GDAŃSK UNIVERSITY OF TECHNOLOGY

TASK Quarterly 29 (4) 2025

# Game of Questions: An automated method for unconventional evaluation of Large Language Models

**Przemysław Świat** *, **Łukasz Hein, Marcel Cymanowski**

Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gabriela Narutowicza 11/12, 80-233 Gdańsk, Poland

*swiatprzemyslaw@gmail.com

https://doi.org/10.34808/tq2025/29.4/b

## Abstract

The rapid advancement of Large Language Models (LLMs) has created a need for methods to evaluate their performance, particularly in assessing their domain-specific knowledge and the ability to apply such knowledge in reasoning tasks. Current benchmarks often require substantial manual effort for test case construction and answer scoring. We address this limitation by providing a robust, automatic evaluation method that relies only on unstructured domain text. We introduce the **Game of Questions**, a method that allows the model's knowledge to be tested via an interaction with another model, inspired by the popular web-based game Akinator. The approach requires minimal input from the evaluator and no prepared questions, making it convenient to apply.

## Keywords:

Large Language Model, benchmark

# 1. Introduction

Since the emergence of Transformer architecture [1], the natural language processing field has advanced significantly. Generative Pre-Training was introduced to use large amounts of unstructured text for training with successful results [2]. Large Language Models, like ChatGPT released in November 2022, show revolutionary performance in a wide range of text generation tasks [3]. LLMs can be used in various problems: translation, question answering, reasoning, coding, problem solving, and more. They are widely used in many areas such as software engineering, science, education, medicine, and entertainment [4]. Even though widely used, they still face serious limitations [5], hallucinations [6], and security risks [7]. Due to these challenges, a vast number of benchmarks have been introduced to assess models' capabilities and compare them [8].

There exists an inherent risk in adopting new models [9]. With rapid AI releases, end users and developers of LLM-based systems need a practical way to evaluate the models' knowledge and capabilities in their particular domain of applications. If there is no existing test or benchmark for their use case, they are left with the responsibility to design such tests.

Current benchmarking methods, for both knowledge and reasoning abilities, face a wide range of problems including data contamination, static test cases, and substantial manual effort in the construction of tests. Usually, a human is tasked with the construction of questions for the model to answer [8].

The question arises: how can we simplify the process of testing domain-specific knowledge of LLMs without manually constructing questions and answers?

We propose a new method for automatic benchmarking of Large Language Models. For a given domain, it requires a body of unstructured text and a list of entities. Then, a guessing game between two LLM models is conducted. The rules of the game are very similar to *Akinator*, a popular web-based game. The model under evaluation tries to ask the most informative questions and use the answers to infer the entity.

The contributions of this paper are:

- We propose the *Game of Questions*, an automatic framework for LLM domain knowledge evaluation
- We demonstrate that more advanced models show superior performance in guessing games
- We show that frontier LLM models are able to play the guessing game in both roles
- We analyze the strengths, limitations, and possible extensions of such interaction-based evaluations

The rest of the paper is organized as follows: we start with the background on benchmarking methods and LLM performance in guessing games. Next, we discuss our approach for automatic testing. Then, we show the results of experiments: LLM performance in guessing games and performance of frontier models in our benchmark framework.

# 2. Background

## 2.1. Evaluation methods

In the context of LLMs, benchmarks are used to assess many different properties of models. General benchmarks often focus on three domains: linguistic abilities, knowledge, and reasoning. These cover multiple fields and languages as they are designed to assess the overall capabilities of the model. Some examples of popular benchmarks include: MMLU (Massive Multitask Language Understanding), GPQA (Graduate-level Google-Proof Q&A), MCQA (Multiple-Choice Question Answering), and LogicBench [8].

Many benchmarks are designed for one specific domain. Their goal is to evaluate the model's knowledge and reasoning in a particular field. Such fields include natural sciences, humanities, engineering, and medicine. These kinds of benchmarks are particularly useful for domain experts interested in the model's performance in their area. Examples of such benchmarks are FrontierMath, TPBench, and LegalBench [8].

Recently, there has been an emergence of benchmarks that aim to measure other properties than knowledge and reasoning. These properties include risk and reliability or agentic capabilities. [8]

Knowledge benchmarks typically include a set of questions and answers designed manually by domain experts. There are variations of the form these can have: multi-choice questions, open questions, and similar [8].

There is a substantial portion of benchmarks that evaluate more than just a model's knowledge or reasoning. More holistic approaches are suggested. HELM [10] and BIG-Bench [11] integrate the assessment of knowledge into a larger evaluation, including multiple other metrics. On the other hand, KoLA distinguishes different factors that together form the broad concept of knowledge [12]. This shows that it does not only matter what the model *knows*, but also if it can apply this knowledge to various tasks.

Evaluation of a model's answers can generally be conducted in two ways: automatically or manually. In the former option, some algorithm is used to assign a score to the responses. In the latter case, a human expert needs

to verify all the answers. Automatic evaluation is harder to design as appropriate metrics and algorithms need to be constructed. It has multiple advantages over human evaluation: less biased verification, rapid evaluation speed, and repeatability [13].

There exist many metrics used for automatic evaluation. Some simple ones often encountered in the field of NLP are F1 score and exact match accuracy [13]. These aren't necessarily sufficient for evaluation of various text generation tasks, like summaries or open question answering. In such cases, the usage of synonyms or different sentence structure can result in dramatically different scores [14]. For such tasks, other metrics are suggested. ROUGE assigns a score to a summary by measuring the number of overlapping units (such as n-grams, word sequences) between a generated summary and an ideal summary written by humans [15]. BERTScore uses contextual embeddings to measure similarity between two text sequences [16].

Another method used for automatic evaluation of answers is LLM-as-a-Judge – In this method a separate LLM instance is responsible for calculating various metrics. This method is convenient for researchers, as it provides great flexibility and rapid evaluation. It can process a wide range of data types and be fine-tuned with specific instructions and examples. There are also disadvantages to such an approach; it requires careful experimental design, LLMs have biases that are difficult to mitigate, and they also suffer from hallucinations [17].

To remedy some of the above problems, the LLM-as-a-Judge framework was proposed. Multiple models collaborate to determine the final scores. Collaboration might take on various forms, such as competition or voting [18]. In a different study, the multi-model consistency framework was suggested. It is an iterative algorithm that utilizes the concept of consistency. Multiple models are assessed at the same time and their responses are compared with each other. This avoids the preparation of gold labels and mitigates the issues that arise when a single model is used for evaluation [19].

Due to problems with static benchmarks, such as data contamination, obsolescence, slow construction, and designer biases, there is a push towards more dynamic and evolving evaluations. These include multiple classes of benchmarks: knowledge, reasoning, and others [8]. KonTest leverages existing knowledge graphs to automatically generate a set of questions and answers. Each question has semantically equivalent variants. The model's responses are evaluated against golden answers and against each other. This allows testing of both accuracy and consistency [20]. Label-free Knowledge Deficiency Diagnosis suggests using KL divergence to measure the distance between a model's answers before and after injection of knowledge – it omits manual preparation of answers [21]. OKBench proposes an agentic framework that automates the creation of benchmarks [22].

### 2.2. Guessing games

Akinator is a web-based game. The answerer – a human – is responsible for coming up with an entity in one of the categories: people, animals, objects. Then they have to answer yes/no questions about the chosen object. The guesser – a computer algorithm – is tasked with inventing the queries and deducing the answer. There is a similar game available online, Twenty Questions.

The AI Akinator Game introduced in [23] follows the same rules as the online version of Akinator. The LLM model is introduced as the guesser. The human is in the role of the answerer. The game evaluates the model's deductive and multi-hop reasoning.

In [24] the LLM is introduced in the role of the answerer. They evaluate its yes/no responses with and without injections of additional knowledge from different sources. The knowledge injected from Wikipedia seems to significantly increase the accuracy of the answers.

LLMs in both roles – the answerer and the guesser – were introduced in [25]. The models were able to conduct the games successfully with insightful questions and accurate responses. LLMs show that they are able to strategically narrow down potential objects and reduce uncertainty by applying strategic planning.

## 3. Game of Questions

In our approach to automatic testing of LLMs' domain knowledge, we adopt the concepts of the Akinator game. The model under evaluation takes on the role of the guesser. The model equipped with background knowledge is introduced in the role of the answerer. We call such a model the oracle. As the name suggests, we assume that the answers provided by the oracle will be accurate.

The human evaluator is responsible for two tasks:

- Constructing a list of domain entities. These should represent the concepts that are of interest.
- Providing textual knowledge about the entities. It does not need to be structured in any particular way.

The oracle is injected with the knowledge by directly providing it in the initial prompt. The only limitation is the model's input window size.

For each entity, the guesser asks 20 questions. After that, it is prompted for a final guess. In our framework, we adopt exact match accuracy – the guess is a success or a failure. Additionally, if the entity was present in one of

the questions before the final guess, it is also accepted as a success. The complete evaluation procedure is illustrated in Figure 1.

Let there be m entities and n runs for each entity. We can treat each entity's evaluation result as a Bernoulli random variable. Let $X_{ij}$ represent the result of the j-th run for the i-th entity. Denote the average success rate for entity $i$ as in equation 1

$$\hat{p}_i = \frac{1}{n} \sum_{j=1}^{n} X_{ij} \tag{1}$$

The overall score across all entities is shown in equation 2

$$S = \frac{1}{m} \sum_{i=1}^{m} \hat{p}_i \tag{2}$$

and, assuming independence between runs and between entities, the variance of $S$ can be estimated using a plug-in estimator as in equation 3

$$\widehat{\mathrm{Var}}(S) = \frac{1}{m^2 n} \sum_{i=1}^{m} \hat{p}_i \left(1 - \hat{p}_i\right) \tag{3}$$

This formulation provides a concise measure of average performance along with an estimate of its uncertainty.

# 4. Experiments

We conducted a series of experiments to thoroughly investigate LLMs' proficiency in guessing games. In the three following sections, step by step, we show that current models are sufficiently skilled to play in both roles.

We constructed a small testing dataset of entities. They are divided into three categories inspired by the Akinator game: animals, people, and objects. Each category holds ten entities of varying difficulty; some are much less known than the others. It is a purposeful design choice – we aim to test the limits of the models' capabilities. The complete data are presented in Table 1.

**Table 1:** Grouped entities by category

| People | Animals | Objects |
|---|---|---|
| Gary Stevenson | Axolotl | Book |
| Ignacy Lukasiewicz | British Shorthair | Digital Pen |
| Jack Ma | Cat | Durian |
| Janusz Filipiak | Emperor Penguin | Electric Guitar |
| Jeff Bezos | Fire Ant | Excalibur |
| Jerzy Zięba | Grey Parrot | Fridge |
| Neil deGrasse Tyson | Housefly | iPhone |
| Norm Macdonald | Komodo Dragon | Porsche 911 |
| Robert Lewandowski | Red Panda | Rice Cooker |
| Semyon Korsakov | Sea Anemone | Rubik's Cube |

## 4.1. LLM as Akinator

In this step, multiple models are tested on their performance in the role of the guesser. A human, supported by web search, takes on the role of the answerer. The motivation of this section is to answer the question: are more advanced LLMs better at guessing games than simpler ones?

The experiments are conducted for the models gemini-2.5-flash, gpt-5-mini, grok-4, qwen3-14B, internVL3-5-8B, and stablelm-2-12B. The score is the averaged result of one run for each entity.

The results show that larger models tend to outperform smaller ones by a significant margin. Newer models also seem to achieve better scores. The complete results are shown in figures 2, 3, and 4.

## 4.2. LLM vs Akinator

The goal of this step is to check whether the models can reliably play the role of the answerer. It is a crucial task as the Game of Questions relies on an LLM to correctly answer the questions.

The models used in this part are gemini-2.5-flash, gpt-5-mini, and grok-4. The Akinator takes on the role of the guesser. The score is the averaged result of one run for each entity. A run is qualified as a success if the Akinator is able to guess correctly within two tries.

The results show that advanced LLMs can successfully play the role of the answerer, achieving high scores in two categories out of three. The complete results are presented in Figures 5, 6, and 7.

## 4.3. Game of Questions

Finally, we conduct the Game of Questions experiment. The goal is to show that this automated evaluation is possible and meaningful.

Three models are tested: gemini-2.5-flash, gpt-5-mini, and claude-3.5-haiku. We set n = 3 (number of runs per entity). The same categories and entities as previously are used. For each run, two instances of the same model are placed in both roles. Additional knowledge for the answerer is injected as context in the form of fragments of Wikipedia articles.

The total number of tokens used for all runs (total of 30) for one model was around 3 million. It also took between 1 and 2 hours. The results show better performance of the gpt-5-mini model in each category, as expected as it is the newest model of all three. All results are presented in Figures 8, 9, and 10.

It is worth mentioning that sometimes the models got confused when it came to their roles, for example the

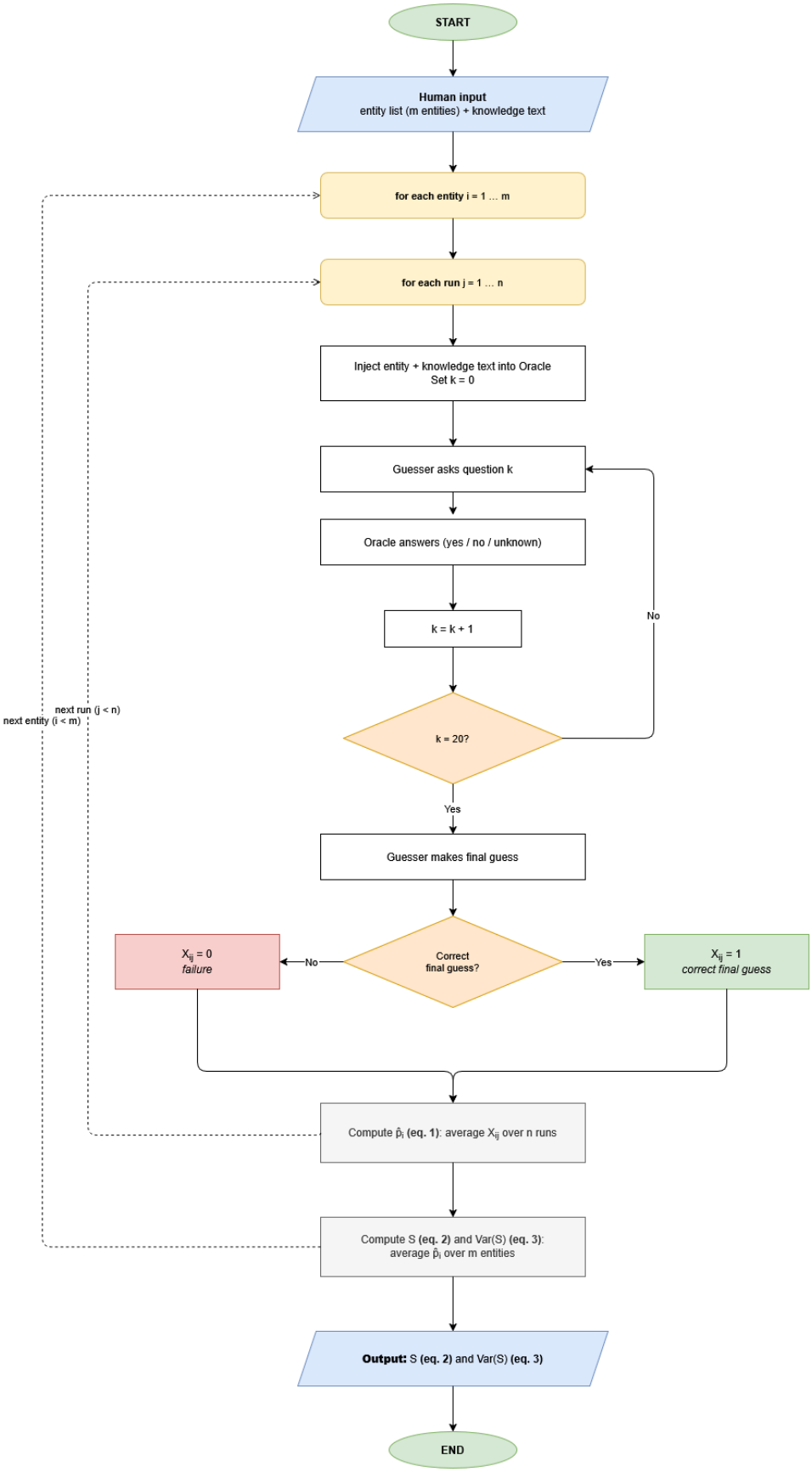


**Figure 1:** Flowchart of the Game of Questions evaluation procedure. The guesser asks up to 20 questions; a run is scored as $X_{ij} = 1$ if the entity is named in any question or the final guess is correct, and $X_{ij} = 0$ otherwise. Per-entity success rates $\hat{p}_i$ (eq. 1) are averaged into the overall score $S$ (eq. 2).

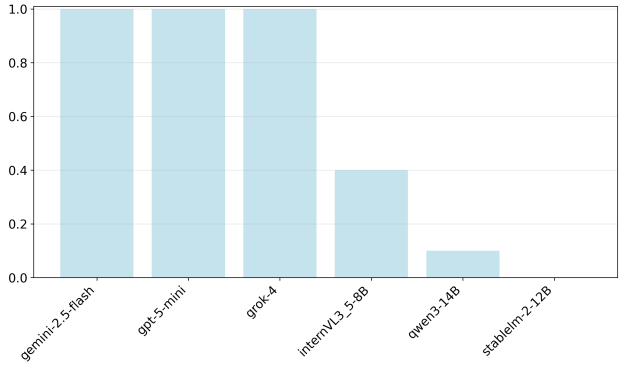


**Figure 2:** Score $S$ (equation 2) of LLM as Akinator for **Animals** category

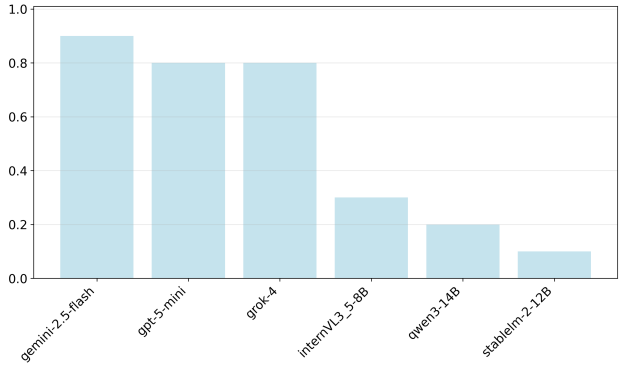


**Figure 3:** Score $S$ (equation 2) of LLM as Akinator for **Objects** category

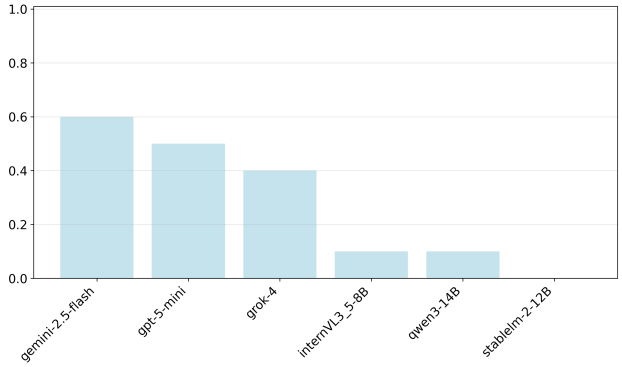


**Figure 4:** Score $S$ (equation 2) of LLM as Akinator for **People** category

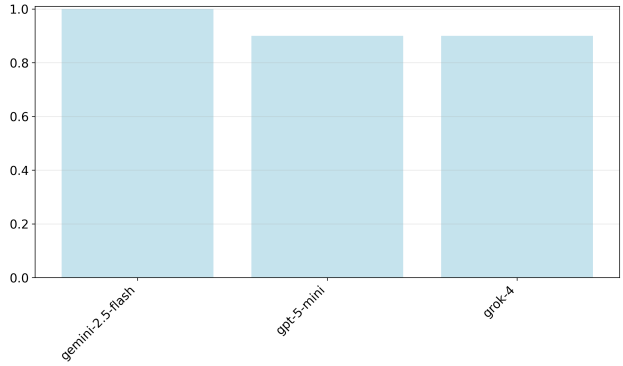


**Figure 5:** Score $S$ (equation 2) of LLM vs Akinator for **Animals** category

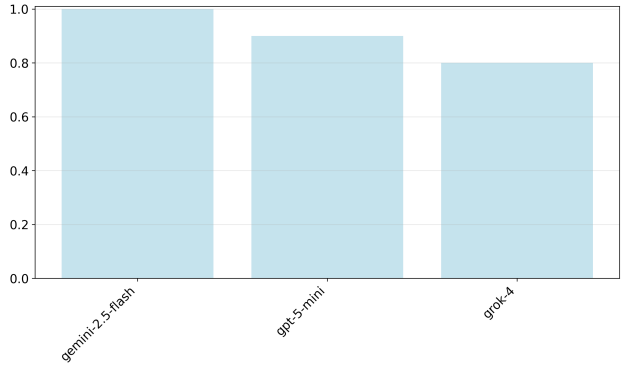


**Figure 6:** Score $S$ (equation 2) of LLM vs Akinator for **Objects** category

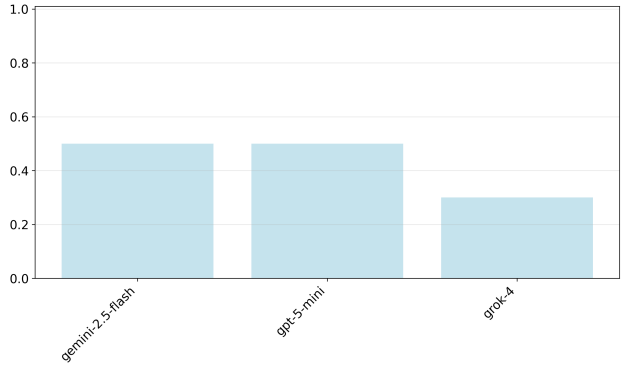


**Figure 7:** Score $S$ (equation 2) of LLM vs Akinator for **People** category

guesser started answering questions. There were some problems with following instructions; the answerer would answer something other than yes/no/unknown in some cases. These were rather singular events.

# 5. Discussion

The results of the Game of Questions experiment show that it is a promising method for almost fully automated LLM benchmarking. Models can successfully play the guessing game in both roles.

There is a clear distinction in the final scores between advanced and simple models. Larger LLMs show better planning and strategic thinking abilities. They also keep track better of previously acquired information. Small models tend to get confused when it comes to the nature of the game – this is a rare occurrence with frontier LLMs.

It should be noted that the current experiments are limited in scale, covering thirty entities across three broad general-purpose categories. The results should therefore be interpreted as a proof of concept rather than a comprehensive evaluation.

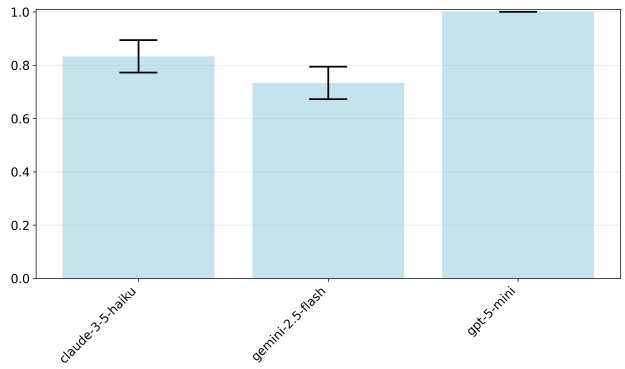


**Figure 8:** Score $S$ (equation 2) of Game of Questions for **Animals** category; error bars show $\pm\sqrt{\widehat{\mathrm{Var}}(S)}$ (equation 3)

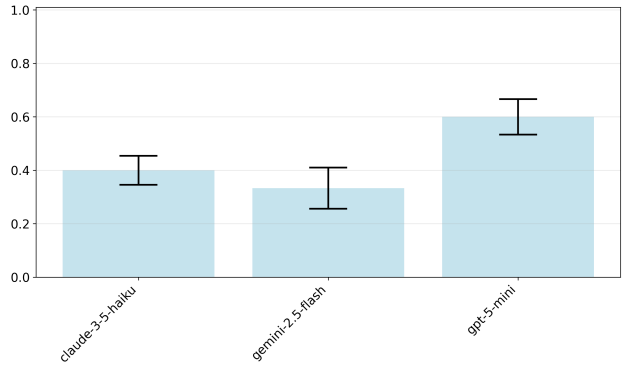


**Figure 9:** Score $S$ (equation 2) of Game of Questions for **Objects** category; error bars show $\pm\sqrt{\widehat{\mathrm{Var}}(S)}$ (equation 3)

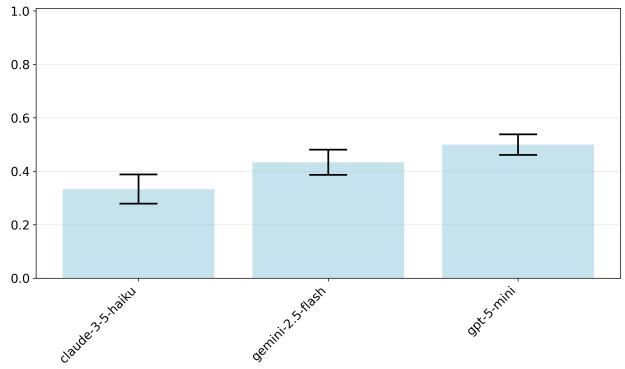


**Figure 10:** Score $S$ (equation 2) of Game of Questions for **People** category; error bars show $\pm\sqrt{\widehat{\mathrm{Var}}(S)}$ (equation 3)

The costs of running a single evaluation scale with n (number of runs per entity) and the number of entities. There is also the factor of injected knowledge – long articles tend to inflate the context size. The total expenses should be significantly smaller than in the case of manually building a full benchmark. Such an action would require considerable human labor.

### 5.1. Future work

Although the current results show that the Game of Questions is a viable tool for LLM benchmarking, we believe that there are some possible enhancements to further limit costs and automate the process.

- **Refined scoring** based on semantic similarity or LLM-as-a-Judge might be more appropriate than exact match. The evaluation could be more resilient to spelling mistakes or the usage of synonyms.
- **Knowledge injection with RAG** instead of copy-pasting Wikipedia articles into context. This would allow for a much greater knowledge base for the oracle model, which could result in better answers and fewer hallucinations. It potentially might also decrease the number of tokens used.
- **Larger and domain-specific datasets**. The current experiments use a small set of thirty entities spread across three broad general-purpose categories. Expanding to a larger entity set within specific domains, such as medicine, computer science, or law, would validate both the method's scalability and its domain applicability.
- **Automated entity discovery**. In the current design, there is still some work expected from the human expert – preparation of the list of entities. We believe that this process could also be automated with the application of LLMs.

## 6. Conclusions

In this paper, we introduced a Game of Questions, a novel approach to automated LLM benchmarking of domain-specific knowledge. Using the interactive structure of a 20-questions game, we have created a framework that is:

- Low-effort. It is easy to set up and relatively inexpensive to run.
- Dynamic. The method tests the model's reasoning, knowledge, and memory in a way that static metrics do not.
- Contamination-resistant. The game relies solely on unstructured text, so it is difficult to train the model specifically for such an evaluation.

Our experiments in multiple models confirm that the framework differentiates between models based on their logical deduction skills, knowledge, and context management. As LLMs continue to evolve, evaluation methods must shift toward the dynamic paradigm demonstrated by the Game of Questions.

# References


[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[2] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.

[3] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2025.

[4] S. M. S. Mohammadabadi, B. C. Kara, C. Eyupoglu, C. Uzay, M. S. Tosun, and O. Karakus, "A survey of large language models: Evolution, architectures, adaptation, benchmarking, applications, challenges, and societal implications," *Preprints*, August 2025.

[5] S. Johnson and D. Hyland-Wood, "A primer on large language models and their limitations," 2024.

[6] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, p. 1–55, Jan. 2025.

[7] M. Q. Li and B. C. M. Fung, "Security concerns for large language models: A survey," 2025.

[8] S. Ni, G. Chen, S. Li, X. Chen, S. Li, B. Wang, Q. Wang, X. Wang, Y. Zhang, L. Fan, C. Li, R. Xu, L. Sun, and M. Yang, "A survey on large language model benchmarks," 2025.

[9] M. N. Sakib, M. A. Islam, R. Pathak, and M. M. Arifin, "Risks, causes, and mitigations of widespread deployments of large language models (llms): A survey," 2024.

[10] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda, "Holistic evaluation of language models," 2023.

[11] A. Srivastava *et al.*, "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," 2023.

[12] J. Yu, X. Wang, S. Tu, S. Cao, D. Zhang-Li, X. Lv, H. Peng, Z. Yao, X. Zhang, H. Li, C. Li, Z. Zhang, Y. Bai, Y. Liu, A. Xin, N. Lin, K. Yun, L. Gong, J. Chen, Z. Wu, Y. Qi, W. Li, Y. Guan, K. Zeng, J. Qi, H. Jin, J. Liu, Y. Gu, Y. Yao, N. Ding, L. Hou, Z. Liu, B. Xu, J. Tang, and J. Li, "Kola: Carefully benchmarking world knowledge of large language models," 2024.

[13] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," 2023.

[14] M. T. R. Laskar, I. Jahan, E. Dolatabadi, C. Peng, E. Hoque, and J. Huang, "Improving automatic evaluation of large language models (LLMs) in biomedical relation extraction via LLMs-as-the-judge," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, eds.), (Vienna, Austria), pp. 25483–25497, Association for Computational Linguistics, July 2025.

[15] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[16] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020.

[17] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, and J. Guo, "A survey on llm-as-a-judge," 2025.

[18] 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," 2024.

[19] Q. Zhu, D. Lyu, X. Fan, X. Wang, Q. Tu, Y. Zhan, and H. Chen, "Multi-model consistency for llms' evaluation," in *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2024.

[20] S. S. Rajan, E. Soremekun, and S. Chattopadhyay, "Knowledge-based consistency testing of large language models," 2025.

[21] K. Xiong, X. Ding, L. Du, J. Ying, T. Liu, B. Qin, and Y. Cao, "Diagnosing and remedying knowledge deficiencies in llms via label-free curricular meaningful learning," 2024.

[22] Y. Li, T. Xu, K. Tang, K. Livescu, D. McAllester, and J. Zhou, "Okbench: Democratizing llm evaluation with fully automated, on-demand, open knowledge benchmarking," 2025.

[23] L. Hu, Q. Li, A. Xie, N. Jiang, I. Stoica, H. Jin, and H. Zhang, "Gamearena: Evaluating llm reasoning through live computer games," 2025.

[24] M. De Bruyn, E. Lotfi, J. Buhmann, and W. Daelemans, "Is it smaller than a tennis ball? language models play the game of twenty questions," in *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP* (J. Bastings, Y. Belinkov, Y. Elazar, D. Hupkes, N. Saphra, and S. Wiegreffe, eds.), (Abu Dhabi, United Arab Emirates (Hybrid)), pp. 80–90, Association for Computational Linguistics, Dec. 2022.

[25] M. Muthu Palaniappan, S. Venkataraman, K. B. Sundharakumar, and S. Natarajan, "An ai-driven approach to the guessing game: Leveraging llama-3.1," in *Evolutionary Artificial Intelligence* (D. Asirvatham, K. Ntalianis, and P. Falkowski-Gilski, eds.), (Singapore), pp. 339–356, Springer Nature Singapore, 2025.