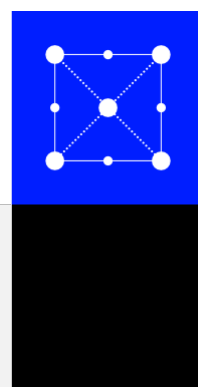
The *LLM-as-a-Judge* correctly recognised 27 out of 30 responses (90%) that were marked as “1” (completely irrelevant) and 44 out of 53 responses (83%) that were marked as “3” (highly relevant), indicating good recall for the 1 and 3 categories. However, it only identified 9 out of 17 responses that were marked as “2” (somewhat relevant), at a much lower recall of 53%.

Also, when it predicted “2”, in only 9 out of 16 cases this aligned with subject matter expert scoring, which corresponds to the precision of 56%. In comparison the precision 1 is 77% and 3 is 93%.

Our conclusion is that the *LLM-as-a-Judge* could not usefully categorise into “2” (somewhat relevant), and instead, seems to be preferring only 1 or 3.

To address these issues, we recommend either refining the prompt to ensure all three categories achieve reasonable precision and recall on a test set or simplifying the task by reducing the categories from three to two.

These results aim to show that *LLM-as-a-Judge* performance can be meaningfully evaluated using standard machine learning metrics and a small dataset labelled by a single subject matter expert. This kind of analysis can easily be extended beyond ML metrics to include a wide range of well-established statistical methods.



## Confusion Matrix: Relevance Score

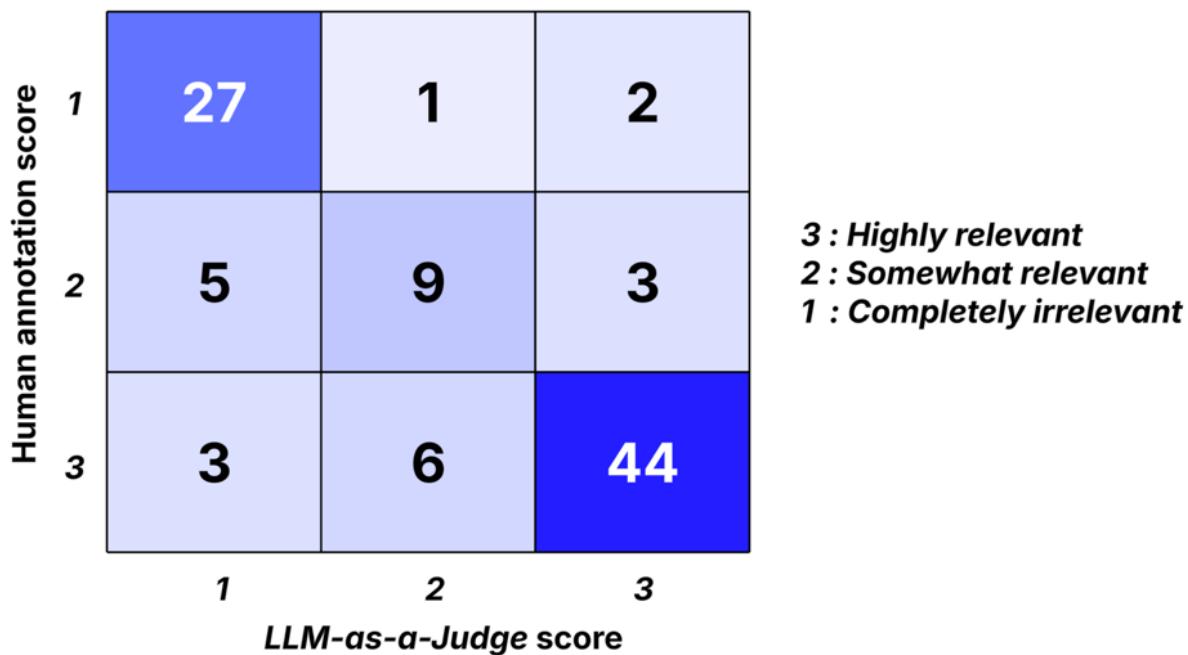


Figure 5: Confusion matrix, comprising of 100 examples evaluated by both a subject matter expert and an LLM-as-a-Judge.

## Limitations of LLM-as-a-Judge

While *LLM-as-a-Judge* presents a promising solution for scalable, automated evaluation of AI-generated content, it is not without its limitations. Three areas that merit closer examination are: the **continued need for subject matter expertise**, the **inherent biases** embedded within LLMs, and the challenges of **safeguarding proprietary data**.

### Limitations: The need for Subject Matter Expertise

While *LLM-as-a-Judge* presents a compelling solution for scalable evaluation, it does not eliminate the need for human expertise. In practice allocating subject matter experts to support evaluation remains a critical, although resource-intensive, component of a successful AI project.

Research from MIT suggests that LLM alignment with human judgment improves significantly when provided with reference responses [6]. This highlights a potential fundamental limitation: LLMs may struggle to evaluate content they themselves could not generate. In production environments, generating reliable reference responses is often impractical, reinforcing the importance of subject matter expertise involvement during pre-production.

### Limitations: Inherent bias of LLMs

Despite their sophistication, LLMs are not immune to bias, particularly when used as evaluators. Researchers have identified several systematic tendencies that can influence the reliability of *LLM-as-a-Judge*. One such example is **position bias**, where the model exhibits a preference based on the order in which evaluation criteria are presented within the prompt. Another notable issue is **self-preference bias**, where an LLM tends to favour responses generated by the same model used for evaluation. For instance, if OpenAI's GPT-4.1 is tasked with comparing its own output against that of another provider, such as Anthropic's Claude 4, it may consistently prefer its own responses. These biases underscore the importance of designing evaluation protocols with care and, where possible, incorporating cross-model comparisons and human oversight to mitigate skewed judgments [4].

### Limitations: Proprietary Data

A significant constraint in deploying *LLM-as-a-Judge* is the imperative to protect proprietary and sensitive data – a concern that is widespread within the financial industry. This requirement introduces practical challenges as *LLM-as-a-Judge* typically operates downstream of the primary AI application. As a result, it may only have access to a limited subset of the data used to generate the original response, impairing the *LLM-as-a-Judge* ability to accurately assess the output. Addressing this limitation requires thoughtful system design, where evaluation is not treated as an afterthought, but as an integral component of the AI development lifecycle, with appropriate data access and safeguards built in from the outset.



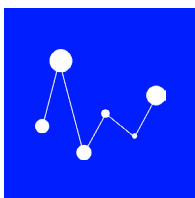


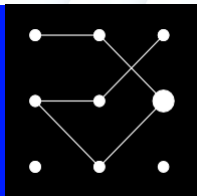
## Final Thoughts: LLM-as-a-Judge will help build trust in AI


As AI continues to gain traction, it's worth recalling a well-known aphorism familiar to most every statistician: *"All models are wrong, but some are useful,"* a sentiment attributed to George Box. Today, in the Financial Industry, "AI" is largely synonymous with Large Language **Models**, and this quote serves as a timely reminder of the inherent uncertainty that comes with using models. Statisticians have long accepted this uncertainty as part of the modelling process, however, as AI becomes more mainstream, this tolerance may not be as widely shared. To handle this uncertainty, users will need to trust AI; *LLM-as-a-Judge* offers the potential for the large-scale evaluation of AI required to build this trust.

## References

- [1] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," 24 February 2020. [Online]. Available: <https://arxiv.org/pdf/1904.09675>.
- [2] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez and I. Stoica, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," 24 December 2023. [Online]. Available: <https://arxiv.org/abs/2306.05685>.
- [3] LSEG Analytics, "Comparing Large Language Models: How to decide which is the best state-of-the-art model," January 2025. [Online]. Available: [https://www.lseg.com/content/dam/data-analytics/en\\_us/documents/brochures/lseg-comparing-large-language-models.pdf](https://www.lseg.com/content/dam/data-analytics/en_us/documents/brochures/lseg-comparing-large-language-models.pdf).
- [4] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Z. Lin, Y. Wang, L. Ni, W. Gao and J. Guo, "A Survey on LLM-as-a-Judge," 9 March 2025. [Online]. Available: <https://arxiv.org/pdf/2411.15594>.
- [5] H. Li, Q. Dong, J. Chen, H. Su, Y. Zhou, Q. Ai, Z. Ye and Y. Liu, "LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods," 10 December 2024. [Online]. Available: <https://arxiv.org/pdf/2412.05579>.
- [6] M. Krumbick, C. Lovering, V. Reddy, S. Ebner and C. Tanner, "No Free Labels: Limitations of LLM-as-a-Judge Without Human Grounding," [Online]. Available: <https://arxiv.org/abs/2503.05061>.





Visit [lseg.com](https://lseg.com) |  @LSEGplc   
LSEG

© 2025 LSEG. Reproduction or redistribution of LSEG content, including by framing or similar means, is prohibited without the prior written consent of LSEG. LSEG is not liable for any errors or delays in LSEG content, or for any actions taken in reliance on such content.

LSEG Data & Analytics logo wordmark is a trademark of LSEG and its affiliated companies.



**LSEG** DATA &  
ANALYTICS