

# Definition Modeling for Mathematics: Understanding Audience Bias and its Problem Solving Relevance

Carolina Dias-Alexiou, Edison Marrese-Taylor, Alfredo Solano, Yutaka Matsuo

Graduate School of Engineering, The University of Tokyo

{carolina.dias,emarrese,asolano,matsuo}@weblab.t.u-tokyo.ac.jp

## Abstract

Definition modeling is used as a probe of what a language model has internalized about a domain, and it has been studied on general lexicography and on jargon-heavy specialist domains. We extend this task to mathematics, where a term can be defined very differently for different communities. In this context we study the baseline capacity of models to retrieve generate definitions, and their audience-bias when explaining mathematical terms without specific guidance. We further asses to what extent can they follow a certain style when directed, and to which degree being exposed to definitions aid or hurt model performance when solving mathematical questions.

## 1 Introduction

A mathematician, a Wikipedia editor, and a general audience speaker, when asked to define a mathematical term like ‘*Klein bottle*’ will likely give different definitions that can all be correct. Definitions are often not a single normalized strings per term, instead are shaped by readership and purpose, and we argue that is this of particular importance in mathematics. This makes such a domain a useful setting for separating whether a model “knows” a concept from whether it can explain it to a specified community.

Definition modeling was introduced as a probe of lexical representations (Noraset et al., 2017) and extended to contextual, multilingual, scientific, and specialist settings (Ishiwatari et al., 2019; August et al., 2022; Marrese-Taylor et al., 2025; Jhirad et al., 2023). We ask whether the same probe transfers to mathematics, where common concepts may be well represented in pretraining but their community-specific presentations differ sharply. The same definition bank also provides a direct test of definition-only retrieval: if the model was already exposed to a term, adding its definition may add distraction rather than information.

We align 2,963 headwords from the Concise Oxford Dictionary of Mathematics (Clapham and Nicholson, 2014) to entries on the English Wikipedia, the Simple English Wikipedia, nLab (a wiki for collaborative work on Mathematics, especially from the point of view of category theory), and the subreddit r/explainlikeimfive. We then two tasks designed to help shed light into the issue at hand: definition generation for both audience-agnostic and audience-tailored, and MMLU mathematics question answering with retrieved definitions. Our experiment utilize three different models: Qwen2.5-7B-Instruct, GPT-5-mini, and GPT-5.2.

## 2 Related Work

### Definition modeling and controllable definitions.

Definition modeling was introduced as a readout of lexical representations (Noraset et al., 2017) and surveyed by Gardner et al. (2022). Later work added contextual disambiguation (Gadetsky et al., 2018; Ishiwatari et al., 2019), generative dictionary systems (Periti et al., 2024; Tuan et al., 2024), scientific definitions with controllable complexity (August et al., 2022), and audience-controlled dictionary examples (He and Yiu, 2022). We extend this line to mathematical concepts and to five naturally occurring community registers rather than one complexity scale.

**Knowledge and retrieval.** Work on knowledge awareness asks whether a model can distinguish concepts it knows from concepts it does not (Ferrando et al., 2025); related work studies comprehension of newly emerging concepts under distribution shift (Mei et al., 2024). Retrieval-augmented generation (RAG) (Lewis et al., 2020) and selective variants (Asai et al., 2024; Jeong et al., 2024) likewise recognize that retrieved context is not always useful. We provide a controlled test in which retrieval is fixed and only definition text varies, con-

| Register | Readership          | Terms | Med. len. |
|----------|---------------------|-------|-----------|
| OXF      | reference, terse    | 2,963 | 30        |
| WIKI     | general             | 2,229 | 160       |
| SIMP     | plain language      | 1,038 | 95        |
| NLAB     | research math       | 685   | 110       |
| ELI5     | lay, conversational | 321   | 125       |

Table 1: The five-register corpus. Median length in words defines the audience-tailored length target.

necting term knowledge and MMLU mathematics performance.

### 3 Setup

#### 3.1 Data

We begin with 2,963 headwords from the Concise Oxford Dictionary of Mathematics (OXF) (Clapham and Nicholson, 2014). For each headword, we collect independently authored text from four communities: the English Wikipedia article (general encyclopedic) denoted as WIKI, the Simple English Wikipedia (plain language) denoted as SIMP, nLab (research mathematics with a category-theoretic perspective) referred to as NLAB, and an the content of the subreddit r/explainlikeimfive answers (lay, conversational), denoted ELI5.

Starting from OXF as a fixed vocabulary of terms, we built parallel term-aligned corpora against the other sources via a candidate-generation and LLM-adjudication pipeline. For each source, candidates were first retrieved with a lightweight, source-appropriate method. An LLM agent then judged, per term, whether any candidate named the same mathematical concept, applying source-specific disambiguation rule.

For WIKI, SIMP and NLAB, this judgment was followed by a second verification stage in which a another agent re-adjudicated each accepted pair, now grounded in the candidate’s actual lead text. Finally, because a Reddit thread’s title is uninformative about its content, its single judgment stage instead conditioned directly on the thread’s answer text and emitted a high/medium/low confidence label rather than going through a separate verification pass. Only confirmed/related pairs from each source were retained and joined back onto the Oxford vocabulary by term to produce the final parallel corpus. Please see §F for details.

Table 1 summarizes the details of the data we collected and utilized. Length values reported in the table for WIKI and SIMP are only for the first

paragraph of the article, which is what we employ for the main experiments in this paper, unless otherwise specified. Even under this scenario, we note that length and technicality/complexity are not strictly correlated. We see that both the nLab and the subreddit entries have a very similar length (110 and 125 words respectively) albeit having almost opposite audience as targets. We therefore avoid interpreting the registers as a single complexity scale.

Because term coverage depends on the source, our common-term analyses use explicit intersections. The *three-way* core contains 957 terms with Oxford, Wikipedia, and Simple Wikipedia text, and the *five-way* core contains 97 terms with text in all five registers. Please see §3.2 for details.

#### 3.2 Tasks

**Definition modeling.** We begin by testing a panel of models on the task of definition modeling in this domain via our collected data. We evaluate *Qwen2.5-7B-Instruct*, *GPT-5-mini*, and *GPT-5.2*. A bare-term prompt asks for a two-to-three-sentence audience-agnostic definition. In a second, tailored condition, the prompt names the target readership and gives its median-derived word-count target. All generations are scored against each gold register using BLEU, METEOR, COMET, and Baseline-rescaled BERTScore, plus an LLM-as-a-Judge approach based on GPT-5.2 and Gemini-3.5-Flash for correctness and register-fit on a 0–3 scale.

**MMLU mathematics performance.** We use 1,189 test questions from six MMLU mathematics subjects (Hendrycks et al., 2020) to assess whether having access to gold-standard definitions helps models solve the questions correctly. Since there is no gold-standard dataset of question-term relations for MMLU, we fuse an TF-IDF and dense retrieval approach with a cross-encoder re-ranker to produce ten candidate terms related to each question. The term-question mapping was derived independently by two LLM judges over all 1,189 questions, each verifying every selected term against the same three-criterion rubric (definitional support, necessity, semantic role) and flagging required terms missed among the rejected candidates. For this we utilized GPT-5.2 (orig.), which we also compare against Gemini-3.5-Flash (Judge A) and GPT-5-mini (Judge B). Table 2 shows the details.

As shown in the table, both additional judges

| Comparison          | EM   | Jac.  | $\kappa$ | P     | R     |
|---------------------|------|-------|----------|-------|-------|
| orig. vs. judge A   | 70.9 | 0.772 | 0.840    | 0.968 | 0.773 |
| orig. vs. judge B   | 51.1 | 0.599 | 0.629    | 0.911 | 0.525 |
| judge A vs. judge B | 61.7 | 0.702 | 0.680    | —     | —     |

Table 2: Validation of question-to-term mapping, where EM stands for Exact Match, Jac. is short for Jaccard Index,  $\kappa$  indicates Cohen’s Kappa, and P and R are short for Precision and Recall, computed assuming GPT-5.2 as gold standard.

reach high agreement with the original annotation, largely by confirming its selections (precision  $\approx 0.9$ ) while trimming about a quarter of them. Judge B applies a stricter necessity criterion, removing roughly half of the original’s picks. While GPT-5.2 and Gemini-3.5-Flash have both high Jaccard Index ( $0.736 \sim 0.834$ ) and high Exact Match ( $69.1\% \sim 72.9\%$ ) for all subjects, GPT-5-mini has a highest disagreement with GPT-5.2. Its per-subject Jaccard is highest on `high_school_statistics` (0.666) and `elementary_mathematics` (0.648) and lowest on `abstract_algebra` (0.540) and `college_mathematics` (0.485), but more striking is the difference in the Exact match which drops to 35.7% and 35.0% for `formal_logic` and `college_mathematics` respectively. The mapping final averages 1.37 terms per question, with 230 no-term questions. Although we did not test this empirically, we believe this suggest results would not change substantially when using a similarly-capable model for this kind of annotation.

For the RAG stage, retrieval is run once over the OXF text and the same top-three term identifiers are reused in every condition. Only the injected text changes: none, Oxford, Wikipedia lead, or Simple Wikipedia. This isolates definition register from retrieval quality. As mentioned earlier, the full Wikipedia article was tested separately, please see below.

## 4 Results

### 4.1 Definition Modeling

Table 3 summarizes our results on the definition modeling experiments. Although all entries have relatively low values (non greater than 5), WIKI shows the highest BLEU scores, especially for Qwen2.5-7B and GPT-5-mini. This is true for both agnostic and tailored answers. For the latter, even GPT-5.2 shows a relatively high value of BLEU. This could indicate a bias in terms of

model exposure during training. We also noticed that Qwen2.5-7B retains the strongest BERTScore in every register.

Comparing between agnostic and tailored scenarios, Qwen2.5-7B on NLAB correctness falls to 0.99/3 over 684 tailored terms, well below its own general NLAB score, while retaining near-ceiling register-fit on ELI5 (2.98) despite the panel’s lowest correctness there. We think this is not a knowledge gap since, when scored under the plain, audience-agnostic prompt on the full 684-term NLAB set, Qwen2.5-7B scores 2.32/3. The failure appears only once the prompt asks it to write in the NLAB register.

Since the correctness judge is GPT-5.2, one of the three models it scores, we independently re-judged all (model, register, term) triples with Gemini-3.5-Flash. The two judges exactly agree on 79.6% of items, with Cohen’s  $\kappa = 0.313$  (unordered categories) and Kendall’s  $\tau = 0.487$ . Gemini is uniformly more lenient (mean score 2.88 vs. 2.66), concentrated in upgrading a 2 to a 3 rather than reversing outright failures, so the two judges rarely disagree about which generations are broken, only about how generously to score borderline ones. Please see detailed results in §A.

Table 4 shows the comparison of each tailored generation against all five golds on the terms for which we have definitions at all levels (the *five-way* core) using the correctness measure. Rows are the register the generation was written for; columns are the gold it is scored against.

The most immediate observation is that there is little variation between the values, although with a few exceptions. This shows a potential limitation our LLM-as-a-Judge evaluation, since one would expect values to be more concentrated/higher for a single audience. We think this could be an indication that our tested models are not sufficiently good at making this kind of judgments, but we reckon that it can also be due our limited budget in terms of prompt design and optimization.

Overall, we see tailoring to a named audience is close to free on OXFORD and WIKIPEDIA ( $|\Delta\text{Corr}| \leq 0.16$  and  $\leq 0.06$ , respectively) and increasingly costly moving to SIMPLE ( $-0.19$  to  $-0.54$ ), ELI5 ( $-0.45$  to  $-0.83$ ), and NLAB ( $-0.27$  to  $-1.33$ ). The losses are generally smallest for the two API models and largest for Qwen2.5-7B.

When limited to cross-register scoring on the *five-way* core, we can see more clearly that the

| Reg. | Model      | Agnostic |       |       | Tailored |        |       | Reg  |
|------|------------|----------|-------|-------|----------|--------|-------|------|
|      |            | BL       | BS    | Corr. | BL       | BS     | Corr. |      |
| OXF  | Qwen2.5-7B | 1.78     | 5.59  | 2.44  | 1.24     | 6.97   | 2.34  | 2.26 |
|      | GPT-5-mini | 1.36     | 0.91  | 2.64  | 0.50     | -3.96  | 2.49  | 2.71 |
|      | GPT-5.2    | 1.00     | 1.34  | 2.63  | 0.30     | -4.33  | 2.55  | 2.80 |
|      |            |          |       |       |          |        |       |      |
| WIKI | Qwen2.5-7B | 3.43     | 4.08  | 2.59  | 4.55     | 4.42   | 2.50  | 2.70 |
|      | GPT-5-mini | 3.40     | -1.70 | 2.82  | 4.63     | 2.88   | 2.78  | 2.92 |
|      | GPT-5.2    | 1.64     | -6.23 | 2.80  | 3.09     | 0.91   | 2.78  | 2.99 |
|      |            |          |       |       |          |        |       |      |
| SIMP | Qwen2.5-7B | 3.22     | 5.17  | 2.67  | 1.96     | 0.27   | 2.13  | 2.90 |
|      | GPT-5-mini | 2.56     | -2.66 | 2.81  | 2.59     | 2.12   | 2.63  | 2.78 |
|      | GPT-5.2    | 1.31     | -5.76 | 2.81  | 1.68     | 1.08   | 2.53  | 2.93 |
|      |            |          |       |       |          |        |       |      |
| NLAB | Qwen2.5-7B | 1.37     | 0.54  | 2.32  | 0.81     | -1.90  | 0.99  | 1.90 |
|      | GPT-5-mini | 1.62     | -1.60 | 2.65  | 1.09     | -10.53 | 2.10  | 2.88 |
|      | GPT-5.2    | 0.61     | 0.57  | 2.59  | 0.53     | -7.14  | 2.31  | 2.94 |
|      |            |          |       |       |          |        |       |      |
| ELI5 | Qwen2.5-7B | 1.42     | 2.43  | 2.82  | 0.77     | 0.27   | 1.99  | 2.98 |
|      | GPT-5-mini | 1.29     | -3.19 | 2.95  | 0.70     | -1.89  | 2.50  | 3.00 |
|      | GPT-5.2    | 0.55     | -7.06 | 2.94  | 0.60     | -2.78  | 2.48  | 3.00 |
|      |            |          |       |       |          |        |       |      |

Table 3: BLEU (BL), Baseline-rescaled BERTScore (BS, all  $\times 100$ ), and Correctness and tailored Register-fit (Reg.) via GPT-5.2 for audience-agnostic and audience-tailored generation; judge scores are 0  $\sim$  3. Scores are computed over each register’s native available term set. Baseline-rescaled BERTScore may be negative.

NLAB-tailored generation itself seems to be the reason of the failure: Qwen2.5-7B’s NLAB-tailored row is the worst row of the matrix against every gold, not only against NLAB’s own.

We inspected the low-scoring outputs to better understand why this may happen. We found that for every one of the 547 terms whose gold definition uses no categorical-theoretic vocabulary, Qwen2.5-7B’s nLab-tailored generation nonetheless forces one, e.g. defining *acceleration* as a “2-morphism between functors” rather than the “second derivative of a curve”. The model over-applies the requested register rather than omitting content it lacks. An explicit guardrail against inventing structure recovered over half of the gap (correctness 0.99  $\rightarrow$  1.57) without a compensating drop in register-fit, and a held-out real exemplar gives a smaller, Qwen-specific gain (0.99  $\rightarrow$  1.18); neither fully restores plain-prompt performance (please see §B for details).

## 4.2 MMLU Mathematics Performance

For our RAG experiments, against the three-way corpus ( $n=1,189$ ), we split questions into two cohorts: *has-term* ( $n=549$ ) questions, whose required term falls inside that corpus, so the definition retrieved and injected is genuinely the one the question needs; and *no-term* ( $n=640$ ) questions, for which any injected definition is necessarily off-target. This split lets us test whether RAG helps

|           | Model      | OXF         | WIKI        | SIMP        | NLAB        | ELI5        |
|-----------|------------|-------------|-------------|-------------|-------------|-------------|
| OXFORD    | Qwen2.5-7B | <b>2.71</b> | 2.67        | 2.77        | 2.55        | 2.81        |
|           | GPT-5-mini | <b>2.81</b> | 2.86        | 2.84        | 2.72        | 2.87        |
|           | GPT-5.2    | <b>2.86</b> | 2.88        | 2.90        | 2.70        | 2.90        |
|           |            |             |             |             |             |             |
| WIKIPEDIA | Qwen2.5-7B | 2.43        | <b>2.61</b> | 2.66        | 2.30        | 2.70        |
|           | GPT-5-mini | 2.75        | <b>2.81</b> | 2.88        | 2.56        | 2.89        |
|           | GPT-5.2    | 2.64        | <b>2.76</b> | 2.79        | 2.45        | 2.78        |
|           |            |             |             |             |             |             |
| SIMPLE    | Qwen2.5-7B | 1.98        | 2.07        | <b>2.25</b> | 1.93        | 2.20        |
|           | GPT-5-mini | 2.44        | 2.60        | <b>2.70</b> | 2.26        | 2.74        |
|           | GPT-5.2    | 2.42        | 2.54        | <b>2.68</b> | 2.25        | 2.68        |
|           |            |             |             |             |             |             |
| NLAB      | Qwen2.5-7B | 1.08        | 1.18        | 1.19        | <b>1.21</b> | 1.15        |
|           | GPT-5-mini | 2.14        | 2.21        | 2.16        | <b>2.24</b> | 2.25        |
|           | GPT-5.2    | 2.20        | 2.36        | 2.31        | <b>2.36</b> | 2.35        |
|           |            |             |             |             |             |             |
| ELI5      | Qwen2.5-7B | 1.65        | 1.65        | 1.86        | 1.55        | <b>1.76</b> |
|           | GPT-5-mini | 1.98        | 2.09        | 2.18        | 1.80        | <b>2.15</b> |
|           | GPT-5.2    | 1.93        | 1.98        | 2.07        | 1.76        | <b>2.15</b> |
|           |            |             |             |             |             |             |

Table 4: Correctness judge (0  $\sim$  3) for every (written-for, scored-against) register pair on the *five-core* ( $n=97$ ). Bold marks the Correctness judgment for the matching target and underline marks the uniformly worst row.

| Model      | Cohort | Base  | OXF   | WIKI   | SIMP   |
|------------|--------|-------|-------|--------|--------|
| Qwen2.5-7B | all    | 53.32 | -0.17 | -1.09  | -2.27  |
|            | has    | 49.73 | +0.91 | +2.37  | -1.28  |
|            | no     | 56.41 | -1.09 | -4.06* | -3.12  |
| GPT-5-mini | all    | 65.01 | -1.09 | -0.76  | -1.35  |
|            | has    | 64.30 | -1.64 | -1.82  | -3.64* |
|            | no     | 65.62 | -0.62 | +0.16  | +0.62  |
| GPT-5.2    | all    | 74.26 | -0.42 | -0.08  | -0.17  |
|            | has    | 72.86 | -1.09 | +0.73  | -0.73  |
|            | no     | 75.47 | +0.16 | -0.78  | +0.31  |

Table 5: MMLU-math accuracy change (percentage points) from injecting a retrieved definition. Retrieval is fixed. Cohorts are all ( $n = 1,189$ ), has-term ( $n = 549$ ), and no-term ( $n = 640$ ). \* marks a paired-bootstrap 95% CI excluding zero; all such effects are losses.

specifically where its content is relevant and only distracts where it is not.

As shown in Table 5, no model gains significantly from any short definition register. Qwen2.5-7B has the arguably expected split on relevant versus irrelevant questions, but only the irrelevant WIKI condition survives the bootstrap test ( $-4.06$  points, 95% CI  $[-7.66, -0.62]$ ). GPT-5-mini is significantly harmed on has-term questions by SIMPLE; GPT-5.2 is statistically flat. Thus, relevance alone does not guarantee benefit. In this benchmark, correct definitions are usually redundant and can become distraction.

We also observed that the effect is subject-dependent. Qwen2.5-7B benefits most in concept-dense subjects such as abstract algebra and for-

mal logic, while all three short-definition registers hurt elementary mathematics. Manual inspection suggests that the latter questions often mention a related operation without requiring its definition, exposing a limitation of the LLM term-necessity annotation (full breakdown in § D).

Finally, more context is substantially worse for the smaller model. Replacing the roughly 1,200-character WIKIPEDIA lead with the entire article changes Qwen’s accuracy by  $-9.34$  points overall, while GPT-5-mini loses 1.93 points. Raw-output tracing shows genuine incoherent multilingual generation under the extended context, not a scoring artifact. For Qwen, this significantly negative effect may be due to enabling extended context via YaRN scaling (Peng et al., 2024).

## 5 Conclusion

We extended definition modeling to mathematics with five community registers and paired it with definition-only retrieval for MMLU mathematics. Surface metrics and correctness judges disagree systematically; Qwen’s nLab failure is a register-conditioning instruction-following problem rather than a knowledge gap, and is largely, though not fully, corrected by explicit prompt guardrails; and short retrieved definitions produce no reliable gains, while excessive context causes substantial degradation. These results argue for multi-register evaluation and targeted, compact context rather than treating one reference definition or more retrieved text as universally helpful.

## Limitations

**Self-judging.** We GPT-5.2 as a judge for both experiments, but this is also one of the three models being evaluated, so a third of every judged comparison is a self-assessment. The judge-score band across models is narrow (2.63–2.94 audience-agnostic), and we cannot rule out that leniency toward its own and its sibling’s outputs contributes to it.

The reference-based metrics are judge-independent and show the same model ordering, and Qwen2.5-7B’s nLAB failure. These results show that GPT-5.2 has no incentive to favor is additionally corroborated by a directly inspectable mechanism. A replication with a judge outside the panel would nonetheless be worth running.

In our experiments, each audience instruction is paired with a word-count target derived from that

register’s own gold median, so the tailored generations differ from the audience-agnostic ones in both register and length. We therefore cannot separate the instruction effect from the length constraint.

Only 23% of headwords have an nLAB page and 11% an ELI5 thread, and neither subset is random: nLAB skews toward algebraic and structural mathematics, ELI5 toward terms popular enough to be asked about on a general forum. Comparisons across registers are therefore comparisons across different, non-exchangeable term sets, except on the 97-term five-way core.

Three models is a thin panel, with a single open model at 7B and no mid-scale open models between it and the two API models, so we cannot tell whether Qwen2.5-7B’s nLAB failure is a scale effect, a Qwen-family effect, or specific to this checkpoint. Furthermore, the retrieval deltas in §4.2 are small relative to sampling error at  $n=549-1,189$ ; we report which ones survive a paired bootstrap and confine our claims to those, but the study is not powered to detect gains below roughly 3pp at the cohort level.

Finally, the term-question mapping is produced by an LLM annotator and validated only by two further LLM judges (Table 2); no human review of the 1,189-question mapping has yet been completed, and the elementary\_mathematics evidence in §4.2 suggests the necessity criterion is looser than intended in the procedural regime.

## References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations*.
- Tal August, Katharina Reinecke, and Noah A. Smith. 2022. [Generating scientific definitions with controllable complexity](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8298–8317.
- Christopher Clapham and James Nicholson. 2014. *The Concise Oxford Dictionary of Mathematics*, 5 edition. Oxford University Press.
- Javier Ferrando, Oscar Balcells Obeso, Senthoran Rajamanoharan, and Neel Nanda. 2025. [Do i know this entity? knowledge awareness and hallucinations in language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. [Conditional generators of words definitions](#).

- In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271.
- Noah Gardner, Hafiz Khan, and Chih-Cheng Hung. 2022. [Definition modeling: literature review and dataset analysis](#). *Applied Computing and Intelligence*, 2(1):83–98.
- Xingwei He and Siu Ming Yiu. 2022. [Controllable dictionary example generation: Generating example sentences for specific targeted audiences](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 610–627.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. [Measuring massive multitask language understanding](#). *arXiv preprint arXiv:2009.03300*.
- Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. [Learning to describe unknown phrases with local and global contexts](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3467–3476.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. [Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 7036–7050.
- James Jhirad, Edison Marrese-Taylor, and Yutaka Matsuo. 2023. [Evaluating large language models’ understanding of financial terminology via definition modeling](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 93–100.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Edison Marrese-Taylor, Erica K. Shimomoto, Alfredo Solano, and Enrique Reid. 2025. [Multilingual definition modeling](#). In *Findings of the Association for Computational Linguistics: ACL 2025*.
- Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, and Xueqi Cheng. 2024. [SLANG: New concept comprehension of large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12558–12575.
- Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. [Definition modeling: Learning to define word embeddings in natural language](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. [YaRN: Efficient context window extension of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Francesco Periti, David Alfter, and Nina Tahmasebi. 2024. [Automatically generated definitions and their utility for modeling word meaning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.
- Kai-Wen Tuan, Hai-Lun Tu, and Jason S. Chang. 2024. [Generative dictionary: Improving language learner understanding with contextual definitions](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 390–396.

## A Full Definition-Modeling Results

Table 6 reports all reference metrics (BLEU, METEOR, COMET, BERTScore) alongside both correctness judges (GPT-5.2 and gemini-3.5-flash,  $\text{Corr}_G$ ) for the audience-agnostic condition, and all reference metrics alongside correctness and register-fit for the audience-tailored condition, expanding the summary in Table 3.

## B Audience Instructions

Each register’s tailored prompt appends one instruction and one length target (Table 1) to the bare-term prompt. Abridged: OXFORD — “precise, formal, dictionary-style, for a reader with mathematical training; be terse and exact; do not simplify, add examples, or add motivation”. WIKIPEDIA — “standard encyclopedia-style explanation for an educated general reader with no specialized mathematics background, like a Wikipedia opening paragraph: define the term and give brief context”. SIMPLE — “simple explanation for a non-specialist or early learner; plain vocabulary and short sentences; avoid jargon”. NLAB — “rigorous, structural explanation for a research mathematician familiar with category theory; use categorical or structural language and, where natural, situate the notion in a more general setting”. ELI5 — “conversational, intuitive explanation for a curious adult with no mathematics background, in the style of an r/explainlikeimfive answer; use everyday language and an analogy if it helps; avoid equations and jargon”.

The one-shot exemplar for each register is the gold entry whose length is closest to that register’s median, excluding the five-way core; it is held out from that register’s evaluation set.

## C Retrieval Confidence Intervals

Table 9 gives the paired-bootstrap 95% CIs (2,000 resamples over per-question correctness, resampling questions so that the baseline and RAG runs stay paired) behind Table 5.

## D Per-Subject Breakdown

Table 10 breaks Qwen2.5-7B’s has-term cohort down by MMLU subject. Cell sizes are 63–120 questions, so individual deltas are exploratory; the pattern of interest is that elementary\_mathematics is negative at all three registers while the concept-dense, low-baseline subjects are mostly positive.

## E Infrastructure Notes

**Reasoning-effort truncation.** GPT-5-mini defaults its hidden reasoning effort to a level that, on short-answer prompts with a modest output budget, consumes the entire budget before emitting any visible text; the response returns incomplete and empty. Left unfixed this silently collapsed every prediction to the first option and produced a spurious 21% baseline. We pin the lowest effort level each variant supports and raise the output-token floor for all API calls in both studies.

**Long-context serving.** The full-article condition requires more than Qwen2.5-7B’s native 32K context; we serve it with static YaRN scaling (Peng et al., 2024) at 131K. No article needed truncation at that budget (longest  $\approx 351$ K characters). vLLM’s echo-plus-logprob scoring path fails server-side at these lengths with a non-finite value error in both fp16 and bf16, so all full-article runs score multiple-choice questions by generation and letter extraction instead of per-option logprobs.

## F Term Matching

### F.1 English Wikipedia

Candidates were generated per letter by fuzzy-matching each Oxford term (via difflib.SequenceMatcher) against Wikipedia’s mathematics-article title list, keeping the top 5. An LLM judge then picked the single best title-only match using explicit disambiguation rules (rejecting narrower/broader pages and cross-subfield homonyms), after which a second agent re-verified each accepted pair against the article’s real fetched lead paragraph, producing a confirmed/related/wrong verdict. Any pair left unconfirmed went through a repair stage in which a tool-using agent actively searched Wikipedia for a better target article before giving up.

For the case of the Simple English Wikipedia, this structurally identical to the above. We use same fuzzy-matching candidate generation and the exact same judge and verify prompts — with only the target host swapped (simple.wikipedia.org). However, no repair/repoint stage was run for this source.

### F.2 nLab

Since the nLab exposes no search API, candidates were generated by matching each Oxford term locally against the wiki’s complete page-name index,

| Register | Model      | Agnostic |       |       |       |      |                   | Tailored |       |       |        |      |      |
|----------|------------|----------|-------|-------|-------|------|-------------------|----------|-------|-------|--------|------|------|
|          |            | BL       | MET   | COM   | BS    | Corr | Corr <sub>G</sub> | BL       | MET   | COM   | BS     | Corr | Reg  |
| OXF      | Qwen2.5-7B | 1.78     | 18.11 | 56.96 | 5.59  | 2.44 | 2.72              | 1.24     | 13.60 | 54.15 | 6.97   | 2.34 | 2.26 |
|          | GPT-5-mini | 1.36     | 17.89 | 55.64 | 0.91  | 2.64 | 2.85              | 0.50     | 12.02 | 51.24 | -3.96  | 2.49 | 2.71 |
|          | GPT-5.2    | 1.00     | 15.90 | 56.67 | 1.34  | 2.63 | 2.86              | 0.30     | 9.70  | 51.29 | -4.33  | 2.55 | 2.80 |
| WIKI     | Qwen2.5-7B | 3.43     | 16.66 | 62.04 | 4.08  | 2.59 | 2.85              | 4.55     | 22.72 | 68.90 | 4.42   | 2.50 | 2.70 |
|          | GPT-5-mini | 3.40     | 17.55 | 60.32 | -1.70 | 2.82 | 2.95              | 4.63     | 24.68 | 69.62 | 2.88   | 2.78 | 2.92 |
|          | GPT-5.2    | 1.64     | 13.56 | 58.06 | -6.23 | 2.80 | 2.96              | 3.09     | 23.24 | 69.72 | 0.91   | 2.78 | 2.99 |
| SIMP     | Qwen2.5-7B | 3.22     | 18.19 | 62.90 | 5.17  | 2.67 | 2.92              | 1.96     | 16.57 | 58.90 | 0.27   | 2.13 | 2.90 |
|          | GPT-5-mini | 2.56     | 18.29 | 59.55 | -2.66 | 2.81 | 2.96              | 2.59     | 20.63 | 63.40 | 2.12   | 2.63 | 2.78 |
|          | GPT-5.2    | 1.31     | 14.56 | 58.38 | -5.76 | 2.81 | 2.97              | 1.68     | 18.66 | 62.38 | 1.08   | 2.53 | 2.93 |
| NLAB     | Qwen2.5-7B | 1.37     | 14.74 | 57.11 | 0.54  | 2.32 | 2.75              | 0.81     | 16.27 | 59.66 | -1.90  | 0.99 | 1.90 |
|          | GPT-5-mini | 1.62     | 15.97 | 57.02 | -1.60 | 2.65 | 2.94              | 1.09     | 17.18 | 54.91 | -10.53 | 2.10 | 2.88 |
|          | GPT-5.2    | 0.61     | 12.90 | 59.04 | 0.57  | 2.59 | 2.93              | 0.53     | 14.74 | 56.24 | -7.14  | 2.31 | 2.94 |
| ELI5     | Qwen2.5-7B | 1.42     | 12.81 | 58.22 | 2.43  | 2.82 | 2.98              | 0.77     | 16.69 | 58.18 | 0.27   | 1.99 | 2.98 |
|          | GPT-5-mini | 1.29     | 13.31 | 57.02 | -3.19 | 2.95 | 2.98              | 0.70     | 18.03 | 58.71 | -1.89  | 2.50 | 3.00 |
|          | GPT-5.2    | 0.55     | 10.80 | 55.35 | -7.06 | 2.94 | 2.98              | 0.60     | 16.61 | 59.52 | -2.78  | 2.48 | 3.00 |

Table 6: BLEU (BL), METEOR (MET), COMET (COM), BERTScore (BS, all  $\times 100$ ), correctness (Corr), and tailored register-fit (Reg) for audience-agnostic and audience-tailored generation, judged at a 2,000-character cap; judge scores are 0–3. Corr is judged by GPT-5.2; Corr<sub>G</sub> is the same agnostic generations judged by gemini-3.5-flash (§4.1). Scores use native available term sets; nLab and ELI5 tailored runs contain 684 and 320 scored terms because one term per register is held out for the one-shot ablation. Negative BERTScore values arise from baseline rescaling.

| Model / tier      | Plain       | Guarded            | Scaffold           |
|-------------------|-------------|--------------------|--------------------|
| Qwen2.5-7B / NLAB | 0.99 / 1.90 | <b>1.57 / 1.85</b> | 1.52 / 1.84        |
| GPT-5-mini / NLAB | 2.07 / 2.89 | <b>2.49 / 2.54</b> | 2.29 / 2.94        |
| GPT-5.2 / NLAB    | 2.30 / 2.92 | <b>2.67 / 2.49</b> | 2.46 / 2.97        |
| Qwen2.5-7B / ELI5 | 1.99 / 2.98 | 2.17 / <b>2.96</b> | <b>2.26 / 2.94</b> |
| GPT-5-mini / ELI5 | 2.52 / 3.00 | 2.70 / 2.92        | 2.63 / 2.98        |
| GPT-5.2 / ELI5    | 2.54 / 3.00 | <b>2.73 / 2.96</b> | 2.58 / 3.00        |

Table 7: Correctness / register-fit (0–3) for the plain audience-tailored prompt versus two guardrail variants, at the 2,000-character cap. Guarded forbids inventing structure the term does not need; scaffold requires one plain content sentence first. Neither fully restores plain-prompt (audience-agnostic) correctness, but guarded closes most of Qwen2.5-7B’s NLAB gap without a register-fit cost.

then probing terms with no match via HEAD requests to recover synonym redirects the index does not list. An LLM judge picked the best candidate from page names alone, using rules tuned to the nLab’s category-theoretic style (a more abstract page can still be the right match) and its main failure mode (a same-cluster redirect, e.g. “ceiling”, “floor”, is not a match). A second agent then re-verified each match against the page’s real fetched prose, with added handling for biography pages and content-free “stub” pages.

| Model / NLAB | Corr. 0→1-shot             | Reg. 0→1-shot      |
|--------------|----------------------------|--------------------|
| Qwen2.5-7B   | 0.99 → <b>1.18 (+0.19)</b> | 1.90 → 1.90 (flat) |
| GPT-5-mini   | 2.10 → 2.13 (+0.03)        | 2.88 → 2.87 (flat) |
| GPT-5.2      | 2.31 → 2.25 (−0.06)        | 2.94 → 2.91 (flat) |

Table 8: One-shot exemplar ablation on NLAB ( $n=684$ ). The exemplar helps only Qwen2.5-7B, and only content, not register-fit; a first release of this ablation used the corrupted exemplar at the 500-character cap and reported an inflated +0.33 for Qwen2.5-7B, which this table supersedes.

### F.3 Redddit ELI5

Candidates were found by searching Reddit per term and keeping only threads where the term appears as a whole phrase in the question title, restricted by default to Mathematics-flaired threads for precision. Because a thread’s title is a question rather than a description, the judge conditioned directly on the thread’s fetched best-answer text rather than the title, and — since that already grounds the judgment in real content — this source has no separate verification stage; the judge instead emits a high/medium/low confidence label in place of a confirmed/related/wrong verdict.

| Model      | Injected               | all ( $n=1,189$ )      | has ( $n=549$ )       | no ( $n=640$ )          |
|------------|------------------------|------------------------|-----------------------|-------------------------|
| Qwen2.5-7B | OXFORD                 | -0.17 [-2.27, +1.93]   | +0.91 [-1.82, +3.83]  | -1.09 [-4.06, +2.03]    |
|            | WIKIPEDIA              | -1.09 [-3.62, +1.43]   | +2.37 [-1.09, +5.83]  | -4.06* [-7.66, -0.62]   |
|            | SIMPLE                 | -2.27 [-4.88, +0.17]   | -1.28 [-4.74, +2.19]  | -3.12 [-6.72, +0.16]    |
|            | WIKIPEDIA <sub>F</sub> | -9.34* [-12.70, -5.80] | -4.37 [-9.47, +1.09]  | -13.59* [-18.13, -8.75] |
| GPT-5-mini | OXFORD                 | -1.09 [-3.20, +0.93]   | -1.64 [-4.92, +1.46]  | -0.62 [-3.28, +2.19]    |
|            | WIKIPEDIA              | -0.76 [-2.94, +1.35]   | -1.82 [-4.74, +1.28]  | +0.16 [-2.97, +2.97]    |
|            | SIMPLE                 | -1.35 [-3.36, +0.59]   | -3.64* [-6.56, -0.72] | +0.62 [-2.19, +3.28]    |
|            | WIKIPEDIA <sub>F</sub> | -1.93 [-3.95, +0.17]   | -3.64* [-6.74, -0.73] | -0.47 [-3.12, +2.19]    |
| GPT-5.2    | OXFORD                 | -0.42 [-2.35, +1.51]   | -1.09 [-4.01, +1.82]  | +0.16 [-2.34, +2.50]    |
|            | WIKIPEDIA              | -0.08 [-2.02, +1.85]   | +0.73 [-2.19, +3.83]  | -0.78 [-3.44, +1.88]    |
|            | SIMPLE                 | -0.17 [-2.02, +1.85]   | -0.73 [-3.64, +2.37]  | +0.31 [-2.34, +2.97]    |

Table 9: Accuracy change (pp) with paired-bootstrap 95% CIs. \* marks CIs excluding zero. No positive delta in the table is distinguishable from zero.

| Subject          | $n$ | Base | OXF   | WIKI  | SIMP  |
|------------------|-----|------|-------|-------|-------|
| abstract algebra | 66  | 37.9 | +9.09 | +9.09 | -7.58 |
| college maths    | 63  | 38.1 | -3.17 | +3.17 | -4.76 |
| formal logic     | 84  | 41.7 | +1.19 | +5.95 | +4.76 |
| HS maths         | 102 | 51.0 | +0.98 | +3.92 | -1.96 |
| HS statistics    | 120 | 52.5 | +1.67 | +2.50 | +5.00 |
| elementary maths | 114 | 64.9 | -2.63 | -6.14 | -6.14 |

Table 10: Qwen2.5-7B, has-term cohort only, accuracy change (pp) by subject, ordered by baseline accuracy.

| Subject                 | $n$ | Jaccard | Exact match |
|-------------------------|-----|---------|-------------|
| formal_logic            | 126 | 0.834   | 69.1%       |
| college_mathematics     | 100 | 0.812   | 71.0%       |
| high_school_statistics  | 216 | 0.795   | 71.3%       |
| abstract_algebra        | 100 | 0.783   | 70.0%       |
| elementary_mathematics  | 377 | 0.751   | 72.9%       |
| high_school_mathematics | 270 | 0.736   | 68.9%       |

Table 11: GPT-5.2 (original) vs. Gemini judge — overall  $\kappa = 0.840$ , Jaccard = 0.772, exact match = 70.9%

| Subject                 | $n$ | Jaccard | Exact match |
|-------------------------|-----|---------|-------------|
| elementary_mathematics  | 377 | 0.648   | 62.9%       |
| high_school_statistics  | 216 | 0.666   | 53.7%       |
| formal_logic            | 126 | 0.584   | 35.7%       |
| high_school_mathematics | 270 | 0.548   | 49.6%       |
| abstract_algebra        | 100 | 0.540   | 40.0%       |
| college_mathematics     | 100 | 0.485   | 35.0%       |

Table 12: GPT-5.2 (original) vs. gpt-5-mini judge — overall  $\kappa = 0.629$ , Jaccard = 0.599, exact match = 51.1%

| Subject                 | $n$ | Jaccard | Exact match |
|-------------------------|-----|---------|-------------|
| elementary_mathematics  | 377 | 0.819   | 79.8%       |
| high_school_mathematics | 270 | 0.701   | 65.2%       |
| high_school_statistics  | 216 | 0.683   | 56.0%       |
| formal_logic            | 126 | 0.606   | 40.5%       |
| abstract_algebra        | 100 | 0.597   | 45.0%       |
| college_mathematics     | 100 | 0.529   | 40.0%       |

Table 13: Gemini vs. gpt-5-mini judge — overall  $\kappa = 0.680$ , Jaccard = 0.702, exact match = 61.7%

### Wikipedia Judge Prompt (Stage 1)

You are matching dictionary entries from the Oxford Dictionary of Mathematics to Wikipedia  
→ article titles. Decide, for each dictionary entry, whether any candidate Wikipedia title  
→ refers to the SAME mathematical topic/concept.  
Read this JSON file: \${infile}  
Each element has "term", "definition" (may be truncated with ...), and "candidates" (up to 5  
→ Wikipedia titles with fuzzy scores you may ignore).  
For each entry pick the ONE candidate that refers to the same topic as term+definition, or null  
→ if none does.  
Rules for "same topic":  
- Singular/plural, word order, and disambiguation suffixes like "(mathematics)" DO count as the  
→ same topic (e.g. "antipodal points" == "Antipodal point").  
- A page whose title CONTAINS the term but is about a narrower/different concept is NOT a match (e.g. "angle" vs "Angle trisection"; "algebra" vs "Azumaya algebra"; "abacus" vs "Abacus school").  
- A broader page counts ONLY if it is genuinely the primary article for the term (e.g. "analysis" == "Mathematical analysis"); "affine geometry" vs "Geometry" is too broad, NOT a match.  
- Biographies of mathematicians match their person page, but not an unrelated person or a journal/organization.  
- Beware false friends: the same word can name different concepts in different subfields -- use → the definition to disambiguate.  
- When in doubt, prefer null over a wrong match.  
Write a JSON array to EXACTLY this path: \${outfile}  
Each element: {"term": <term>, "match": <chosen Wikipedia title string or null>, "confidence": "→ high"|"medium"|"low", "reason": <short phrase>}  
Keep every input term, in order. After writing the file, reply with just the count of non-null → matches.

Figure 1: Stage-1 judge prompt for the English Wikipedia source. The judge sees only candidate article *titles* (no article text) and picks the single best title-only match, or null. Reused verbatim, with host swapped, for Simple English Wikipedia.

### Wikipedia Verification Prompt (Stage 2)

You are verifying candidate matches between Oxford Dictionary of Mathematics entries and  
→ Wikipedia articles, using the ACTUAL Wikipedia article intro text.  
Read this JSON file: \${infile}  
Each element has "term", "definition" (full Oxford definition), "wikipedia\_title", "wiki\_intro" (→ actual lead paragraph(s)), "prior\_confidence" (ignore).  
For each element decide whether the Wikipedia article (per its intro) describes the SAME  
→ mathematical concept as the dictionary definition.  
Verdicts:  
- "confirmed" - clearly the same concept.  
- "related" - overlapping/adjacent but NOT the same concept (a sub-case, a broader field, a → different-but-related notion, or a disambiguation page).  
- "wrong" - different topic / wrong sense / homonym collision.  
Judge on MEANING using both the definition and the intro, not on title spelling. Watch for false → friends (same word, different concept in a different subfield). Singular/plural and → disambiguation suffixes are still the same concept.  
Write a JSON array to EXACTLY this path: \${outfile} Each element: {"term": <term>, "→ wikipedia\_title": <title>, "verdict": "confirmed"|"related"|"wrong", "note": <short reason → grounded in the intro>}  
Keep all input elements, in order. After writing, reply with just a tally "confirmed=X related=Y → wrong=Z".

Figure 2: Stage-2 verification prompt for the English Wikipedia source. A fresh agent re-adjudicates each Stage-1 match, now grounded in the article's real fetched lead text. Also reused verbatim for Simple English Wikipedia.

### Wikipedia Repoint Prompt (Stage 3)

You are fixing one dictionary->Wikipedia match that was judged NOT to be the same topic. Find a  
↪ better Wikipedia article for it, if one exists. Target wiki host: \${host}.

Step 1. Read item index \${i} (0-indexed) of this JSON array file: \${pipe}/repoint\_todo.json  
It has: "term", "definition" (full Oxford definition), "current\_title" (the REJECTED article), "  
↪ verdict", "note".

Step 2. Search Wikipedia for the article that describes the SAME concept as term+definition. Run  
↪ this helper (it prints JSON [{title, resolved, intro, missing}]):  
python3 \${oxdir}/wiki\_search.py --host \${host} "your search query"

Try 1-3 queries using the term and key words from the definition. You may also check specific  
↪ titles:  
python3 \${oxdir}/wiki\_search.py --host \${host} --titles "Some Title" "Another Title"

Step 3. Decide, grounded in the fetched intros:

- If some article clearly describes the SAME concept (exact concept, or the primary article for  
↪ it), choose it. Disambiguation suffixes and singular/plural are fine.
- If the best you can find is still only broader/narrower/related, or the concept has no  
↪ standalone article (e.g. it only redirects to a disambiguation page or is a mere section),  
↪ choose NONE.

Step 4. Write a JSON object to EXACTLY this path: \${pipe}/repoint\_out/\${i}.json  
Shape: {"term": <term>, "current\_title": <the rejected title>, "new\_title": <the better article  
↪ title, or null if none exists>, "verdict": "confirmed" if you found a same-concept article  
↪ else "no\_better\_target", "note": <short reason grounded in the intro you read>}

After writing, reply with one line: term -> new\_title (or "no better target").

Figure 3: Repair-loop prompt run only over English Wikipedia pairs left unconfirmed after Stage 2. Unlike the judge/verify prompts, this agent is tool-using: it actively issues search queries against `wiki_search.py` before deciding on a replacement target. No analogous stage exists for the other three sources.

### nLab Judge Prompt (Stage 1)

You are matching entries from the Oxford Dictionary of Mathematics to pages on the nLab (<https://ncatlab.org/nlab/>), a category-theory/higher-mathematics wiki.

Read this JSON file: {infile}

Each element has "term", "definition" (the Oxford definition, may be truncated with ..., and "candidates": nLab page names. Each candidate has a "source" field ("name" = matched by page name, "redirect" = the nLab redirects this term to that page) which you may use as weak context but should not treat as evidence of correctness -- redirect candidates in particular are often a related-but-different concept. Some entries have only one candidate.

For each entry pick the ONE candidate naming the same mathematical concept as term+definition, or null if none does. Judge on MEANING using the definition, not on title spelling. Rules for "same topic":

- Singular/plural, word order, hyphenation, British/American spelling, and disambiguation
- suffixes DO count as the same topic ("zero divisors" == "zero-divisor", "centre" == "center").
- The nLab is a category-theory wiki: it usually treats an elementary notion in a more general or abstract setting. A more general page IS a match when it is genuinely the nLab's primary treatment of the term ("antiderivative" == "indefinite integral", "sinh" == "hyperbolic sine", "iff" == "logical equivalence").
- But a page about a DIFFERENT member of the same concept cluster is NOT a match, even though the nLab redirects one to the other. These are common here and must be rejected: "ceiling" vs "floor", "imaginary part" vs "real part", "odd integer" vs "even number", "numerator" vs "fraction", "column operation" vs "row operation".
- A page covering the term as one of several cases may still match if the nLab has no dedicated page and the term is genuinely treated there ("maximum" vs "extremum" -- judge on whether the definition's content is actually covered).
- A page whose title merely CONTAINS the term but is about a narrower concept is NOT a match ("group" vs "group cohomology").
- Beware false friends: the same word names different concepts in different subfields. "bias" (statistics) is NOT the type-theoretic "biased definition"; "tension" (mechanics) is NOT "Nambu-Goto action"; "validation" is NOT "judgment".
- When in doubt, prefer null over a wrong match.

Write a JSON array to EXACTLY this path: {outfile}

Each element: {"term": <term>, "match": <chosen nLab page name or null>, "confidence": "high"|"medium"|"low", "reason": <short phrase>}

Keep every input term, in order. After writing the file, reply with just the count of non-null matches.

Figure 4: Stage-1 judge prompt for the nLab source. Candidate page names are tiered (exact/variant/redirect/phrase) upstream; the rules here specifically guard against the nLab's redirect-driven "concept cluster" failure mode (e.g. "ceiling" redirecting to "floor").

### nLab Verification Prompt (Stage 2)

You are verifying candidate matches between Oxford Dictionary of Mathematics entries and pages on  
→ the nLab (a category-theory / higher-mathematics wiki), using the ACTUAL nLab page text.  
Read this JSON file: {infile}  
Each element has "term", "definition" (full Oxford definition), "nlab\_page", "nlab\_intro" (the  
→ real opening prose of that page), "nlab\_stub" (true when the nLab page exists but carries no  
→ definitional prose at all -- typically just a few links), and "prior\_confidence"/"source" (  
→ both to be IGNORED -- judge fresh).  
For each element decide whether the nLab page, PER ITS INTRO, describes the SAME mathematical  
→ concept as the dictionary definition.  
Verdicts:  
- "confirmed" -- the same concept. The nLab routinely states an elementary notion in a more  
→ general or abstract setting (a dictionary's "antiderivative" against a page on indefinite  
→ integrals, or a concrete construction stated category-theoretically). That is still "confirmed  
→ " so long as the intro is genuinely about the term's concept.  
- "related" -- overlapping or adjacent but NOT the same concept: a sibling in the same cluster (a  
→ "ceiling" definition against a page about the floor function), one case of a broader page, a  
→ specialization, or a page that merely mentions the term.  
- "wrong" -- different topic, wrong sense, or a homonym collision across subfields (statistical "  
→ bias" against a type-theoretic "biased definition").  
Judge on MEANING, grounded in the intro text, not on title spelling. Singular/plural, spelling  
→ variants and word-order differences are still the same concept.  
BIOGRAPHIES: when the Oxford definition is a biographical entry (it opens with life dates, e.g.  
→ "(1802-29) Norwegian mathematician...") and the nLab page is that same person's entry, that is  
→ "confirmed". The nLab's person pages are typically just a list of the person's own writings,  
→ and that list identifies whose page it is -- it is sufficient evidence, short as it is. Use "  
→ related"/"wrong" only if the page is a different person or not a person at all.  
Otherwise, if "nlab\_stub" is true or the intro is too short to tell, the page carries no content  
→ to check against: use "related" and begin the note with "STUB:" so these stay separable from  
→ genuine concept mismatches. Do NOT infer agreement from the page name alone.  
Write a JSON array to EXACTLY this path: {outfile}  
Each element: {"term": <term>, "nlab\_page": <page>, "verdict": "confirmed"|"related"|"wrong", "  
→ note": <short reason grounded in the intro>}  
Keep all input elements, in order. After writing, reply with just a tally "confirmed=X related=Y  
→ wrong=Z".

Figure 5: Stage-2 verification prompt for the nLab source, grounded in each page's real fetched prose. Adds a biography carve-out and an explicit stub-detection rule (STUB:-prefixed notes) not present in the Wikipedia verification prompt.

### Reddit ELI5 Judge Prompt (Single Stage)

You are checking whether posts from the subreddit r/explainlikeimfive can serve as plain-language explanations of entries from the Oxford Dictionary of Mathematics.

Read this JSON file: {infile}

Each element has "term", "definition" (the Oxford definition, may be truncated with ...), and "candidates": a list of threads, each with "id", "title" (the question asked), "flair" (topic tag Reddit users assigned) and "answer" (the thread's best answer, truncated).

For each entry pick the ONE candidate thread whose answer explains the same concept as term+ definition, or null if none does.

What counts as a match:

- The thread's ANSWER must explain the same concept the definition describes. The question title is context; judge on the answer.
- The answer does not have to be rigorous or complete -- it is a lay explanation, and being informal is not a reason to reject it. It has to be ABOUT the same thing.
- A thread asking about one narrow aspect, application, or consequence of the term is NOT a match unless its answer still explains what the term is (e.g. "the abacus finger theory" explains a bead technique, not what an abacus is -> null).
- Beware word-sense collisions, which are the main failure mode here: the same word names different things in different fields, and this subreddit is general-audience. "matrix" (the film), "argument" (a disagreement), "acre" (real estate), "aggregate" (economics) are not the mathematical senses. Use the definition to disambiguate.
- A thread comparing the term with something else counts if the answer defines the term along the way (e.g. "difference between a t-test and ANOVA" does explain ANOVA). - If several threads qualify, pick the one whose answer explains the concept most directly and completely.
- The definition may start with "See SOMETHING. --"; the text after the dash is the resolved cross-reference and is the real definition.
- When in doubt prefer null. A wrong match is worse than a miss. Write a JSON array to EXACTLY this path: {outfile}

Each element: {"term": <term>, "match": <chosen candidate "id" string, or null>, "confidence": "high"|"medium"|"low", "reason": <short phrase>}

IMPORTANT: "match" must be the candidate's "id" field, not its title.

Keep every input term, in order. After writing the file, reply with just the count of non-null matches.

Figure 6: Single-stage judge prompt for r/explainlikeimfive. Unlike the wiki sources, the judge here already conditions on real content (the thread's fetched answer text, not just its title), so no separate verification stage follows; the output confidence label (high/medium/low) is emitted directly rather than a confirmed/related/wrong verdict.