arXiv:2507.08491v2 [cs.CL] 26 Feb 2026

A Third Paradigm for LLM Evaluation: Dialogue Game-Based Evaluation using `clembench`

David Schlangen[1,2] **Sherzod Hakimov**[1] **Chalamalasetti Kranti**[1]
Jonathan Jordan[1] **Philipp Sadler**[1]
[1] Computational Linguistics, University of Potsdam, Germany
[2] DFKI (German Research Center for AI), Berlin

Abstract

There are currently two main paradigms for evaluating large language models (LLMs), *reference-based evaluation* and *preference-based evaluation*. The first, carried over from the evaluation of machine learning models in general, relies on pre-defined task instances, for which reference task executions are available. The second, best exemplified by the LM-arena, relies on (often self-selected) users bringing their own intents to a site that routes these to several models in parallel, among whose responses the user then selects their most preferred one. The former paradigm hence excels at control over what is tested, while the latter comes with higher ecological validity, testing actual use cases interactively. Recently, a third complementary paradigm has emerged that combines some of the strengths of these approaches, offering control over multi-turn, reference-free, repeatable interactions, while stressing goal-directedness: *dialogue game based evaluation*. While the utility of this approach has been shown by several projects, its adoption has been held back by the lack of a mature, easily re-usable implementation. In this paper, we present `clembench`, which has been in continuous development since 2023 and has in its latest release been optimized for ease of general use. We describe how it can be used to benchmark one's own models (using a provided set of benchmark game instances in English), as well as how easily the benchmark itself can be extended with new, tailor-made targeted tests.[1]

1 Introduction

When a new large language model is released, it is customary now to provide two types of evaluation results (see, e.g., (Gemma Team et al., 2025)): First, results are given from user rankings, as collected on the LM Arena (formerly Chatbot Arena, Chiang et al. (2024)). This is meant to represent general user perception of model quality. Second, results on "standard benchmarks" are provided, for example such as are collected by the LLM-harness (Gao et al., 2023). This, in turn, is meant to measure specific capabilities in more depth and with more control than is brought out by the user preferences. While these two types of evaluation complement each other (controlled and static, less controlled and interactive), it has recently been argued that a third combination (controlled and interactive) is needed to acquire a well-rounded impression of the capabilities of a model (Momentè et al., 2025).

feature / *paradigm*	*rb*	*pb*	*dgb*
control over task	✓	✗	✓
replicability	✓	✗	✓
turns	1	1+	1+
parity text / image	✗	✗	✓
cost of benchmarking	fixed	open	fixed
leakage danger	↑	↓↓	↓
extension cost (instances)	↑	open	↓↓
extension cost (tasks)	↑	no control	↓
saturation danger	↑	↓	↓

Table 1: A feature matrix comparing *reference-based (rb)*, *preference-based (pb)* and *dialogue game-based (dgb)* evaluation paradigms across dimensions

As discussed below (Section 2.3), there have been several attempts at realising controlled, interactive evaluation through making LLMs self-play conversational games. Arguably, widespread adoption of this paradigm has been held back by the fact that many of these attempts appear to have been built for a single evaluation campaign only, with less attention being paid to re-usability of code. The package highlighted here, `clembench` (Chalamalasetti et al., 2023), was one of the first instantiations of this idea, and has driven a leader-

---

[1] All code required to run the benchmark, as well as extensive documentation, is available at `https://github.com/clembench/clembench`.

A (quietly reads game card): `Target word: ugly. Taboo words: displeasing, despicable, unattractive.`
(1a) A (to B): Something that is not pleasing to the eye, often making a bad impression.
(1b) B: Eyesore
(2a) A: More about the quality of a person or thing, rather than a nuisance or distraction.
(2b) B: Ugly.
(3a) A: Yay! Well done us.

Figure 1: A constructed example of a successful episode of the two-player Dialogue Game `Taboo`, as played by humans

board that has been continuously updated since early 2023. Thanks to a grant by the University of Potsdam, we can now offer `clembench`$_2$, which has been optimized for ease of use in various use cases, including local evaluation of LLMs or LLM fine-tunes. In this paper, we first briefly describe the “dialogue game-based evaluation” paradigm in general, explaining how it differs from the other paradigms (as illustrated in the feature matrix in Table 1). We then focus specifically on `clembench`$_2$, with the use case of evaluating new model in mind. For this, we show how the existing benchmark can be run, to yield results that can directly be compared to the long history of measurements recorded on the official leaderboard.[2] We also show how easily the benchmark can be extended with games highlighting special abilities.[3]

2 Evaluation Through Self-Play of Dialogue Games

2.1 The General Idea

Figure 1 shows an example of the kind of game that has been used in the "dialogue game-based evaluation paradigm". In this (constructed) example, two humans are playing the game "taboo", in which a *describer* has to provide clues about a concept to a *guesser*, while following the constraint of not using the concept name and some related terms.

Game Master (GM) to A: We are playing a collaborative word guessing game. Your task is to describe a concept, without using its name, and without using some other related terms. The target concept is: ugly; the related words are: displeasing, despicable, unattractive. Start with "CLUE: ", and be brief.
(1a) A to GM: CLUE: Something that is not pleasing to the eye, often making a bad impression.
(1a') GM to B: We are playing a word guessing game. You need to guess a target word that another player is describing to you. You can make one guess at each trial. After each trial you will get another hint. Start with "GUESS: ", and only give a single word. The other player gave the following clue: Something that is not pleasing to the eye, often making a bad impression.
(1b) B to GM: GUESS: Eyesore
(1b') GM to A: Your partner guessed: “eyesore”. Please provide another clue, starting with "CLUE:".

Figure 2: Llama-3.1-405B in prompted self-play of the same instance of `Taboo` as in Figure1 (excerpt).

What makes games like these useful evaluation instruments? To address this question, we can see that even this simple interaction challenges certain capabilities: understanding of game rules and how they constrain the available actions; understanding of a target concept sufficiently to be able to describe it; ability to construct utterances under negative constraints (avoiding certain lexical items); decoding a clue; ability to integrate repeated clues. As has been argued in the literature discussed below, these are capabilities that static datasets such as MMLU-Pro (Wang et al., 2024), Big-bench Hard (Suzgun et al., 2023), and IFEval (Zhou et al., 2023) do not, or at least not as systematically and holistically, test.[4]

How can such games be played with LLMs? While for human player, the acquisition of the rules of a game, making the decision to play it, and the actual game play may constitute separate activities, for LLMs, these need to be folded into one. Figure 2, which is an actual instance of self-play of an LLM,[5] shows how *prompted self-play* can be set up with such models. The crucial element here is the introduction of a *Game Master* which mediates the interaction by prompting each player role into existence. In the figure, one can see how the ‘scaffolding’ shown in a lighter font weight supports the game play (darker weight), which taken on its own is what looks more like the human/human play from Figure 1.

Below, we follow Chalamalasetti et al. (2023) in making a distinction between the *game* itself (e.g.,

---

[2] `https://clembench.github.io/leaderboard.html`

[3] The scientific basis behind the approach is discussed, evaluated, and validated at length in (Chalamalasetti et al., 2023; Hakimov et al., 2025a). The use of the framework as an environment for reinforcement learning of multi-turn problems has been explored by Horst et al. (2025). The present paper aims at making the community aware of it as a resource for easy evaluation of new models or model checkpoints during development.

[4] See also Schlangen (2023, 2019) for an extensive discussion of this question.

[5] Specifically, it is LLama-3.1-405B playing; `https://github.com/clembench/clembench-runs/blob/main/v2.0/Meta-Llama-3.1-405B-Instruct-Turbo-t0.0--Meta-Llama-3.1-405B-Instruct-Turbo-t0.0/taboo/2_low_en/episode_2/transcript.html`

`Taboo`), the way it is explained to a model through prompts (which we call the *dialogue game instantiation*), the specific *game instance* that is to be played (e.g., `Taboo` with a specific target word and set of taboo words), and the *experiment* which collects a set of instances.

2.2 Some Unique Properties

What makes *dialogue game-based evaluation* (DGBe) special? For Table 1, we have compiled into a feature matrix some of the properties as claimed in the work cited below. As can be seen, it is the combination of properties that makes DGBe special. Through the selection of games, instantiations, and game instances as non-changing, pre-constructed elements, a high degree of control over the task is conferred, while the fact that by self-play, only models and no human players are involved, provides a certain degree of replicability. Unlike in static datasets, many of the games allow for or even require multi-turn interactions. The separation of *game* and *game instance* makes it possible to explore different ways of presenting game material; the `clembench`, for example, provides, for some games, coordinated text-only and text/image variants (Hakimov et al., 2025a). The fact that pre-defined experiments are being evaluated also brings DGBe closer to *reference-based evaluation* (RBe), and makes the cost of running the benchmark on a model more predictable and fixed; whereas in *preference-based evaluation* (PBe), this in principle is unlimited and the results get better / more stable, the longer the model is provided and compared. The last three features mentioned in Table 1 all relate to the relative ease with which new game instances and even new games can be created (more on that below), compared to static datasets, which need to be authored or collected and curated.

2.3 Existing Frameworks

The idea of using game self-play for evaluation has been implemented by various frameworks in recent years. An early precursor to this was TextWorld (Côté et al., 2019), which however operated with a single genre of game (text adventures / interactive fiction), and with the expectation of training specialist models. Only with the advent of generalist models that can be *prompted* into being specialists (Brown et al., 2020; Wei et al., 2021) did it become possible to implement this idea at a larger scale, for single games (Bertolazzi et al., 2023) and more generally in frameworks in which various games can be implemented (Chalamalasetti et al., 2023; Qiao et al., 2023a; Gong et al., 2023a; Wu et al., 2024; Zhou et al., 2024; Duan et al., 2024a; Guertler et al., 2025; Cui et al., 2025).

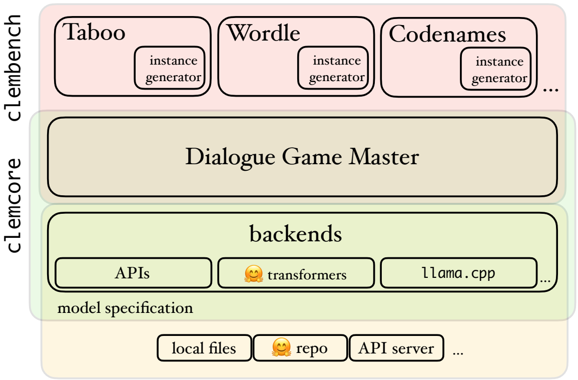

Figure 3: Schematic view of the abstraction layers provided by `clemcore` / `clembench`

2.4 `clemcore` / `clembench`

The `clem` framework demonstrated in this paper has two components: A) `clemcore`, which is the pip-installable backbone abstracting access to LLM into a common interface and providing the infrastructure for setting up games and running experiments on them (e.g., logging, scoring, aggregating results). B) `clembench`, which is a collection of implemented games, and, for each version number of the benchmark, a different pre-compiled set of instances to run.[6] This division of labour is illustrated in Figure 3.

The framework has been used to power the public '`clembench` leader board' since 2023.[7] The long period for which this has already been run has made it possible to track interesting trends; for example, the plot in Figure 5 (Appendix) clearly shows the moment at which open-weight models started to catch up, as well as the generally fast improvements over time. (See (Beyer et al., 2024) for a more detailed discussion.) We also host a *transcript browser* which makes it particularly easy to browse the by now quite large database of self-play transcripts of various types of proprietary and open-weight models.[8]

---

[6] `clemcore`: `https://github.com/clembench/clemcore`; `clembench`: `https://github.com/clembench/clembench`. Both are licensed under the permissive MIT license.

[7] `https://clembench.github.io/leaderboard.html`

[8] Transcript browser: `https://clembench.github.io/transcript-browser.html`. The transcripts are

Currently, 15 games with text-only input and 6 multimodal games (image and text inputs) are made publicly available with more being developed in the pipeline.

3 Running `clembench`

Let us assume that you have trained a new LLM, or produced a new fine-tune of an existing one, and now want to evaluate it for its interactive, dialogic reasoning capabilities. How can you use `clembench` to do that? Detailed and up-to-date instructions can be found on the repository site,[9] and we only highlight some key points here.

The first decision you have to make is how you want to serve your model. Thanks to the abstraction layers described above, `clembench`$_2$ offers a large degree of flexibility here. Besides integrating a number of proprietary APIs (openAI, anthropic, mistral), and emerging API standards (in the backend `openAI-compatible`), the package also allows for inference via the huggingface transformers library (Wolf et al., 2020), via vLLM (Kwon et al., 2023), and via `llama.cpp`.[10] Other ways of interfacing with model inference can easily be implemented as a new backend.

In either case, the model itself is described via a structured entry in the `clembench` *model registry* which specifies all details required for inference, from the backend to use up to additional information like chat-templates and special tokens like EOS.[11] This makes it possible to call the model by a provided name in the command line call to run the evaluation pipeline, with the framework automatically loading the appropriate backend and potentially also any additional files. (Figure 4 illustrates this for a model hosted on a huggingface repository and run locally via the transformers library.)

The games making up the benchmark are similarly described by structured specification files (`clemgame.json`). This makes it possible to conveniently denote specific games or subsets of the whole benchmark when starting the evaluation pipeline. Completing the pipeline (inferencing, transcribing, scoring) results in detailed records of the self-play runs, consisting of transcripts (in machine-readable formats, but for convenience also as html and tex files; for the latter, see Figure 6 in the Appendix for an example), detailed statistics for all metrics specified by the individual games – and, as the condensed form of all metrics, a single `clemscore` (between 0 and 100), which can conveniently be reported.[12]

```
{
  "model_name": "my-model-8b",
  "huggingface_id": "my-org/My-Model-8B",
  "backend": "huggingface_local",
}
```

Figure 4: A minimal *model registry* entry for a model hosted on a public repository and inferenced with the `transformers` library

In the current respective latest versions (v2.0 for text-only, v1.6 for multimodal), running the whole benchmark (14 text-only games with in total 817 instances; 5 multimodal games with 560 instances), takes about 360 minutes on two NVIDIA A100 80GB GPUs for a 70B model run via the huggingface backend (text only) / 350 minutes on one A100 for a 40B model (multimodal).

4 Extending the Benchmark

As mentioned above, a particularly interesting feature of game-based evaluation is the ease with which the tests can be extended or tailored to particular questions. (Of course, for *comparability*, a fixed state of the benchmark must be chosen, such as that defined by the `clembench` repository.) We briefly describe the process here, again referring the reader to the extensive documentation of the package for details.

4.1 Creating New Instances

For certain use cases (such as using the framework for *learning*, as described by Horst et al. (2025)), it can be necessary to create new instances of problems for the existing games. All games in `clembench` come with an *instance generator* step, which automatically creates new stimuli. How this is done differs between games, from sampling from a larger pre-compiled set (e.g., in wordle) to programmatically generating stimuli (as in imagegame). If the larger set is exhausted, creating that resource may require some manual effort (e.g., in taboo, words and related words must be

---

also all stored at `https://github.com/clembench/clembench-runs/` (more than 3GB by now).

[9] `https://github.com/clp-research/clembench`

[10] `https://github.com/ggml-org/llama.cpp`

[11] Similarly to the newly emerging `model.yaml` approach (`https://modelyaml.org`), but predating it for several years. We are actively exploring possibilities for adoption.

[12] See `https://clembench.github.io/leaderboard.html` to explore the various evaluation depths.

derived in a semi-automatic process making use of resources such as wordnet (Fellbaum, 1998)). In any case, what is never needed is the collection of reference answers – an important point about game-based evaluation is that it is *reference-free*!

4.2 Creating New Games

Even though the standard `clembench` benchmark comes with a large number of games, covering a wide spectrum of capabilities (Beyer et al., 2024), for certain special interests, it may be advantageous to define new games. Again, we provide extensive documentation to guide through this process,[13] of which we only cover a few highlights here. To implement the game logic, classes from `clemcore` that implement the concept of the *Game Master* (as described in Section 2.1 above) can be instantiated. For simple round-robin games, only scoring rules and success conditions need to be defined; more flexibility can be achieved by overwriting defaults. `clemcore` also abstracts away all details of handling experiments and individual runs. The instance generator takes the prompt templates that describe the game to the player, together with an instance set, and creates the specifications of individual instances that are collected in an experiment. All in all, depending on the complexity, realising a game can take as little as 2-3 hours (for simpler abstract games); importantly, however, the framework also allows for the implementation of ambitous game using complex environments such as AI2-THOR (Kolve et al., 2022).

4.3 Multilingual Evaluation with `clembench`: Adding New Languages

As explained, the way a game is *instantiated* so that LLMs can be made to play it is via textual descriptions on the one hand, and the game logic implemented in code on the other. The latter part is language agnostic. The `clembench` as referred to above comes with prompts written in English, and, for games where this is relevant (e.g., wordle, taboo), with instances in English. While this is something we have only begun to explore ourselves (see initial results reported by (Beyer et al., 2024)), the framework is ready for defining experiments with stimuli in other languages. Hakimov et al. (2025b) evaluated the negotiation abilities and multilingual aspect in LLMs by developing dialogue games in three languages: English, German, Italian. The flexibility of the machinery for running experiments (see Section 3) makes it possible to easily define language-specific as well as cross-lingual benchmark runs. We are currently setting up a mechanism for leveraging the community to collect game instantiations in a large set of languages.

5 "Vibe-checking" the Agentic Abilities of a Model

A new addition in `clembench`$_2$ is the integration of the dialogue experiment framework `slurk` (Götze et al., 2022). This integration makes it possible to interact locally (on one's own machine) with a model as a partner in the implemented dialogue games and thus to get a feel for how it behaves in constrained agentic conditions.[14] The `slurk` framework has also proven to be able to handle larger crowdsourcing experiments (e.g., Ilinykh et al. (2019)), so that more thorough quantitative studies are also possible.[15]

6 Evaluating the Evaluator

6.1 Human Game-play Performance

To assess the validity of games and to compare with LLM performance, in (Beyer et al., 2024) we performed a user study on the initial set of games. We asked the participants to play 10-15 episodes per game (leaving out wordle-clue and wordle-critic, as these are only variants of the main wordle game). All episodes were played to the end where the human participants obliged the formatting-related game rules (reaching 100% in *Played* score). The resulting *Quality* scores were as follows: wordle: *72*, taboo: *80.5*; drawing: *95.2*; reference: *100*; leading to an average of *86.93*. The best performing model *o3-mini* on the *clembench leaderboard (v2.0)* still lags behind the average human performance, it has an average Quality score of *82.23* on the same games.

---

[13] `https://github.com/clembench/clemgame-template`

[14] A common practice that has been systematically argued for e.g. by (Dunlap et al., 2025).

[15] A preview of the public interface is hosted at `https://clemp.ling.uni-potsdam.de`; note however that we cannot guarantee that backend models as game play partners are available at any time, so this should be seen only as a demo of the interface. The game play itself will be demonstrated at the conference.

6.2 Reproducibility of Benchmark Results

All games used for benchmarking are publicly available as well as the their specific instance versions. We run the vanilla (without any agentic or other tools) version of benchmarked LLMs by keeping the same parameters (temperature, library/API version, etc.) across them. Thus, as long as the model providers do not make changes to their models (applies to commercial models), the benchmark results are reproducible. (To the extent that LLM calls are deterministic. However, slight differences in responses are unlikely to affect final scores, which are an average over the perfomance on a larger set of instances.) All existing benchmark runs are made publicly available.[16]

6.2.1 The Value of Game-Based Evaluation

Momentè et al. (2025) have compared the use of games for interactive evaluation (as implemented in `clembench`), the use of targetted cognitive test sets, and the use of more generalist benchmarking datasets. A central finding was that games provide a more fine-grained means to discriminate between models; the authors recommend an approach that "triangulates" performance through a combination of interactive and static benchmarking.

7 Conclusions

This paper accompanies a demonstration of the $\texttt{clembench}_2$ evaluation package. We hope this paper has illustrated that running game-based evaluations does not need to be complicated, and in fact, using `clembench`, in most cases is a matter of running a single command (and providing a single file, describing the model that is to be evaluated). The reasons *why* models should be evaluated in this way and what exactly is being measured could only be sketched here, and for a full argumentation, we refer the reader to recent works such as (Momentè et al., 2025) as well as the papers mentioned in Section 2.3 above. We hope that the work reported here can contribute to further adoption of this complementary evaluation paradigm, in particular in light of the increased use of LLMs as backbones for interactive and agentic tasks.

References

Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. 2023. LMRL Gym: Benchmarks for Multi-Turn Reinforcement Learning with Language Models. *arXiv preprint*. ArXiv:2311.18232 [cs].

Leonardo Bertolazzi, Davide Mazzaccara, Filippo Merlo, and Raffaella Bernardi. 2023. ChatGPT's information seeking strategy: Insights from the 20-questions game. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 153–162, Prague, Czechia. Association for Computational Linguistics.

Anne Beyer, Kranti Chalamalasetti, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2024. clembench-2024: A Challenging, Dynamic, Complementary, Multilingual Benchmark and Underlying Flexible Framework for LLMs as Multi-Action Agents. *arXiv preprint*. ArXiv:2405.20859 [cs].

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. 2024. How Well Can LLMs Negotiate? NegotiationArena Platform and Analysis. *arXiv preprint*. ArXiv:2402.05863 [cs].

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. *arXiv preprint*. ArXiv:2308.07201 [cs].

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. ArXiv: 2403.04132.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng,

---

[16] `https://github.com/clembench/clembench-runs`

Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. GameBench: Evaluating Strategic Reasoning Abilities of LLM Agents. *arXiv preprint*. ArXiv:2406.06613 [cs].

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2019. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pages 41–75. Springer.

Christopher Zhang Cui, Xingdi Yuan, Zhang Xiao, Prithviraj Ammanabrolu, and Marc-Alexandre Côté. 2025. Tales: Text adventure learning environment suite. *arXiv preprint arXiv:2504.14128*.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024a. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations. *Preprint*, arXiv:2402.12348.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024b. GTBench: Uncovering the Strategic Reasoning Limitations of LLMs via Game-Theoretic Evaluations. *arXiv preprint*. ArXiv:2402.12348 [cs].

Lisa Dunlap, Krishna Mandal, Trevor Darrell, Jacob Steinhardt, and Joseph E Gonzalez. 2025. Vibecheck: Discover and quantify qualitative differences in large language models. *Preprint*, arXiv:2410.12851.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and et al. 2025. Gemma 3 Technical Report. *arXiv preprint*. ArXiv:2503.19786 [cs].

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. 2023a. Mindagent: Emergent gaming interaction. *Preprint*, arXiv:2309.09971.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. 2023b. MindAgent: Emergent Gaming Interaction. *arXiv preprint*. ArXiv:2309.09971 [cs].

Jana Götze, Maike Paetzel-Prüsmann, Wencke Liermann, Tim Diekmann, and David Schlangen. 2022. The slurk interaction server framework: Better data for better dialog models. In *Proceedings of the Language Resources and Evaluation Conference*, pages 4069–4078, Marseille, France. European Language Resources Association.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. TextArena. *arXiv preprint*. ArXiv:2504.11442 [cs].

Sherzod Hakimov, Yerkezhan Abdullayeva, Kushal Koshti, Antonia Schmidt, Yan Weiser, Anne Beyer, and David Schlangen. 2025a. Using game play to investigate multimodal and conversational grounding in large multimodal models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5686–5718, Abu Dhabi, UAE. Association for Computational Linguistics.

Sherzod Hakimov, Roland Bernard, Tim Leiber, Karl Osswald, Kristina Richert, Ruilin Yang, Raffaella Bernardi, and David Schlangen. 2025b. The price of thought: A multilingual analysis of reasoning, performance, and cost of negotiation in large language models. *Preprint*, arXiv:2510.08098.

Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. Playpen: An environment for exploring learning from dialogue game feedback. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29842–29879, Suzhou, China. Association for Computational Linguistics.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Meetup! a corpus of joint activity dialogues in a visual environment. In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2019 / LondonLogue)*, London, UK.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. 2022. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint*. ArXiv:1712.05474 [cs].

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents. *arXiv preprint*. ArXiv:2308.03688 [cs].

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. *arXiv preprint*. ArXiv:2401.13178 [cs].

Filippo Momentè, Alessandro Suglia, Mario Giulianelli, Ambra Ferrari, Alexander Koller, Oliver Lemon, David Schlangen, Raquel Fernández, and Raffaella Bernardi. 2025. Triangulating LLM progress through benchmarks, games, and cognitive tests. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20051–20072, Suzhou, China. Association for Computational Linguistics.

Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. 2023a. Gameeval: Evaluating llms on conversational games. *Preprint*, arXiv:2308.10032.

Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. 2023b. GameEval: Evaluating LLMs on Conversational Games. *arXiv preprint*. ArXiv:2308.10032 [cs].

David Schlangen. 2019. Language tasks and language games: On methodology in current natural language processing research. *CoRR*, abs/1908.10747.

David Schlangen. 2023. Dialogue games for benchmarking language understanding: Motivation, taxonomy, strategy. *CoRR*, abs/2304.07007.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Ruoyao Wang, Graham Todd, Eric Yuan, Ziang Xiao, Marc-Alexandre Côté, and Peter Jansen. 2023. ByteSized32: A Corpus and Challenge Task for Generating Task-Specific World Models Expressed as Text Games. *arXiv preprint*. ArXiv:2305.14879 [cs].

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned Language Models Are Zero-Shot Learners. pages 1–46. ArXiv: 2109.01652.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP 2020, Demo Track*, pages 38–45.

Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. 2023. SmartPlay : A Benchmark for LLMs as Intelligent Agents. *arXiv preprint*. ArXiv:2310.01557 [cs].

Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. 2024. Smartplay: A benchmark for llms as intelligent agents. *Preprint*, arXiv:2310.01557.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint*. ArXiv:2404.07972 [cs].

Qingchen Yu, Zifan Zheng, Ding Chen, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2025. GuessArena: Guess Who I Am? A Self-Adaptive Framework for Evaluating LLMs in Domain-Specific Knowledge and Reasoning. *arXiv preprint*. ArXiv:2505.22661 [cs].

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents. In *Proceedings of ICLR 2024*, pages 1–45.

A Appendix

Table 2 lists all related frameworks. The column “Maintained” is determined whether there have been changes on the respective Git repository in the last three months. Figure 5 shows trends observed on the `clembench` leader board. Figure 6 shows an example transcript.

Framework	**Git repo**	**First commit**	**Latest commit**	**Maintained**
Tales (Cui et al., 2025)	`https://github.com/microsoft/tale-suite`	2018-06-06	2025-04-23	✓
Sotopia (Zhou et al., 2024)	`https://github.com/sotopia-lab/sotopia`	2023-03-31	2025-05-05	✓
Bytesized32 (Wang et al., 2023)	`https://github.com/cognitiveailab/BYTESIZED32`	2023-05-12	2024-07-08	✗
Clembench (Chalamalasetti et al., 2023)	`https://github.com/clembench/clembench`	2023-06-07	2025-06-30	✓
GameEval (Qiao et al., 2023b)	`https://github.com/jordddan/GameEval`	2023-08-13	2023-09-03	✗
MindAgent (Gong et al., 2023b)	`https://github.com/mindagent/mindagent`	2023-09-19	2024-06-12	✗
Negotiation Arena (Bianchi et al., 2024)	`https://github.com/vinid/NegotiationArena`	2023-09-30	2024-02-15	✗
SmartPlay (Wu et al., 2023)	`https://github.com/microsoft/SmartPlay`	2023-10-02	2024-04-11	✗
OSWorld (Xie et al., 2024)	`https://github.com/xlang-ai/OSWorld`	2023-10-16	2025-07-02	✓
AgentBench (Liu et al., 2023)	`https://github.com/THUDM/AgentBench`	2023-10-18	2025-01-30	✗
LMRL Gym (Abdulhai et al., 2023)	`https://github.com/abdulhaim/LMRL-Gym`	2023-11-25	2024-07-02	✗
GameBench (Costarelli et al., 2024)	`https://github.com/Joshuaclymer/GameBench`	2023-12-14	2024-06-27	✗
AgentBoard (Ma et al., 2024)	`https://github.com/hkust-nlp/AgentBoard`	2024-01-17	2024-04-23	✗
GTBench (Duan et al., 2024b)	`https://github.com/jinhaoduan/GTBench`	2024-02-07	2024-09-07	✗
ChatEval (Chan et al., 2023)	`https://github.com/thunlp/ChatEval`	2024-04-15	2024-10-19	✗
TextArena (Guertler et al., 2025)	`https://github.com/LeonGuertler/TextArena`	2024-09-23	2025-06-25	✓
GuessArena (Yu et al., 2025)	`https://github.com/IAAR-Shanghai/GuessArena`	2025-02-16	2025-05-29	✓

Table 2: Game-based LLM Benchmarking Frameworks

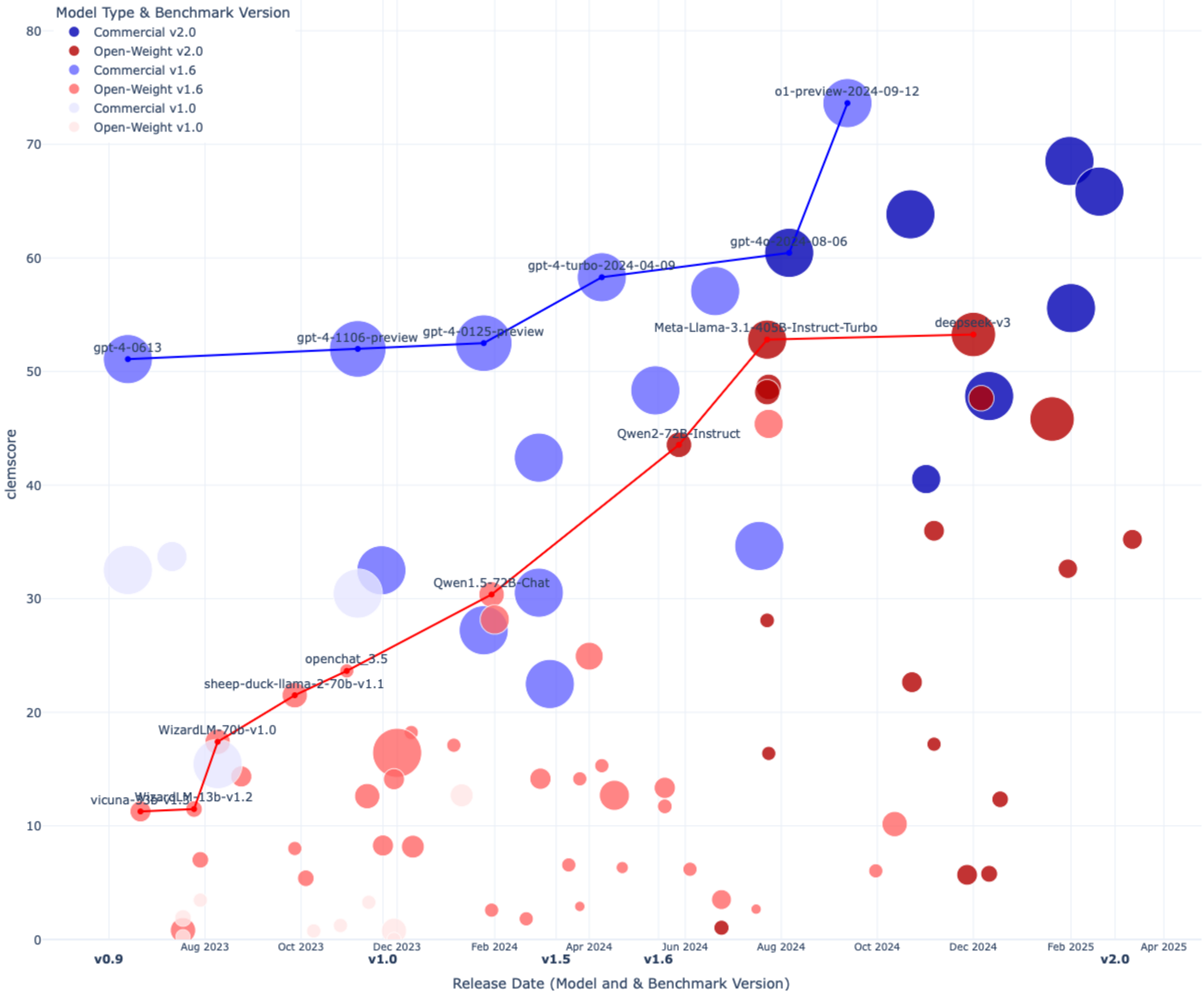

Figure 5: Development of achieved scores on clembench since 2023. Interactive version available at `https://clembench.github.io/leaderboard.html` ("trends").

#	A	Game Master	B
1	[A⟨GM] You are playing a collaborative word guessing game in which you have to describe a target word for another player to guess. Rules: (a) You have to reply in the form: CLUE: <some text>. Guesses from the other player will start with GUESS. (b) You cannot use the target word itself, parts or morphological variants of it in your description. (c) In addition, the same rules apply for related words which are provided below. End conditions: (i) If you use the target word or a related word in your description, then you lose. (ii) If the other player can guess the target word in 3 tries, you both win. Let us start. This is the target word that you need to describe and that the other player needs to guess: winter Related words are: - cold - snow - season Important: You are under time pressure, give short descriptions that are to the point!		
2	[A⟩GM] CLUE: Time of year with shortest days and lowest temperatures.		
3		[GM|GM] continue	
4		[GM|GM] time of year with shortest days and lowest temperatures	
5			[GM⟩B] You are playing a collaborative word guessing game in which you have to guess a target word that another player describes to you. You can make one guess at each trial. You win when you guess the target word. You lose when you cannot guess it in 3 tries. After each trial you will get a new hint from the other player which starts with CLUE. Make your guesses by just saying the word using the following form: GUESS: <a word> Let us start. CLUE: Time of year with shortest days and lowest temperatures.
6			[GM⟨B] GUESS: Winter
7		[GM|GM] continue	
8		[GM|GM] winter	
9		[GM|GM] end game	

Figure 6: Example of the automatically generated transcript format. (Here: Llama-3.3-70B-instr playing Taboo, epsiode "0-high-en", 0.)