# SarcBench: A Pilot Benchmark for Sarcasm Understanding in Language Models

Navadeep Budda

**Abstract**

Sarcasm is easy to recognize in some cases, but hard to evaluate cleanly. A sarcastic sentence often means the opposite of what it says, but that is not always enough. The listener also has to know who is being criticized, whether the speaker is joking or serious, and how the surrounding context changes the meaning.

This report introduces SarcBench v0.1, a pilot multiple-choice benchmark for evaluating sarcasm understanding in language models. Instead of asking models to only label a sentence as sarcastic or not sarcastic, SarcBench asks models to choose the answer that best captures what the speaker most likely means. The benchmark uses six-option multiple-choice questions across five categories: intended meaning, target identification, sentiment reversal, sincere control, and context dependence.

In the v0.1 private evaluation, nine language models were tested on 125 hidden questions. Each model was evaluated over five independent runs with the same standard zero-shot prompt. The main scores are Avg@5, the mean accuracy over five runs, and Maj@5, the accuracy of the model's majority answer across those runs. The clean merged results show that several current models perform extremely well: Claude Opus 4.7, Claude Sonnet 4.6, Gemini 3 Flash Preview, GPT-5.5, and Gemini 3.1 Pro Preview all reached 100.0% Avg@5 and 100.0% Maj@5. DeepSeek V4 Flash reached 99.68% Avg@5, DeepSeek V4 Pro reached 98.56%, Grok 4.1 Fast reached 97.92%, and Kimi K2.6 reached 89.28%.

These results show that the evaluation pipeline works, but they also show that SarcBench v0.1 is not difficult enough to separate the strongest current models. The main contribution of this pilot release is the benchmark format, category design, scoring setup, and public reporting structure. Future versions should include a larger and harder private set, independent human validation, and a formal human baseline.

## 1 Introduction

Sarcasm is a form of meaning mismatch. A person may say something positive while clearly meaning something negative, or say something mild while implying frustration, criticism, or disbelief. This makes sarcasm difficult for language models because the literal words are often misleading. A model has to use context, speaker intent, social expectations, and common sense to decide what is actually being communicated.

Earlier work on sarcasm detection has often treated the problem as binary classification: given a sentence, decide whether it is sarcastic or not sarcastic. This is useful, but it does not fully test whether a model understands the meaning behind the sarcastic statement. A model can sometimes detect that sarcasm is present without correctly explaining what the speaker means, who the sarcasm targets, or why the context makes the statement non-literal.

This was the motivation for SarcBench. Rather than asking only "is this sarcastic?", SarcBench asks a more practical question: can the model infer what the speaker actually means?

This shift matters because real communication is rarely just a yes-or-no sarcasm label. A sarcastic comment usually has a target, a reason, and an intended meaning. Sincere comments can also look sarcastic if a model overreacts to certain words. For example, a phrase like "great timing" can be sarcastic when something goes wrong, but sincere when something works out at the perfect moment. SarcBench is designed to test that distinction.

## 2 Background and Related Work

Sarcasm has been studied in natural language processing for years, often through datasets that ask models to classify text as sarcastic or not sarcastic [2]. Sarcasm Corpus V2 is one example. It is based on the Internet Argument Corpus and includes posts annotated for sarcasm across general sarcasm, hyperbole, and rhetorical questions [4, 8]. Its general sarcasm subset contains 3,260 sarcastic posts and 3,260 non-sarcastic posts [8].

Previous sarcasm datasets are valuable, but they also show why the task is complicated. Sarcasm Corpus V2 contains binary labels and comes from online debate forums, which can make the language domain specific. A 2024 survey-style analysis of sarcasm datasets notes that Sarcasm Corpus V2 is relatively large and context-aware during annotation, but also points out that the released dataset itself does not provide the original contexts and is shaped by its online debate source domain [9].

This project also builds on an earlier study comparing ChatGPT-4, ChatGPT-3.5, and Bard on sarcasm detection. That study tested 100 sentences, split evenly between sarcastic and non-sarcastic examples, using the prompt "Is the following sentence sarcastic or not? Please provide a brief explanation for your answer." The results showed that models differed in how they handled sarcastic and non-sarcastic sentences, with ChatGPT-4 and Bard showing more balanced behavior and ChatGPT-3.5 performing better on non-sarcastic cases than sarcastic ones [7].

SarcBench is meant to move beyond that earlier binary setup. Instead of measuring whether a model can classify sarcasm, it measures whether the model can choose the intended interpretation from plausible alternatives.

SarcBench is also inspired by benchmark design choices from recent language model evaluations. MMLU helped popularize broad multiple-choice testing across many knowledge areas, and its public repository describes it as the ICLR 2021 benchmark "Measuring Massive Multitask Language Understanding" [1, 10]. BIG-bench used a large collection of tasks to probe abilities believed to be difficult for language models, with 204 tasks contributed by hundreds of authors [6]. HELM emphasized transparent and standardized evaluation, comparing models across shared scenarios and reporting more than one metric [3].

The closest design influence for SarcBench is SimpleBench. SimpleBench uses handcrafted six-option multiple-choice questions, keeps part of the benchmark private to reduce contamination, evaluates models over five runs, and reports both average accuracy and majority-vote accuracy [5]. SarcBench uses a similar scoring style, but focuses specifically on sarcasm and non-literal meaning.

## 3 Benchmark Design

SarcBench v0.1 is a six-option multiple-choice benchmark. Each item includes a short context, an utterance, a question, six possible answers labeled A through F, and one correct answer. The goal is not simply to detect sarcasm. The goal is to identify the meaning that best fits the situation.

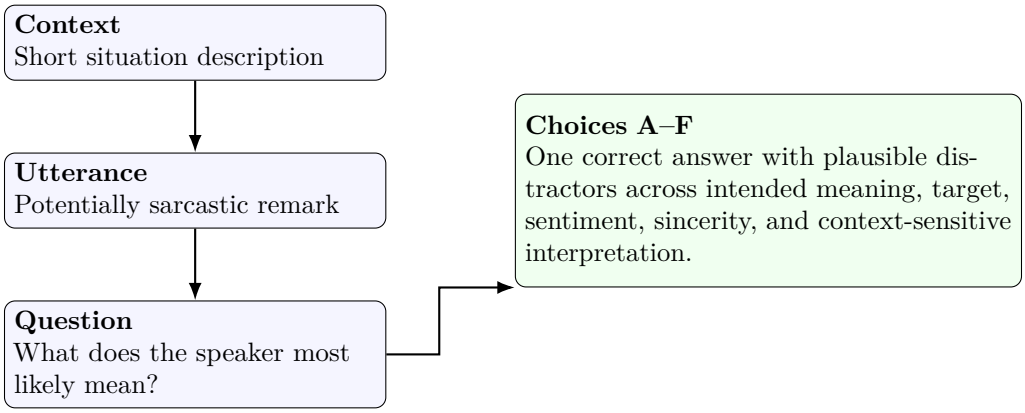


Figure 1: SarcBench item structure. Each question combines context, an utterance, and six candidate interpretations.

The private scored set contains 125 questions. The questions are grouped into five categories:

Table 1: SarcBench v0.1 categories.

| **Category** | **What it tests** |
|---|---|
| Intended meaning | Whether the model can infer what the speaker actually means when the literal wording conflicts with the context. |
| Target identification | Whether the model can identify who or what the sarcastic comment is aimed at. |
| Sentiment reversal | Whether the model can detect when positive wording communicates negative meaning, or the reverse. |
| Sincere control | Whether the model can avoid falsely labeling sincere comments as sarcastic. |
| Context dependence | Whether the model can use surrounding events to decide whether the remark is sarcastic, sincere, playful, or understated. |

The sincere control category is especially important. A benchmark made only of sarcastic examples can train a model, or a human test taker, to assume that every sentence is sarcastic. SarcBench includes sincere and understated cases so that models must pay attention to the context instead of relying on sarcasm-shaped wording.

Each question also includes a quality checklist in the dataset, including whether the item has enough context, whether there is only one best answer, whether the answer is not obvious from keywords alone, whether the distractors are plausible, and whether the item is safe for release.

A simplified example of the kind of reasoning SarcBench tests is shown in Figure 2.

This item is not hard because the vocabulary is complicated. It is hard only if the model follows the literal phrase “very efficient” instead of using the context. That is the core idea behind SarcBench.

**Context:** A cafe promises pickup in ten minutes. After forty minutes, the order is still being prepared.
**Utterance:** "Very efficient."
**Correct interpretation:** The speaker is sarcastically criticizing the slow service.

Figure 2: Simplified example of the kind of reasoning SarcBench tests.

## 4 Evaluation Setup

The v0.1 evaluation used the hidden 125-question private set. Each model was run five times on the same questions using the same standard zero-shot prompt. The prompt instructed the model to answer the multiple-choice question and output only the final answer in the form:

```
Final Answer: X
```

where X is A, B, C, D, E, or F.

The evaluation script then extracted the answer letter and compared it with the correct answer. A response was counted as correct only if the extracted answer matched the ground truth. The raw result logs include the model ID, returned model name when available, question ID, category, correct answer, predicted answer, correctness, token usage, cost, latency, and any error. Raw outputs show the expected answer format, such as "Final Answer: F," with successful parsing and no error.

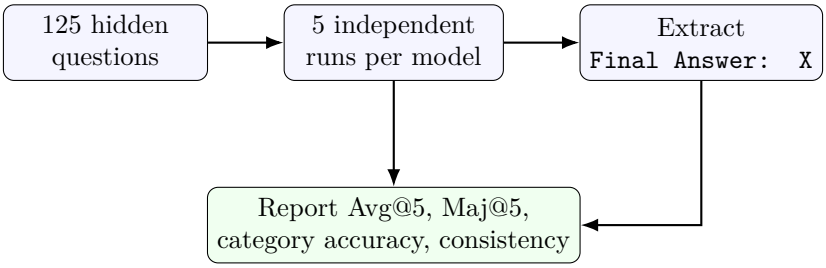


Figure 3: Evaluation pipeline used for the v0.1 private pilot leaderboard.

The main metrics are:

Table 2: Primary reported metrics.

| Metric | Meaning |
|---|---|
| Avg@5 | The average accuracy across five independent runs. |
| Maj@5 | The accuracy after taking the model's majority answer across five runs for each question. |
| Category accuracy | Accuracy within each of the five SarcBench categories. |
| Consistency rate | How often the model produced the same final correctness pattern across repeated runs. |

Because SarcBench uses six answer choices, random guessing gives an expected Avg@5 score of about 16.7%. This is the same chance baseline used in other six-option multiple-choice benchmarks such as SimpleBench.

The reported runs used the benchmark script defaults unless changed in the command line: temperature 0.7, top-p 0.95, and standard zero-shot prompting. For most models, the final official run used a 128-token output limit. Four models were later rerun with a larger output limit because their earlier scores were affected by answer truncation. The final merged leaderboard uses the corrected rerun for Gemini 3.1 Pro Preview, DeepSeek V4 Flash, DeepSeek V4 Pro, and Kimi K2.6.

# 5 Results

The final v0.1 pilot leaderboard combines two clean runs. The first run evaluated nine models and produced valid scores for the top group, including Claude Opus 4.7, Claude Sonnet 4.6, Gemini 3 Flash Preview, GPT-5.5, and Grok 4.1 Fast. The second run corrected the models that had been affected by output-length issues: Gemini 3.1 Pro Preview, DeepSeek V4 Flash, DeepSeek V4 Pro, and Kimi K2.6.

Table 3: Final SarcBench v0.1 pilot leaderboard.

| **Rank** | **Model** | **Provider** | **Avg@5** | **Maj@5** |
|---|---|---|---|---|
| 1 | Claude Opus 4.7 | Anthropic | 100.0% | 100.0% |
| 1 | Claude Sonnet 4.6 | Anthropic | 100.0% | 100.0% |
| 1 | Gemini 3 Flash Preview | Google | 100.0% | 100.0% |
| 1 | GPT-5.5 | OpenAI | 100.0% | 100.0% |
| 1 | Gemini 3.1 Pro Preview | Google | 100.0% | 100.0% |
| 6 | DeepSeek V4 Flash | DeepSeek | 99.68% | 100.0% |
| 7 | DeepSeek V4 Pro | DeepSeek | 98.56% | 98.4% |
| 8 | Grok 4.1 Fast | xAI | 97.92% | 98.4% |
| 9 | Kimi K2.6 | MoonshotAI | 89.28% | 92.0% |
| Baseline | Random chance | None | 16.7% | 16.7% |

The top result is not a single model winning by a small margin. Instead, five models hit the ceiling of the benchmark. Claude Opus 4.7, Claude Sonnet 4.6, Gemini 3 Flash Preview, GPT-5.5, and Gemini 3.1 Pro Preview all scored 100.0% on both Avg@5 and Maj@5. DeepSeek V4 Flash and DeepSeek V4 Pro also scored extremely high after rerunning with a larger output limit. Kimi K2.6 was the only evaluated model with a noticeably lower score, especially on sincere control.

The corrected category scores show the same pattern. Gemini 3.1 Pro Preview scored 100.0% in every category. DeepSeek V4 Flash scored 100.0% on intended meaning, target identification, sentiment reversal, and sincere control, with 98.0% on context dependence. DeepSeek V4 Pro scored 100.0% on intended meaning, target identification, and sentiment reversal, with 96.0% on sincere control and context dependence. Kimi K2.6 scored 98.0% on intended meaning and 100.0% on sentiment reversal, but dropped to 68.0% on sincere control.

# 6 Discussion

The main finding is simple: SarcBench v0.1 is too easy for the strongest current models.

That does not mean the benchmark failed. In a pilot release, this is useful information. The evaluation code worked, the answer extraction worked, the scoring format worked, and the benchmark produced a clean leaderboard. The problem is that the private set did not create enough separation at the top of the model field.

This ceiling effect matters. If several frontier models score 100%, then the benchmark cannot tell us which of those models has better sarcasm understanding. It can still show that weaker models may struggle with certain categories, but it cannot yet serve as a hard frontier benchmark.

The results also suggest that modern models are much better at this specific format than older sarcasm-detection studies might lead us to expect. The earlier binary sarcasm-detection work found that ChatGPT-4, ChatGPT-3.5, and Bard made different kinds of mistakes on sarcastic and non-sarcastic examples. In contrast, the current SarcBench v0.1 results show that many newer models can solve short, well-structured sarcasm interpretation questions very reliably.

This does not prove that sarcasm understanding is solved. SarcBench v0.1 uses short written contexts, clean answer choices, and controlled scenarios. Real sarcasm can depend on voice, relationship history, shared memory, timing, culture, or prior conversation. A multiple-choice benchmark can test one important part of sarcasm understanding, but not all of it.

The most useful interpretation is that SarcBench v0.1 validates the benchmark format, but not the benchmark difficulty. The next version should keep the same basic structure while making the private set harder.

## 7 Limitations

SarcBench v0.1 has several important limitations.

First, there is no formal human baseline yet. This means the current leaderboard should be described as a model-only pilot evaluation. It should not claim that models are above or below human performance. Human comparison should remain marked as pending until human participants complete the same private set under controlled instructions.

Second, the current private set is too easy for top models. Five evaluated models reached 100.0% Avg@5 and Maj@5. A benchmark with that many perfect scores is useful as a pilot, but not as a final frontier evaluation.

Third, the dataset is still small. The scored private set contains 125 questions. That is enough to test the pipeline and publish a pilot leaderboard, but a stronger benchmark should include more questions to reduce noise and improve category-level reliability.

Fourth, multiple-choice evaluation makes grading clean, but it also makes the task easier. The correct answer is always visible among the choices. A model may succeed by eliminating bad answers rather than fully generating the intended meaning itself.

Fifth, the current questions are written in clear, compact English. That helps with fairness and scoring, but it also removes many features that make sarcasm hard in the real world, such as long conversation history, ambiguous speaker relationships, cultural references, and tone.

Sixth, the item validation process is not yet as strong as it should be for a final benchmark. The

current items include internal quality checks, but they have not yet gone through independent human agreement testing. A future version should measure whether human annotators reliably choose the same answer.

## 8 Future Work

The next version of SarcBench should focus on making the benchmark harder, not larger for its own sake.

A stronger v0.2 should include more examples where the literal answer is tempting, but wrong. It should include sincere comments that look sarcastic, sarcastic comments without obvious cue words, and paired examples where one small context change flips the answer. It should also include more target-identification questions where multiple people or objects are plausible targets.

A good v0.2 should also include human validation. At minimum, each private question should be answered by several human participants. Items should be kept only when humans agree on the correct answer at a high rate. This would make the ground truth more defensible and allow the leaderboard to report a real human baseline.

Another useful direction is an open-ended version. Instead of giving six choices, the model could be asked to explain what the speaker means in one sentence. That would be harder to grade, but it would better match real-world interpretation. One compromise is to keep multiple-choice scoring for the official leaderboard while also collecting open-ended outputs for qualitative analysis.

Finally, SarcBench should be treated as a living benchmark. HELM argues for transparent and continuously updated evaluation, including shared scenarios and standardized model comparisons [3]. SarcBench can follow a similar idea on a smaller scale by keeping a stable public set, a hidden private set, a reserve set for future refreshes, and clear version numbers for every leaderboard release.

## 9 Conclusion

SarcBench v0.1 introduces a focused benchmark for testing sarcasm understanding in language models. Its main idea is simple: sarcasm evaluation should not stop at detecting whether sarcasm is present. A useful benchmark should ask whether the model understands what the speaker actually means.

The pilot results show that the current version is not difficult enough for the strongest models. Several frontier systems reached perfect scores, and two more came very close. This means SarcBench v0.1 should be presented honestly as a pilot leaderboard, not as a final proof that models struggle with sarcasm.

Even with that limitation, the project has a clear contribution. It defines a sarcasm-specific multiple-choice format, separates the task into meaningful categories, evaluates models over repeated runs, and produces a reproducible leaderboard. The next step is to build a harder v0.2 with stronger human validation and a formal human baseline.

SarcBench v0.1 answers one question clearly: the pipeline works. The next version should answer a

harder question: where do strong models still misunderstand what people really mean?

## References


[1] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A Large Self-Annotated Corpus for Sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, 2018.

[3] Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*, 2023. https://openreview.net/forum?id=iO4LZibEqW.

[4] Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles, 2016. Association for Computational Linguistics.

[5] Philip and Hemang. SimpleBench: The Text Benchmark in which Unspecialized Human Performance Exceeds that of Current Frontier Models. Google Docs report, October 31, 2024. https://drive.google.com/file/d/1mddNFK5UbBFVr3oDftd2Kyc6D8TFctfe/view?usp=embed_facebook.

[6] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models. *Transactions on Machine Learning Research*, 2023. https://openreview.net/forum?id=uyTL5Bvosj.

[7] Navadeep Budda and Naveen Budda. A Comparative Analysis of ChatGPT-4, ChatGPT-3.5, and Bard (Gemini Pro) in Sarcasm Detection. *Journal of Student Research*, 13(2), 2024. DOI: 10.47611/jsrhs.v13i2.6497.

[8] Natural Language and Dialogue Systems. Sarcasm Corpus V2. https://nlds.soe.ucsc.edu/sarcasm2.

[9] Hyewon Jang and Diego Frassinelli. Generalizable Sarcasm Detection is Just Around the Corner, of Course! In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4238–4249, Mexico City, Mexico, 2024. Association for Computational Linguistics. https://aclanthology.org/2024.naacl-long.238/.

[10] Dan Hendrycks et al. GitHub repository for Measuring Massive Multitask Language Understanding. https://github.com/hendrycks/test.