How Correct Is Your Answer?
A Semantic Correctness Framework for Open QA Evaluation

Anonymous ACL submission

Abstract

Reliable evaluation of open-ended question answering remains a bottleneck for measuring the factual competence of modern LLMs. Unlike multiple-choice tasks, free-form answers may be correct in many surface forms and may fail in qualitatively different ways, including incompleteness, contradiction, overgeneration, and acceptance of false premises. Existing judgment-based and similarity-based metrics often collapse these distinctions. We address this gap with three reusable contributions. First, we introduce a fine-grained semantic correctness taxonomy that assigns Open-QA answers to eight ordered classes, separating verbose-but-correct answers from answers contaminated by hallucinated content. Second, we release CAP-Correctness, a 10k-example benchmark spanning widely used QA datasets, and CAP-Statements, an 11k-example dataset for converting QA pairs into declarative statements for NLI training and statement-based evaluation. Third, we introduce CAP, Context-Aware Precision, a reference-based metric that scores question-conditioned statements using bidirectional NLI. Under a monotonicity protocol that tests whether metrics respect the taxonomy's intended ordering, CAP outperforms established baselines. We release our data and code at `http://anonymous.for.review`

1 Introduction

Open-ended question answering (Open QA) is a long-established task requiring systems to generate free-form answers to factual questions across diverse domains (Fan et al., 2019). Despite advances in Large Language Models (LLMs), producing accurate answers remains difficult, making Open QA a rigorous benchmark for factual recall, reasoning, and answer generation. Yet the open-ended nature of these answers makes automatic correctness evaluation a persistent bottleneck. Unlike multiple-choice question answering (MCQA), where answers are restricted to fixed options, Open QA allows many distinct correct answers. These responses may differ substantially from a reference answer while still being factually correct or semantically equivalent.

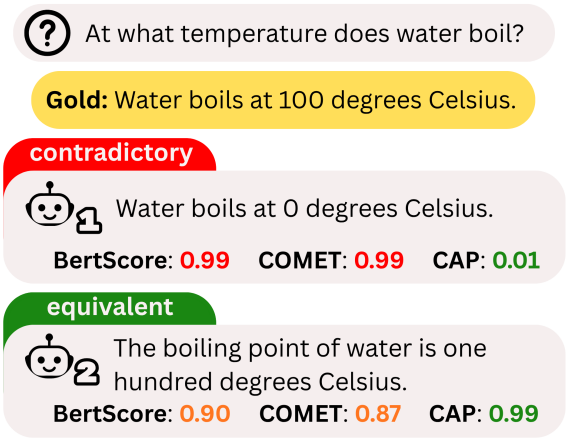

Figure 1: Comparison of QA evaluations, including our proposed CAP.

A prediction can be correct while sharing few tokens with the reference (Bulian et al., 2022), while a lexically similar prediction may omit required information, add unsupported claims, or contradict the gold answer. Fig. 1 illustrates this limitation: similarity-based metrics may reward contradictory answers while penalizing paraphrased correct ones. Evaluating Open QA, therefore, requires modeling correctness beyond textual similarity.

These shortcomings of existing protocols have also become apparent in shared-task evaluations. ClinIQLink 2025 (Colelough et al., 2025) exposes this problem in clinical QA: BLEU (Papineni et al., 2002) fails to recognize correct paraphrased answers, leading organizers to introduce a task-specific semantic scoring scheme. Similarly, AraHealthQA 2025 (Alhuzali et al., 2025) reveals a parallel issue in Arabic health QA, where open-ended answer generation is evaluated using BERTScore

(Zhang et al., 2020), despite organizers noting that such automatic metrics do not fully capture the appropriateness or trustworthiness of the proposed answers.

More broadly, even QA-specific evaluation protocols often reduce generated answers to exact-match scores or binary accept/reject judgments. This is too coarse for Open QA, where candidate answers can fail in qualitatively different ways: they may be incomplete, overinclusive, unsupported, contradictory, or based on a false premise (Kamalloo et al., 2023; Adlakha et al., 2024; Yao and Barbosa, 2024). Collapsing these cases into a single binary label obscures what kind of answer the system produced, especially for LLM outputs that mix correct content with extraneous explanations, qualifications, or unsupported details.

A natural alternative is to use LLMs themselves as judges, since they can provide more flexible semantic assessment. However, LLM-as-a-judge methods are costly to run at scale, sensitive to prompting choices, less reproducible, and affected by known biases such as position and verbosity effects (Zheng et al., 2023; Shi et al., 2024). These limitations motivate a reproducible evaluation protocol that treats the correctness of open-ended answers as a structured semantic problem.

To bridge the gap, we propose a fine-grained semantic correctness taxonomy for reference-based Open QA evaluation. Rather than reducing generated answers to binary correct/incorrect labels, the taxonomy defines eight classes that distinguish semantic relations among candidate answers, reference answers, and questions. This enables more diagnostic evaluation by identifying not only whether an answer is correct, but how it relates to the expected answer.

Building on this taxonomy, we introduce CAP, a semantic correctness framework for Open QA evaluation. Given a question, a gold answer, and a predicted answer, CAP reformulates each question–answer pair as a question-conditioned declarative statement and compares the resulting statements using bidirectional natural language inference (NLI). The taxonomy labels are mapped to an ordinal scoring scheme that aligns with human judgments of answer quality, enabling CAP to capture degrees of correctness beyond binary evaluation.

To evaluate this setting, we construct CAP-Correctness, a 10k-example semantic correctness benchmark derived from OpenBookQA, ARC, and MMLU (Mihaylov et al., 2018; Clark et al., 2018; Hendrycks et al., 2021). The benchmark covers diverse domains, question formats, and answer relations, with human verification. Since CAP applies NLI to declarative statements rather than raw question–answer pairs, we also construct CAP-Statements, an 11k-example dataset for question-conditioned QA-to-statement reformulation, and manually evaluate the reliability of the generated statements.

Using these resources, we compare CAP against lexical-overlap, embedding-based, and learned semantic metrics by testing whether their scores preserve the intended ordering of correctness categories. Across comparisons, CAP achieves stronger semantic ranking alignment, higher pairwise ordering accuracy, and fewer monotonicity violations, indicating that it better reflects the taxonomy's intended correctness ordering than similarity-based alternatives.

Our contributions are as follows:

- We propose a semantic correctness taxonomy for Open QA evaluation: an eight-class structure for fine-grained annotation of `valid`, `partial`, `overinclusive`, `invalid`, and `contradictory` model outputs, aligned with human judgement.
- We introduce CAP, a reference-based evaluation framework that scores answers via question-conditioned statement reformulation and NLI-based semantic comparison.
- We construct CAP-Correctness (10k examples), a semantic correctness benchmark with human-validated answer labels, and CAP-Statements (11k examples), a dataset for training and evaluating QA-to-statement reformulation.
- We establish a monotonicity-based evaluation protocol that tests whether metric scores respect the intended ordering of semantic correctness categories, and show that CAP outperforms traditional metrics.

2 Related Work

2.1 NLI-based Evaluation

Natural language inference has been used to evaluate generated answers beyond lexical similarity. Chen et al. (2021) formulate QA verifica-

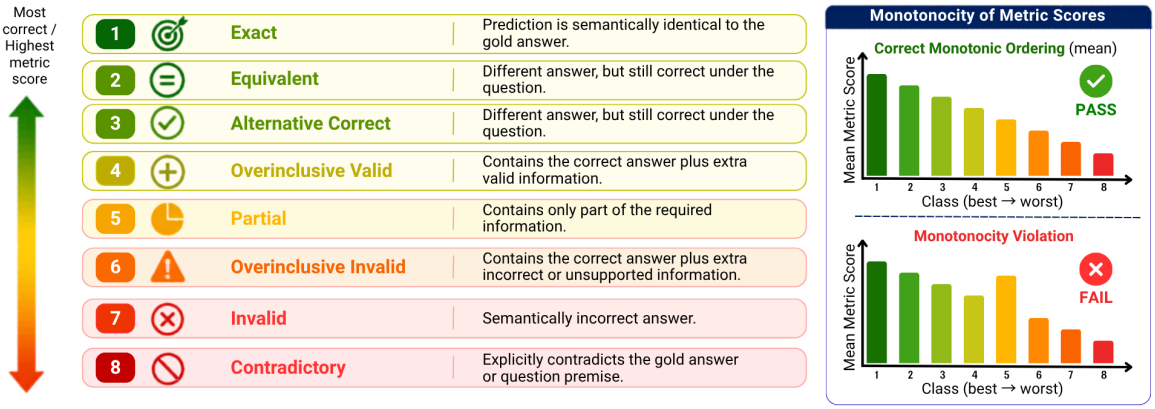

Figure 2: Correctness taxonomy for Open QA and monotonicity criterion. We define eight answer classes, ordered from most to least correct. A metric is considered higher quality if its mean score decreases monotonically across this ordering.

tion as an entailment problem over questions, evidence, and predicted answers, treating correctness as an entailed-or-not decision. Building on this, Honovich et al. (2022) benchmark NLI-based and QA-based metrics across eleven factual consistency datasets, finding large-scale NLI to be among the strongest evaluation approaches. Laban et al. (2022) further show that decomposing documents into sentence-level units before applying NLI yields more reliable inconsistency detection, motivating fine-grained statement-level evaluation. Zha et al. (2023) unify NLI, QA, and fact-verification signals into a single alignment function that achieves GPT-4-level factual consistency scoring. Chen and Eger (2023) demonstrate that NLI-based metrics are more robust than embedding-similarity metrics under meaning-changing perturbations, providing evidence that directional inference captures semantic content more faithfully than symmetric similarity.

These approaches establish NLI as a principled basis for semantic evaluation, but they largely produce binary or undifferentiated outputs: a prediction either entails the reference or does not. CAP instead scores statements bidirectionally with a weighted neutral term, yielding a continuous score that separates equivalence, incompleteness, and overgeneration within a single forward pass. We further introduce a monotonicity benchmark that tests whether a metric's scores respect the ordering induced by our correctness taxonomy, enabling systematic comparison between CAP and prior metrics.

2.2 Correctness Taxonomies for Open QA

Early Open QA evaluation relied on exact match or lexical overlap, which systematically misclassifies paraphrased correct answers as wrong and rewards surface-similar incorrect ones (Papineni et al., 2002; Lin, 2004). Kamalloo et al. (2023) show that these metrics underestimate true QA accuracy by over 50% on modern LLM outputs and that existing learned evaluators also fail on free-form answers. Xu et al. (2023) argue that single-score evaluation of long-form answers is inadequate: metrics must target separable dimensions such as factuality, completeness, and coherence rather than collapsing them into one signal. Adlakha et al. (2024) extend this critique to instruction-following models, showing that correctness and faithfulness are empirically distinct dimensions that standard binary evaluation conflates.

More recent work has moved toward graded correctness and answer equivalence. Bulian et al. (2022) propose entailment-based answer equivalence criteria that accept any answer containing at least all required content without misleading additions. Yona et al. (2024) demonstrate that standard evaluation systematically penalizes overspecific correct answers, and introduce multi-granularity evaluation that separates overspecific and underspecific correctness along distinct axes. Yao and Barbosa (2024) organize Open-QA answers into an NLI-based entailment hierarchy and assign partial or bonus credit to answers that are more general or more specific than the gold answer.

For evaluating verbose LLM-generated answers, however, these schemes share three limitations.

First, they are developed and evaluated primarily on short factoid QA datasets, leaving their transfer to multi-facet or multi-passage answers uncertain. Second, none explicitly separates a correct answer accompanied by valid elaboration from a correct answer contaminated by hallucinated content–a distinction central to evaluating modern LLM outputs. Min et al. (2023) address the related problem of factual precision in long-form text by decomposing outputs into atomic facts and verifying each independently, but this approach operates at subsentence granularity and does not yield an answer-level correctness class. Third, existing partial-credit mechanisms either rely on LLM-generated intermediate reasoning to estimate inference difficulty (Yao and Barbosa, 2024), or on token-level overlap, tying the score to surface form rather than to direct semantic relations between expected and predicted answers.

CAP addresses these gaps directly. We extend the correctness taxonomy to eight classes by adding `overinclusive-valid` and `overinclusive-invalid` categories that capture the verbose behavior characteristic of modern LLM outputs, making the evaluation diagnostic: a correct but verbose answer should not be penalized the same way as a correct answer contaminated by hallucinated content. Rather than eliciting post-hoc reasoning for partial credit, CAP derives a continuous score from bidirectional NLI probabilities over question-conditioned declarative statements, tying the score directly to semantic compatibility and completeness.

3 Proposed Taxonomy for OpenQA

We define semantic correctness as the relation between a predicted answer and a gold answer under the same question context. To evaluate an open-ended answer, exact matching is only the simplest case: a prediction may repeat the gold answer verbatim, giving an `exact` answer, or express the same meaning with different wording, giving an `equivalent` answer. However, Open QA often allows answers that are correct and differ greatly from the reference. For example, for a question such as *Name three cities in Europe*, many completely different sets of cities may still be correct; we treat these as `alternative-correct` answers Fig. 2.

Model responses can fail or deviate from the target answer in distinct ways. A `partial` answer provides only some of the required information, such as naming two cities when three are requested. An `overinclusive-valid` answer includes extra information beyond the request, but the added content is accurate and relevant—for example, briefly describing the cities' locations. If the response contains the correct answer but also introduces an unsupported or false claim, such as listing a non-European city, it becomes `overinclusive-invalid`. Other predictions may directly contradict the gold answer or the question premise, giving a `contradictory` answer, while remaining mistakes that do not fit these cases are labeled `invalid` (Fig. 2). These distinctions also define an expected ordering of answer quality, shown in Eq. (1). Fully correct answers should receive higher semantic correctness scores than incomplete answers, and incomplete answers should generally score above answers that introduce false information or contradict the question. We formalize this expectation as a **monotonicity property**, which is used throughout our evaluation.

$$\begin{aligned}\texttt{exact} \geq \texttt{equivalent} \approx \texttt{alternative-correct} \\ > \texttt{overinclusive\_valid} > \texttt{partial} \\ > \texttt{overinclusive\_invalid} \\ > \texttt{invalid} \geq \texttt{contradictory}.\end{aligned} \tag{1}$$

Some neighboring distinctions, especially between `partial` and `overinclusive-valid` answers, may be context-dependent; nevertheless, the ordering captures the dominant preference expected in reference-based QA evaluation.

4 CAP-QA Framework

An NLI model maps an ordered statement pair $(\mathrm{s}_a, \mathrm{s}_b)$ to a distribution over three labels–*entailment*, *neutral*, and *contradiction*. We write $\mathrm{s}_a \rightarrow \mathrm{s}_b$ for the directed inference from premise s_a to hypothesis s_b, and obtain these probabilities from a pretrained NLI classifier (see App. C.2).

CAP Design. Given a question q, a gold answer g, and a predicted answer p, we first generate corresponding declarative statements $\mathrm{s}(\mathsf{q}, \mathsf{g})$ and $\mathrm{s}(\mathsf{q}, \mathsf{p})$. CAP then measures the semantic relationship between the generated statements using bidirectional entailment scoring. Let $\mathrm{s}_\mathsf{g} := \mathrm{s}(\mathsf{q}, \mathsf{g})$ and $\mathrm{s}_\mathsf{p} := \mathrm{s}(\mathsf{q}, \mathsf{p})$. For these statements, we define a

directional score:

$$\mathbf{D}(\mathbf{s}_g \to \mathbf{s}_p) := \quad (2)$$
$$P_{\text{entailment}}(\mathbf{s}_g \to \mathbf{s}_p) + \lambda P_{\text{neutral}}(\mathbf{s}_g \to \mathbf{s}_p),$$

where $\lambda \in [0, 1]$ controls the contribution of the neutral class. CAP is then defined as:

$$\mathbf{CAP}(\mathbf{s}_g, \mathbf{s}_p) := \quad (3)$$
$$\alpha \mathbf{D}(\mathbf{s}_g \to \mathbf{s}_p) + (1 - \alpha)\mathbf{D}(\mathbf{s}_p \to \mathbf{s}_g),$$

where $\alpha \in [0, 1]$ balances semantic compatibility and semantic completeness.

The bidirectional formulation allows CAP to distinguish between semantically equivalent and semantically incomplete answers by incorporating entailment in both directions between the gold and predicted statements. In our experiments, we use $\alpha = 0.85$ and $\lambda = 0.30$. These values were selected via a sweep on a held-out subset of CAP-Correctness; moderate neutral weighting combined with asymmetric scoring jointly maximizes semantic ranking. The full sweep is reported in App. C.3.

Statement Generation. To reformulate question–answer pairs into declarative statements for NLI evaluation, we fine-tune an mT5-based seq-to-seq model (Xue et al., 2021) on our statement generation dataset described in §5. Fine-tuning details are provided in §5 and App. C. On the held-out test set, the model achieves strong surface-form agreement with the references: 96.64 BLEU, 98.08 ROUGE-L, and 76.7% Exact Match. Manual inspection confirms that most non-exact outputs preserve the intended meaning, making the generated statements suitable for downstream NLI evaluation.

Boundedness. CAP is a convex combination of two directional scores $\mathbf{D} \in [0, 1]$ and therefore yields a continuous score $\mathbf{CAP}(\mathbf{s}_a, \mathbf{s}_b) \in [0, 1]$. Full derivation is given in App. A.

5 Dataset Construction

We instantiate our evaluation framework with two complementary datasets: **(1) CAP-Correctness**, a semantic evaluation benchmark annotated using our proposed correctness label set, and **(2) CAP-Statements**, a dataset for reformulating question–answer pairs into declarative statements for NLI-based evaluation. Details on dataset acquisition, annotation, and statistics are provided in App. B.

Dataset	Train	Val.	Test
CAP-Correctness	-	-	10346
CAP-Statements	8800	1100	1100

Table 1: Dataset statistics.

5.1 Semantic Correctness Benchmark

CAP-Correctness contains 10,346 *[question, gold answer, candidate answer, correctness label]* examples from OpenBookQA, AI2 ARC, and MMLU, spanning elementary through undergraduate-level questions. Candidate answers and labels are produced by an LLM-assisted pipeline, with a 573-example subset (5.5%) re-labeled by human annotators against the same taxonomy. Human–LLM agreement is substantial (Tab. 7: Cohen's $\kappa = 0.716$, quadratic-weighted $\kappa = 0.714$), confirming that the synthetic labels reliably track human judgements of semantic correctness. Full annotation protocol, along with statistics is provided in App. B.3.

CAP-Statements is a collection of 11,000 *[question, answer, statement]* triples, where each statement preserves the semantic content of its corresponding question–answer pair. The dataset spans four question types (Tab. 8); among these, long-form items with multi-sentence context are the hardest reformulation regime, since several sentences must be compressed into a single declarative claim. This makes CAP-Statements a non-trivial benchmark for statement generation as well as a natural supervision source for QA-to-statement reformulation (Demszky et al., 2018). Details on construction and human validation are provided in App. B.5.

6 Experiment Design

We evaluate whether CAP and existing automatic metrics preserve the semantic-correctness ordering induced by our taxonomy (Eq. (1)). The comparison covers widely used metrics from three families: **Lexical overlap:** BLEU, ROUGE-L, METEOR (Banerjee and Lavie, 2005). **Contextual embedding:** BERTScore (F1). **Learned semantic regression:** COMET (Rei et al., 2020).

For each metric we report the exact model checkpoint and version used in App. C.

6.1 Research Questions

Our experiments are designed around three research questions:

- **RQ1 (Monotonicity)** For every class pair (c_i, c_j) with $c_i \succ c_j$ in our taxonomy, does a mean metric score on c_i exceed the mean on c_j?
- **RQ2 (Local separability)** Do metrics distinguish *neighboring* taxonomy classes that are especially challenging for surface- and embedding-based scorers?
- **RQ3 (Generalization to LLM outputs)** Does CAP preserve these properties on free-form answers generated by state-of-the-art LLMs, rather than only on the semi-synthetic answer variants in CAP-Correctness?

6.2 Evaluation Measures

For every gold–prediction–label triple $(\mathsf{g}, \mathsf{p}, \mathsf{c}) \in$ CAP-Correctness, a metric $\mathbf{m}$ produces a score $\mathbf{m}(\mathsf{g}, \mathsf{p}) \in [0, 1]$. We evaluate whether these scores respect the semantic correctness ordering in Eq. (1) using four complementary measures. **Rank correlation** measures global agreement between metric scores and taxonomy ranks using Spearman's ρ and Kendall's τ. **Pairwise ranking accuracy** reports the fraction of class-ordered answer pairs for which the metric assigns a higher score to the more correct answer (random baseline 0.5). **Monotonicity violations** count class pairs whose mean metric scores invert the expected taxonomy order. **Hard neighboring-pair accuracy** restricts pairwise accuracy to adjacent, locally difficult class contrasts in Eq. (1). Full definitions are provided in App. C.1.

6.3 Implementation Details

CAP scores are computed with `cross-encoder/nli-deberta-v3-large` over declarative statements produced by an mT5-base generator fine-tuned on the CAP-Statements training split. CAP-Correctness is used exclusively as a held-out evaluation benchmark for the metric comparisons in §7.1–§7.3; §7.4 additionally evaluates CAP on an independent set of LLM-generated answers to the same questions. Full checkpoint pins, baseline versions, and bootstrap details are in App. C.2.

7 Results

7.1 CAP against Established Metrics

Tab. 2 reports the headline comparison. CAP operates in a markedly different regime from the baselines. Lexical metrics and BERTScore remain close to the 0.5 pairwise-accuracy baseline and show weak rank correlations, suggesting that surface-form and embedding similarity provide little signal for our taxonomy's ordering. COMET performs better than this near-random band, but still recovers only part of the ordering, consistent with its tendency to merge answer relations that our taxonomy distinguishes. CAP achieves the strongest rank correlation by a wide margin and raises pairwise accuracy into a clearly informative range.

Metric	Spearman ρ	Kendall τ	Pairwise Acc.
BLEU	20.24	14.45	55.06
ROUGE-L	26.31	19.00	56.36
METEOR	23.94	17.06	56.81
BERTScore	24.13	16.77	59.09
COMET	33.71	24.46	63.33
CAP	**60.69**	**46.94**	**75.58**

Table 2: Correlation and pairwise ranking accuracy between metric scores and semantic correctness ordering.

The per-pair separability profile in Tab. 3 places this gain on a difficulty axis: CAP is near-perfect on distant class pairs and degrades smoothly as the pair becomes more local.

Comparison	CAP AUC
Exact > Invalid	98.47
Partial > Invalid	95.24
Equivalent > Invalid	92.25
Equivalent > OV	79.47
Partial > OI	90.60
OV > OI	69.08

Table 3: Pairwise semantic separability of CAP across semantic correctness categories.

The next two subsections decompose this picture at the class-mean level (§7.2) and at the per-example level on the hardest neighbors (§7.3).

7.2 Monotonicity Analysis

At the class-mean level, CAP also preserves the taxonomy's *ordering*, not just its global rank correlation. The per-pair separability profile in Tab. 3 shows that CAP is near-perfect on distant class pairs and degrades smoothly toward the local ones, and Fig. 3 makes the geometry visible: CAP's per-class distributions form largely distinct bands on the score axis, while COMET's collapse into a narrow overlapping region. Across the 27 strictly ordered class pairs, lexical metrics and BERTScore invert roughly half, COMET inverts $9/27$, and

CAP reduces this to $4/27$, with the remaining inversions concentrated on `alternative-correct` and the `partial` / `overinclusive-valid` pair. Full counts, per-class means, and mechanism analysis are in App. D.1.

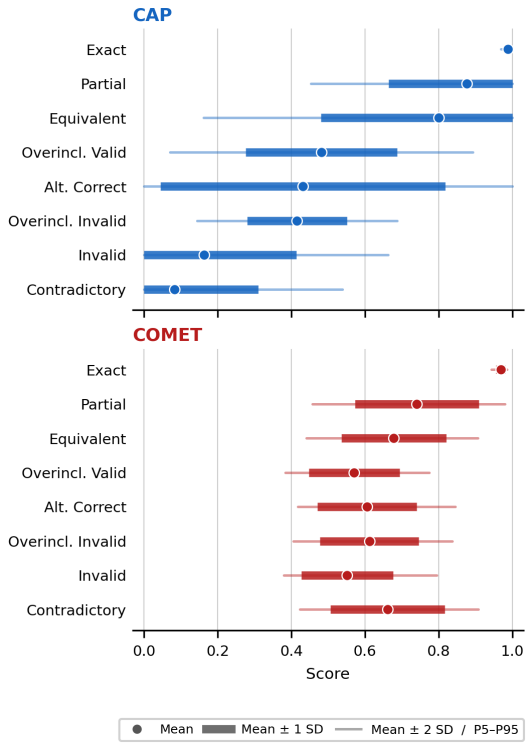

Figure 3: Label semantic distribution.

7.3 Hard Neighboring-Pair Evaluation

While monotonicity captures the ordering at the class-mean level, pairwise accuracy on neighboring class pairs (Tab. 4) tests whether the ordering holds at the per-example level on the hardest contrasts. Most baselines collapse to or below chance here, confirming that their above-baseline global behavior in Tab. 2 is carried by easy pairs at the extremes of the ordering—`equivalent` vs. `invalid`, `exact` vs. `contradictory`—and not by the genuinely hard distinctions in the middle of the taxonomy.

The `overinclusive-valid` vs. `partial` comparison is the most informative failure case. All metrics except METEOR fall below chance on this pair, with CAP showing the largest reversal. This is a direct consequence of CAP's asymmetric bidirectional formulation (§4): partial answers retain high $g \to p$ entailment along the heavily-weighted direction ($\alpha = 0.85$), while overinclusive-valid answers reverse this asymmetry. Tab. 17 and Fig. 3

Metric	Eq > Part	OV > Part	OV > OI	Alt > Inv
BLEU	32.45	45.95	34.16	38.41
ROUGE-L	19.40	21.81	27.90	37.27
METEOR	37.26	**63.71**	35.91	41.59
BERTScore	42.45	26.59	31.12	57.59
COMET	38.85	21.96	41.40	61.61
CAP	**57.14**	14.30	**69.08**	**72.54**

Table 4: Pairwise ranking accuracy on hard neighboring semantic distinctions. OV = overinclusive-valid, OI = overinclusive-invalid, Alt = alternative-correct, Inv = invalid.

confirm the resulting inversion across both label sources. This pair is excluded from monotonicity scoring because the taxonomy treats it as ambiguous, but it remains the clearest diagnostic of CAP's design tradeoff: the same asymmetry that helps CAP elsewhere makes it near-inverted on this axis.

Per-unit entailment over atomic semantic decompositions could expose this asymmetry more directly than whole-statement NLI; we discuss this further in §9.

7.4 Evaluation against LLM outputs

To assess external validity, we test whether our eight-class taxonomy captures how current LLMs answer open-ended questions and whether CAP's class-mean ordering holds for model-generated answers. We collect zero-shot responses from GPT-4o, Gemini 2.0 Flash, and Qwen3-8B-Instruct on a random sample of 1,000 CAP-Correctness questions. Human annotators then label each response according to the taxonomy (see App. B.1).

Model	Exact	Partial	OI	Contradictory
GPT-4o	93.18	35.64	-	19.89
Gemini Flash	87.30	50.22	37.88	23.40
Qwen 3	91.18	39.95	33.91	10.40

Table 5: Mean CAP score on human-labeled LLM-generated answers, grouped by assigned correctness class. "–" indicates the model produced no answers in that class.

The resulting annotations support both claims. First, LLM responses populate nearly all eight taxonomy classes: the only missing model–class combination is GPT-4o in the `overinclusive-invalid` class. This suggests that the proposed categories capture real model behavior, not only CAP-Correctness reference-answer structure. Second, for each of the three models, the mean CAP score by class follows the

expected ordering among the relevant classes , matching the pattern observed on CAP-Correctness. Thus, the monotonicity result from §7.2 extends to model-generated answers.

8 Discussion

8.1 Alignment with Human Judgement

The empirical claim we make for CAP is that its scores follow the semantic correctness ordering induced by our taxonomy. Because both the ordering and the class labels of CAP-Correctness were produced by humans, this correspondence is, indirectly, a correspondence with human judgement: the taxonomy was designed on the basis of how candidate answers can differ from a reference, and the per-example labels were validated by human annotators against that same taxonomy (App. B). The monotonicity, pairwise accuracy, and ranking measures of §6 therefore assess alignment with a human-defined target.

The "ground truth" against which we benchmark metrics is itself constructed and inherits both the design choices of our taxonomy and the subjectivity of human annotation. We view this not as a flaw but as the unavoidable structure of semantic evaluation: there is no taxonomy-free notion of how correct an open-ended answer is. What the framework provides is an explicit, inspectable target ordering, against which any candidate metric—CAP, BLEU, COMET, or future learned alternatives—can be benchmarked on equal footing. The goal of the present work is correspondingly two-fold: (i) to show that CAP follows this human-defined ordering, and (ii) to show that it does so more reliably than existing metrics. The ablation in Tab. 13 shows that neither bidirectionality nor the neutral-class weighting can be removed without degrading ρ by 3–22 points, and the error analysis (App. E) confirms that the alignment holds under human labels and breaks down only on `alternative-correct` and the `partial`/`overinclusive-valid` pair.

8.2 CAP as a Standalone Classifier

A natural next step is to use CAP not only as a scorer but also as the labeler. Because CAP produces a continuous score in $[0, 1]$ that empirically separates the taxonomy classes (Fig. 3), one can define class boundaries directly on the score axis–for example, $[0, 0.125)$ for contradictory, $[0.125, 0.25)$ for invalid, and so on up to the equivalent regime near 1. Calibrating these thresholds on CAP-Correctness would yield a label-free evaluator that, given a question, a gold answer, and a prediction, returns both a continuous score and a taxonomy class.

This protocol also generalizes beyond CAP. A natural evaluation recipe for any future semantic correctness metric would be: (i) re-use CAP-Correctness as the benchmark, (ii) re-use the taxonomy ordering as the monotonicity target, and (iii) calibrate the metric's own thresholds on the same labeled data. Under this protocol, CAP-Correctness becomes a shared substrate for comparing semantic correctness metrics, independent of CAP itself.

9 Conclusion and Future Work

We introduce a correctness taxonomy for Open QA evaluation and CAP, an NLI-based scorer for it. CAP roughly doubles the rank correlation of the strongest existing baseline against the taxonomy ordering.

The framework's value extends beyond the score itself. As a continuous scorer, CAP can replace BLEU, ROUGE, or BERTScore in Open QA pipelines. Its class-mean geometry also serves as a diagnostic of LLM answer style—the proportion of a model's outputs that are `partial`, overinclusive, or `contradictory` is information that binary correctness collapses. With threshold calibration, CAP functions as a label-free classifier returning both a score and a taxonomy class. The released datasets are reusable in their own right: CAP-Correctness as a labeled corpus future metrics can train against, and CAP-Statements as a QA-to-statement reformulation resource.

Several directions remain open for future work. Splitting answers into smaller subcomponents and scoring entailment over each could resolve the bidirectional-NLI artefact on the `overinclusive-valid` / `partial` axis; stronger NLI backbones with broader world knowledge could close the `alternative-correct` ceiling; and multilingual NLI checkpoints could port the framework off English. The taxonomy itself is not fixed either: as LLMs evolve and Open QA expands into new domains, the categories that meaningfully partition model behavior will shift. CAP-Correctness and the monotonicity protocol support adding, splitting, or merging classes accordingly, rather than treating the current eight as final.

Limitations

CAP inherits the limitations of reference-based, whole-statement NLI evaluation. It can structurally invert the `partial/overinclusive-valid` distinction, since partial answers are often entailed by the gold answer, while verbose valid answers often entail the gold but are not entailed by it. It can also under-score `alternative-correct` answers when a valid prediction satisfies the question without entailing the single reference answer, making CAP dependent on the NLI model's world knowledge.

Our benchmark is currently limited to English educational QA datasets derived from multiple-choice sources, with synthetic candidate answers and labels generated by a closed LLM-assisted pipeline. Although we validate a subset with human annotators, only a small portion of CAP-Correctness is human-labeled and examples are singly annotated, so we do not estimate inter-annotator agreement. The statement-generation step is another bottleneck, especially for long-context inputs, where reformulation errors can propagate directly into CAP scores.

Finally, CAP is also more computationally expensive than lightweight metrics, since each score requires statement generation and two NLI passes. The taxonomy assigns one label per answer, while real outputs may combine several correctness dimensions, such as partial correctness and unsupported extra information.

Ethics and Broader Impact

Copyright and Licensing. The correctness and statements datasets are derived from publicly available educational benchmarks–OpenBookQA, AI2 ARC, and MMLU–that permit non-commercial research use. The derivatives retain only the question–answer content and annotations required for semantic evaluation and are intended strictly for non-commercial research.

Ethics and Data Privacy. The source material consists of general-knowledge and academic questions and contains no personal, sensitive, or personally identifiable information. No student records, user identities, or private data are present, and the annotations are limited to educational QA content, posing no privacy risk.

Human Annotation. Validation labels were collected from human annotators under the conditions described in App. B.1.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2024. Evaluating correctness and faithfulness of instruction-following models for question answering. *Transactions of the Association for Computational Linguistics*, 12:681–699.

Hassan Alhuzali, Walid Al-Eisawi, Muhammad Abdul-Mageed, Chaimae Abouzahir, Mouath Abu-Daoud, Ashwag Alasmari, Renad Al-Monef, Ali Alqahtani, Lama Ayash, Leen Kharouf, Farah E. Shamout, and Nizar Habash. 2025. AraHealthQA 2025: The first shared task on arabic health question answering. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 107–118, Suzhou, China. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanran Chen and Steffen Eger. 2023. MENLI: Robust evaluation metrics from natural language inference. *Transactions of the Association for Computational Linguistics*, 11:804–825.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Brandon Colelough, Davis Bartels, and Dina Demner-Fushman. 2025. Overview of the ClinIQLink 2025 shared task on medical question-answering. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 378–387, Viena, Austria. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *Preprint*, arXiv:1809.02922.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic study of position bias in LLM-as-a-judge. *ArXiv preprint*, abs/2406.07791.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Peiran Yao and Denilson Barbosa. 2024. Accurate and nuanced open-QA evaluation through textual entailment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2575–2587, Bangkok, Thailand. Association for Computational Linguistics.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Narrowing the knowledge evaluation gap: Open-domain question answering with multi-granularity answers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6737–6751, Bangkok, Thailand. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Boundedness of CAP

We show that $\mathbf{CAP}(\mathbf{s}_a, \mathbf{s}_b) \in [0, 1]$ for any pair of statements $(\mathbf{s}_a, \mathbf{s}_b)$.

Since NLI probabilities sum to one for any ordered statement pair,

$$P_{\text{entailment}}(\mathbf{s}_a \to \mathbf{s}_b) + P_{\text{neutral}}(\mathbf{s}_a \to \mathbf{s}_b) + P_{\text{contradiction}}(\mathbf{s}_a \to \mathbf{s}_b) = 1. \quad (4)$$

and each probability lies in $[0, 1]$. Subtracting the contradiction term, which itself lies in $[0, 1]$,

$$0 \le P_{\text{contradiction}}(\mathbf{s}_a \to \mathbf{s}_b) \le 1. \quad (5)$$

as $\lambda \in [0, 1]$, this yields

$$0 \le P_{\text{entailment}}(\mathbf{s}_a \to \mathbf{s}_b) + \lambda P_{\text{neutral}}(\mathbf{s}_a \to \mathbf{s}_b) = \mathbf{D}(\mathbf{s}_a \to \mathbf{s}_b) \le 1. \quad (6)$$

Because CAP is a convex combination of two directional scores $\mathbf{D}(\mathbf{s}_a \to \mathbf{s}_b), \mathbf{D}(\mathbf{s}_b \to \mathbf{s}_a) \in [0, 1]$ with weight $\alpha \in [0, 1]$,

$$0 \le \mathbf{CAP}(\mathbf{s}_a, \mathbf{s}_b) \le 1. \quad (7)$$

CAP attains its maximum when both directional entailment scores approach 1, corresponding to semantically equivalent statements, and approaches 0 when the NLI model assigns high contradiction probability in both directions.

B Datasets details

B.1 Annotation

Human annotation is used at three points in this work: validating a 573-example subset of CAP-Correctness against the eight-class taxonomy (App. B.3), validating 1,353 generated declarative statements from CAP-Statements (App. B.5), and labeling the LLM-generated answers used in §7.4. All three annotation tasks were performed by the same two annotators.

Annotator profile. Both annotators hold bachelor's degrees and certified C1-level English proficiency, and are therefore qualified to judge the educational-level QA content used in our benchmarks.

Compensation and consent. Annotators were compensated at 3 times the average pay according to their demographic region. Prior to annotation, they were informed of the purpose of the task, the intended research use of their labels, and the public release of the resulting datasets and labels, and provided consent on these terms. The annotated content consists exclusively of educational questions and candidate answers and contains no sensitive, offensive, or distressing material.

B.2 Statement Generation Dataset

Instructions. Annotators received written guidelines containing the relevant taxonomy (Tab. 6 for the correctness tasks, Tab. 10 for statement validation), one worked example per class, and a short calibration batch before live annotation began.

Assignment. Each example is assigned a single label by one annotator. We do not double-annotate and therefore do not separately estimate inter-annotator agreement; the agreement figures reported in Tab. 7 measure human–LLM agreement, which is the quantity of interest for validating the synthetic labeling pipeline.

Effort. Across the three tasks, each annotator spent approximately 12 hours on annotation.

B.3 CAP-Correctness

Generation Pipeline For each *[question, gold_answer]* pair from the source corpora, the generation pipeline produces a single candidate answer. Generation is controlled to yield an approximately balanced distribution over the semantic labels describing the relationship between the gold answer and the predicted answer. Each label is instantiated with a separate prompt, conditioned on both the class definition in Tab. 6 and the original gold answer. As a result, cases such as `overinclusive-valid` and `overinclusive-invalid` are generated through explicit, class-specific instructions rather than left to model discretion. We use the Claude Haiku 4.5 API. The prompt template and per-class instructions are given in App. B.4. This class-conditioned setup explains the near-uniform per-class counts in Fig. 4, in contrast to the long-tailed distribution expected from sampling natural model outputs.

Distribution. Tab. 6 lists the full set of correctness labels with their definitions, and Fig. 4 reports the resulting per-class counts on the 10,346 examples. The distribution is approximately balanced across all eight categories, with no class accounting for more than ∼13% of the corpus, so downstream metric comparisons are not dominated by any single class. Acquisition per dataset is reported in

Fig. 5.

Label	Definition
exact	Semantically identical to the gold answer.
equivalent	Same meaning expressed through linguistic variation or paraphrasing.
alternative-correct	Different but still semantically correct answer.
partial	Contains only part of the required information.
overinclusive-valid	Correct answer with additional valid information.
overinclusive-invalid	Correct answer with additional incorrect information.
invalid	Semantically incorrect answer.
contradictory	Explicitly contradicts the gold answer or question premise.

Table 6: Semantic correctness labels used in the CAP evaluation benchmark.

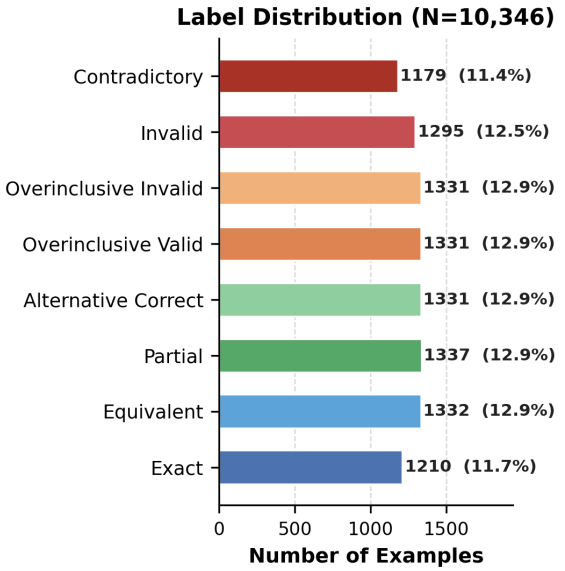

Figure 4: Distribution of semantic correctness labels in CAP-Correctness.

Validation Protocol The 573-example human validation uses the same eight-class taxonomy as the LLM-assisted pipeline (Tab. 6); annotation conditions and protocol are described in App. B.1. Because each example receives a single human label, the figures in Tab. 7 quantify human–LLM agreement rather than inter-annotator agreement, which is appropriate for validating the synthetic labeling pipeline.

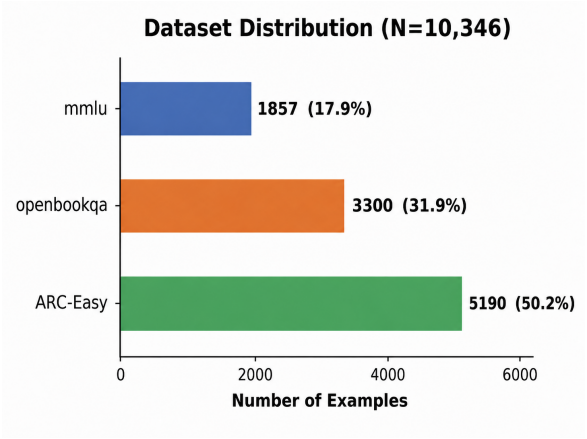

Figure 5: Questions Acquisition Statistics

Interpretation. The values reported in Tab. 7 fall in the *substantial agreement* range under the Landis–Koch convention. The similarity between unweighted (Tab. 7 $\kappa = 0.716$) and quadratic-weighted ($\kappa = 0.714$) agreement further suggests that weighting disagreements by their distance in the taxonomy does not materially change the agreement estimate. In other words, the observed disagreements are not simply near-misses between neighboring categories. We take this as evidence that the synthetic labels broadly align with human semantic judgments under the proposed taxonomy. However, this agreement does not by itself validate the taxonomy as a model of correctness; that is a separate question, which we return to in §8.

Agreement Metric	Score
Cohen's κ	71.57
Linear weighted κ	70.29
Quadratic weighted κ	71.36

Table 7: Human–LLM agreement on a 573-example validation subset of the semantic correctness benchmark.

B.4 Candidate-Answer Generation Prompt

Design. The candidate-answer generator uses a single prompt template instantiated once per target class. Each call receives the question, the gold answer, the target class name, the class definition from Tab. 6, and one class-specific instruction from Tab. 18. Two design choices follow from the structure of the taxonomy. First, conditioning generation on the gold answer (rather than asking the model to produce its own gold) is what allows the overinclusive classes to be split cleanly into the valid- and invalid-extras variants: the model is told both what to preserve and what kind of extra

content to add. Second, the class-specific instruction is the only component that varies across the eight calls per item, which keeps the input/output schema and the surface-form constraints constant across classes and avoids confounding the class signal with formatting drift.

Prompt template. The template below is sent for every $(\mathsf{q}, \mathsf{g}, \mathsf{c})$ triple; bracketed placeholders are filled per call.

```
SYSTEM:
f"Label: label" f"Definition:
label_descriptions[label]" "For each
item below, generate one answer
matching the label definition above."
"Return ONLY a valid JSON array of
strings, one answer per item, in the
same order." "No explanation, no extra
text — just the JSON array."
f"Items:json.dumps(items, indent=2)"
"JSON array:"
```

B.5 CAP-Statements

B.5.1 Generation Pipeline

Source-free synthesis. Unlike CAP-Correctness, which augments existing benchmark questions with candidate answers, CAP-Statements is generated from scratch. We prompt Claude Sonnet 4.6 to produce *[question, answer, statement]* triples directly, conditioning each call on (i) a target academic subject from Fig. 7 (general knowledge plus the seven academic subjects) and (ii) a target question type from Tab. 8. This two-axis conditioning is what yields the balanced subject and question-type distributions reported in Figs. 6 and 7; free-form sampling would otherwise collapse onto short factoid questions in dominant subjects.

Statement as supervision target. The generator is instructed to produce the declarative statement *jointly* with the question and answer, rather than reformulating an existing question–answer pair post hoc. This guarantees that every training example contains a statement that faithfully encodes the same proposition expressed by the question–answer pair, removing one source of label noise from the supervision used to fine-tune the mT5-base reformulator.

Sampling and deduplication. We sample with temperature 1.0 and top_p $= 0.95$ to encourage lexical diversity, issuing one API call per *(subject, question-type)* cell and generating in batches until the target cell count is reached. Exact duplicates and near-duplicates (normalized-text Jaccard ≥ 0.9 on the question field) are removed. The resulting 11,000 triples are split into 8,800 / 1,100 / 1,100 for train / validation / test (Tab. 1).

Prompt template. The template below is sent for every *(subject, question-type)* cell; bracketed placeholders are filled per call.

```
SYSTEM:
You generate training examples for a
question-to-statement reformulation
model. Each example is a JSON object
with exactly three fields: question,
answer, and statement.

The statement must be a single
declarative sentence that expresses the
same proposition as the question-answer
pair, with no question marks, no
second-person address, and no
meta-commentary (e.g., do not write
"the answer is. . . ").

Output ONLY the JSON object, with no
preamble or surrounding text.

USER:
Subject: {subject}
Question type: {question_type}
Question-type definition:
{question_type_definition}

Generate one (question, answer,
statement) triple in the specified
subject and following the specified
question type. Constraints:
- The question must match the surface
form of the specified type (see
definition above).
- The answer should be a phrase or short
sentence; do not explain or justify it.
- The statement must be a single
grammatical declarative sentence and
must contain all the semantic content
of the question-answer pair, with no
extra information.
- For the blank-spaces-task type, the
statement must fill in the blank
explicitly (no underscores remain).
- For the long-question type, the
statement must compress all contextual
sentences and the question into one
declarative clause.
- Do not produce duplicates of
well-known textbook examples (e.g.,
"What gas do plants release during
photosynthesis?" already exists in the
dataset).
```

The {question_type_definition} placeholder is filled from Tab. 8; the subject placeholder is drawn uniformly from {*general knowledge*, Biology, Physics, History, Geography, Chemistry, Mathematics, Computer Science} with a target proportion of 56.2% general knowledge and $\approx 6.3\%$ each for the academic subjects, matching the distribution in Fig. 7.

The four question types are listed with definitions in Tab. 8 and with concrete instances in Tab. 9. The `short-question` type covers direct factoid questions, `sentence-to-complete` covers completion-style prompts, `long-question` covers items where the question is preceded by one or more sentences of context, and `blank-spaces-task` covers cloze-style items with explicit blanks. Long-question items are the most demanding regime for reformulation, since multiple sentences of context must be compressed into a single declarative claim while preserving the question–answer relation.

Label	Definition
short-question	Direct factual question.
sentence-to-complete	Sentence completion or prompt-style question.
long-question	Question preceded by contextual description or introductory sentences.
blank-spaces-task	Question or statement containing blank spaces that must be filled.

Table 8: Label question type used in CAP-Statements

Label	Example
short-question	What gas do plants release during photosynthesis?
sentence-to-complete	The sun is responsible for...
long-question	I have a shirt that is now too small, what can I do to conserve and reuse the fabric?
blank-spaces-task	Light bends when it passes from air into _____ at an angle.

Table 9: Examples for each question type used in CAP-Statements.

Distribution Fig. 6 reports the question-type distribution across the 11,000 triples. The four types are approximately balanced–no type accounts for more than 27.4% of the corpus—ensuring that the statement generator is exposed to all reformulation regimes during training rather than being biased toward the easier short-question setting.

Subject distribution CAP-Statements skews heavily toward general-knowledge questions (56.2%), with the seven remaining academic subjects–Biology, Physics, History, Geography, Chemistry, Mathematics, and Computer Science–each contributing 5–7% (Fig. 7). The mix spans both STEM and humanities content without any technical subject dominating, so the statement generator is not over-specialised to a single domain.

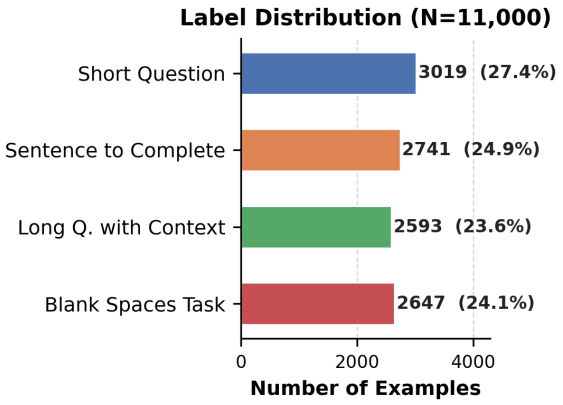

Figure 6: Distribution of question-type labels in CAP-Statements.

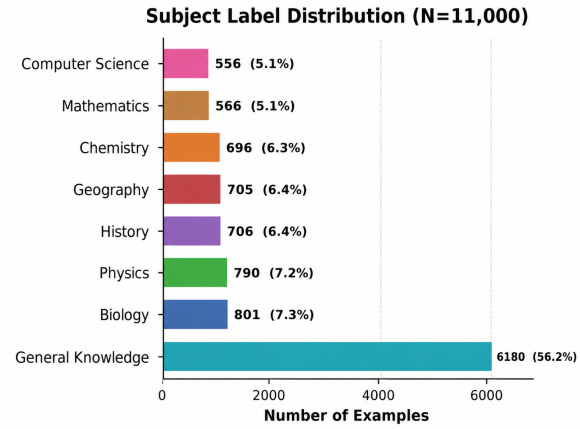

Figure 7: Distribution of subject labels in CAP-Statements.

Validation Protocol. To evaluate the quality of the resulting statement reformulations, we manually annotate a randomly sampled subset of 1,353 generated declarative statements (12.3% of CAP-Statements); annotation conditions and protocol are described in App. B.1. Each generated statement is assigned one of four labels–*correct*, *syntactic*, *semantic*, or *wrong*–following the taxonomy in Tab. 10.

Aggregated across question types (Fig. 8), the manual labels indicate that the statement generation pipeline is generally reliable: 90.0% of generated statements are labeled *correct*, with isolated *semantic* (1.3%) and *syntactic* (1.2%) errors both rare. The largest error category is *wrong* at 7.5%, which by inspection consists overwhelmingly of cases where the model returns a near-verbatim concate-

Label	Definition
correct	The generated statement preserves the full semantic meaning of the original question–answer pair without introducing grammatical or semantic errors.
syntactic	The generated statement contains syntactic or grammatical construction errors, but the underlying semantic meaning remains preserved.
semantic	The generated statement changes or distorts the semantic meaning of the original question–answer pair.
wrong	The generated statement fails as a faithful declarative reformulation, due to severe grammatical errors, semantic corruption, or both.

Table 10: Human evaluation taxonomy for generated declarative statements.

nation of the question and the answer rather than a properly reformulated statement; the residual syntactic errors are dominated by omitted determiners (most often *the*) and minor agreement mistakes that do not alter the underlying meaning. These patterns suggest the dominant failure mode is undertraining on harder reformulation cases rather than systematic semantic distortion.

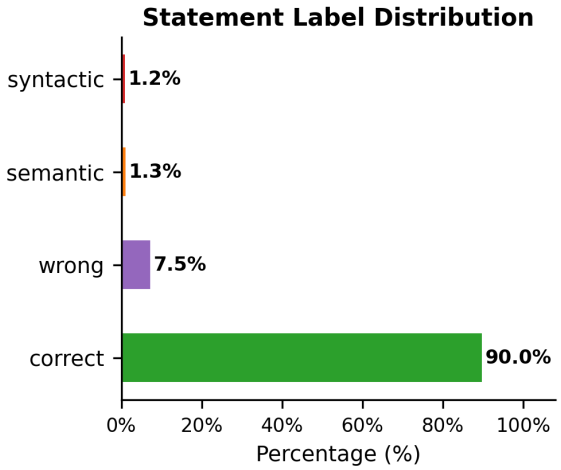

Figure 8: Errors in statements generation

This aggregate picture obscures one important asymmetry across question types: long-question items are substantially harder for the reformulator than the other three. As reported in Tab. 15, 32.9% of long-question statements are labeled *wrong* and 9.6% are labeled *semantic*, compared to a combined error rate below 1% on `sentence-to-complete` and `blank-spaces-task`. The practical consequence is that CAP's downstream reliability is question-type dependent: it is well-supported by the statement generator on short-question, sentence-completion, and blank-spaces items, and degraded on long-context items, where statement-reformulation errors propagate into the NLI scoring stage. We treat this as a limitation of the current statement generator rather than of the CAP framework itself, and discuss in §9 natural directions for closing the gap–scaling the long-question training partition, or replacing mT5-base with a larger sequence-to-sequence reformulator.

C Experiment Details

C.1 Evaluation Measure Definitions

Rank correlation. We compute Spearman's ρ and Kendall's τ between metric scores $\mathbf{m}(\mathsf{g}, \mathsf{p})$ and the integer ordinal rank $\mathbf{r}(\mathsf{c})$, treating tied taxonomy ranks (e.g., `exact` $\approx$ `equivalent` $\approx$ `alternative-correct`) using the standard mid-rank convention.

Pairwise ranking accuracy. Following the random baseline of 0.5, pairwise accuracy is defined as

$$\mathbf{PairAcc}(\mathbf{m}) = \mathbb{E}_{(a,b)\sim\mathcal{D}}\left[\mathbb{1}_{\{\mathbf{m}(a)>\mathbf{m}(b)\}}\right], \quad (8)$$

i.e., the fraction of class-ordered pairs on which the metric assigns a higher score to the more correct answer. Pairs (a, b) are sampled such that $\mathbf{r}(\mathsf{c}_a) > \mathbf{r}(\mathsf{c}_b)$ under the strict (non-tied) part of the taxonomy ordering.

Monotonicity violations. Let $\boldsymbol{\mu}_{\mathbf{m}}(\mathsf{c})$ denote the mean metric score on class c. The taxonomy induces 27 strictly ordered class pairs (excluding the single ambiguous comparison between `partial` and `overinclusive-valid`; see §3). A pair $(\mathsf{c}_i, \mathsf{c}_j)$ with $\mathsf{c}_i \succ \mathsf{c}_j$ is a *violation* if $\boldsymbol{\mu}_{\mathbf{m}}(\mathsf{c}_i) \leq \boldsymbol{\mu}_{\mathbf{m}}(\mathsf{c}_j)$. The violation rate $\mathbf{V}(\mathbf{m}) \in [0, 1]$ is the fraction of violating pairs out of 27.

Hard neighboring-pair accuracy. To isolate locally challenging distinctions, we additionally report pairwise accuracy restricted to adjacent classes in Eq. (1), focusing on the four main contrasts reported in Tab. 4.

C.2 Implementation Details

NLI backbone. CAP scores are computed with `cross-encoder/nli-deberta-v3-large`, a publicly available NLI cross-encoder from the Hugging Face Hub. We additionally evaluated `MoritzLaurer/mDeBERTa-v3-base-mnli-xnli`

and `joeddav/xlm-roberta-large-xnli` during development but selected `nli-deberta-v3-large` for its sharper contradiction-class separation on the CAP-Correctness validation subset. Inputs are tokenized as $(\text{premise}, \text{hypothesis})$ pairs with a maximum joint length of 512 tokens. Entailment, neutral, and contradiction probabilities are read from the softmax output of the classification head. Inference is run in mixed precision (`torch.float16`) with batch size 16.

Directional scoring. For each $(\mathsf{q}, \mathsf{g}, \mathsf{p})$ triple, CAP requires two NLI passes: forward ($\mathsf{g} \to \mathsf{p}$) with $\mathbf{s}(\mathsf{q}, \mathsf{g})$ as premise and $\mathbf{s}(\mathsf{q}, \mathsf{p})$ as hypothesis, and reverse ($\mathsf{p} \to \mathsf{g}$) with the two swapped. Both passes use the same NLI checkpoint without further fine-tuning.

CAP hyperparameters. Unless stated otherwise we use $\alpha = 0.85$, $\lambda = 0.3$, selected on a held-out subset of CAP-Correctness; the full sweep is reported in Tab. 11.

Ordinal rank. The taxonomy ordering of Eq. (1) is realized as an integer severity map when computing rank correlations and monotonicity violations: `exact` $= 7$, `equivalent` = `alternative-correct` $= 6$, `overinclusive-valid` $= 5$, `partial` $= 4$, `overinclusive-invalid` $= 2$, `invalid` $= 1$, `contradictory` $= 0$. Tied severities (`equivalent` vs. `alternative-correct`) contribute no strict pair and are not counted toward the monotonicity total. Spearman and Kendall correlations use the standard mid-rank convention for tied ordinal ranks.

Statement generator. We fine-tune mT5-base (Xue et al., 2021) on the 8,800 training examples of CAP-Statements with the AdamW optimizer, learning rate 5×10^{-4}, batch size 16, and a maximum of 10 epochs with early stopping on validation BLEU. Decoding uses beam search with width 4 and a maximum output length of 64 tokens. Final test performance is reported in §5.

Baseline metric versions. BLEU and METEOR are computed with `nltk.translate` (`sentence_bleu` and `meteor_score` respectively); ROUGE-L is computed with the `rouge_score` package (`rouge_scorer`, `rougeL` variant). BERTScore uses the F1 variant via the `bert_score` package with its default English checkpoint. COMET scores are computed via the `comet` package with the `Unbabel/wmt22-comet-da` checkpoint.

LLM-generated answers. For §7.4 we collect free-form answers to a 1k sample from CAP-Correctness questions from GPT-4o (`gpt-4o-2024-08-06`), Gemini 2.0 Flash (`gemini-2.0-flash`), and Qwen3-8B-Instruct (`Qwen/Qwen3-8B-Instruct`), each prompted in a zero-shot setting with temperature 0.0 and a 256-token output cap.

Prompt template. The same template is issued to every model and every question; the `{question}` placeholder is the only field that varies per call.

```
SYSTEM:
Answer the question in the style of a
person responding on an exam: concise,
factually correct, and free of
elaboration. Use a single short
sentence when sufficient; never exceed
three sentences. Do not restate the
question, include preamble or
qualifications, or open with phrases
such as "the answer is. . . ". Respond
in plain prose without any markdown
formatting: no bold, italics, headers,
lists, or code blocks.

USER:
{question}
```

Hardware. All NLI inference and mT5 fine-tuning are run on a single NVIDIA A100 40 GB GPU; no experiment requires distributed training.

C.3 CAP Hyperparameter Sweep

We select α and λ via a grid sweep on a held-out subset of CAP-Correctness, evaluating each setting on Spearman ρ, pairwise ranking accuracy, and monotonicity violation count (Tab. 11). Moderate neutral weighting ($\lambda = 0.3$) combined with asymmetric scoring ($\alpha = 0.85$) achieves the best ranking and pairwise accuracy while matching the lowest violation count, outperforming both purely entailment-based ($\lambda = 0$) and purely unidirectional ($\alpha = 1$) variants. This supports the intuition that partial credit for neutral relations improves robustness on semantically incomplete and alternative-correct answers while preserving strong contradiction separation.

α	λ	Spearman ρ	PairAcc	Violations
0.3	0.0	57.3	73.5	4/27
0.5	0.0	58.8	74.4	4/27
0.5	0.3	59.3	74.8	4/27
0.7	0.0	59.7	74.9	4/27
0.85	**0.3**	**60.7**	**75.6**	**4/27**
1	0.3	59.8	75.1	5/27

Table 11: Effect of directional asymmetry (α) and neutral-class weighting (λ) on semantic ranking consistency.

D Results

D.1 Monotonicity Analysis

While the main results show that CAP better captures the taxonomy's global ranking structure, monotonicity asks a sharper question: does CAP preserve the taxonomy's ordering on a class-by-class basis? Recall from §6.2 that a metric incurs a violation on a strictly ordered pair $(\mathtt{c}_i, \mathtt{c}_j)$ whenever the mean score for the higher-severity class falls below the mean score for the lower-severity one.

Lexical metrics and BERTScore invert roughly half of the ordering. Even COMET, the strongest baseline, violates $9/27$ comparisons it is expected to respect. CAP reduces this count to $4/27$: on 23 of the 27 ordered pairs, the mean CAP score for the more correct class is higher than for the less correct one (see Tab. 12).

Fig. 3 makes the score geometry visible. CAP's per-class distributions occupy progressively lower ranges as we move down the taxonomy, with classes forming largely distinct bands on the score axis. COMET's distributions, by contrast, collapse into a narrow region with heavy overlap across nearly every class: only `exact` separates cleanly, while `equivalent`, `partial`, the overinclusive classes, and even `invalid` answers occupy largely the same range. CAP's monotone ordering is visible on the score axis; COMET does not exhibit a comparable ordering.

The four class-pair inversions made by CAP, reflect two distinct failure modes. First, `alternative-correct` scores below the other top-tier classes. This class is difficult for NLI-based scoring because a valid alternative need not entail the gold answer, nor be entailed by it: for example, *Jupiter* and *Mars* may both be valid answers in context, while remaining mutually non-entailing. This effect is amplified by the pretrained NLI model's uneven world knowledge across CAP-Correctness topics, which can prevent it from recognizing valid alternatives. The second failure mode is `partial` scores above their expected positions, a structural artifact of bidirectional NLI that we examine in detail in the next section. Despite these localized inversions, CAP preserves the taxonomy's ordering far more consistently than any of the baselines.

Metric	Violating Class Pairs	Violation Rate
BLEU	14 / 27	51.85
ROUGE-L	11 / 27	40.74
METEOR	11 / 27	40.74
BERTScore	12 / 27	44.44
COMET	9 / 27	33.33
CAP	**4 / 27**	**14.81**

Table 12: Monotonicity violations, defined as cases where the mean score of a lower-severity class exceeds the mean score of a higher-severity class.

E Error Analysis

We dissect CAP's residual errors along three axes: statement-generation quality (Tab. 15), label-source stability (16 and 17), and the per-class structure of the failure modes identified in §7.

E.1 Ablation study

To assess the contribution of CAP's bidirectional formulation, we compare it against unidirectional entailment-scoring variants. Entailment-only scoring substantially reduces semantic ranking consistency, especially for semantically incomplete answers such as partial responses. Incorporating the reverse direction improves the modeling of semantic completeness, supporting our hypothesis that bidirectional entailment captures asymmetric semantic information missed by one-directional evaluation.

Variant	Spearman ρ	Kendall τ	PairAcc
Entailment $gold \rightarrow pred$	56.76	41.29	72.50
Entailment + λ Neutral	59.81	46.12	75.14
Entailment $pred \rightarrow gold$	39.07	28.02	65.27
Entailment + λ Neutral	40.82	29.73	66.20
Bidirect avg entailment	58.77	44.83	74.43
Full CAP	**60.69**	**46.94**	**75.58**

Table 13: Ablation study of CAP components.

E.2 Per-Model Ranking Statistics

Per-model ranking statistics (Tab. 14) give the same picture: GPT-4o and Qwen 3 yield correlations comparable to those on CAP-Correctness, with

Gemini Flash a notable outlier. The likely cause is Gemini's output style–more verbose/hedged answers. Verbose exacts lose entailment certainty (extra material the gold doesn't entail back), and verbose partials end up retaining more of the gold than a partial would.

Model	Spearman ρ	Kendall τ	PairAcc
GPT-4o	61.67	48.38	78.44
Gemini Flash	39.71	30.10	67.47
Qwen 3	56.82	44.45	76.47

Table 14: CAP ranking statistics on human-labeled LLM-generated answers, per model.

Statement-generation reliability across question types CAP's reformulation step relies on the mT5 statement generator, whose error rate is markedly uneven across question types (Tab. 15; full breakdown in App. B). Sentence-completion and blank-spaces items are produced cleanly, short-question items show a moderate *wrong* rate; long-context items are the clear bottleneck at 32.9% *wrong* and 9.6% *semantic* errors. Downstream CAP scores therefore inherit a question-type-dependent reliability, with the dominant source of error attributable to the statement generator on long-context inputs rather than to the metric itself.

Question Type	syntactic	semantic	wrong
Short question	3.0%	0.6%	10.5%
Long context	3.6%	9.6%	32.9%
Sentence completion	0.2%	0.0%	0.0%
Blank-space task	0.4%	0.0%	0.0%

Table 15: Human validation of generated statements by question type. Percentages indicate statements judged syntactically valid, semantically faithful, or incorrect during manual evaluation.

Stability under synthetic vs. human labels

We re-evaluate CAP on the 573-example human-validated subset using both label sources–the synthetic labels from the LLM-assisted pipeline and the labels independently assigned by human annotators against the same taxonomy. Tab. 16 shows that CAP correlates marginally *better* with the human-labeled ordering ($\rho = 57.20$, PairAcc $= 74.35$) than with the synthetic one ($\rho = 52.25$, PairAcc $= 71.89$): the headline numbers in Tab. 2 are not an artifact of the synthetic pipeline; if anything, synthetic labels modestly understate CAP's alignment with human judgement.[1]

Label Source	Spearman ρ	Kendall τ	PairAcc
Synthetic	52.25	39.40	71.89
Human-labeled	57.20	43.31	74.35

Table 16: Stability of CAP evaluation under synthetic and human semantic labels.

Tab. 17 breaks the comparison down by class. Six of eight classes shift by less than two points; the two large shifts are concentrated where the main results already identify failure modes (§7.2). The `alternative-correct` mean drops from 37.54 under synthetic labels to 26.79 under human labels ($\Delta = +10.75$): the cases human annotators correctly identify as alternative-correct receive even lower CAP scores than the pipeline's synthetic alternative-correct examples, confirming that the low alt-correct mean is a property of bidirectional NLI on factually distinct alternatives and not a labeling artifact. `Partial` drops by 7.50 points (87.21 → 79.71), narrowing–but not eliminating–the structural inversion against `overinclusive-valid` described in §7.3.

Label-semantic synthetically generated vs human label

Class	Synthetic	Annotator	Δ
Exact	98.84	98.88	-0.04
Equivalent	86.53	88.23	-1.70
Partial	87.21	79.71	-7.5
OV	50.79	49.65	-1.14
OI	38.46	38.22	0.54
Invalid	12.32	17.16	-4.84
Contradictory	5.14	6.63	-1.49
Alt.	37.54	26.79	10.75

Table 17: Mean CAP scores per semantic class under synthetic and human label sources.

Per-class failure modes

The per-class means in Tab. 17 concentrate CAP's residual error in two locations. `Partial` sits above its expected taxonomic position (mean 87.21 synthetic, 79.71 annotator), and `alternative-correct` sits below the top group (mean 37.54 synthetic, 26.79 annotator). The remaining classes–including

[1] Both values sit below the full-benchmark $\rho = 60.69$ of Tab. 2 because the validation subset is small (573 vs. 10,346 examples) and was sampled without per-class stratification.

overinclusive-invalid–occupy roughly their expected positions.

Partial scores too high. The mechanism is the structural NLI artefact described in §7.3. Partial answers are subsets of the gold information, so the dominant g $\rightarrow$ p direction–carrying weight $\alpha = 0.85$–measures whether the gold entails the partial answer, which it largely does. The reverse direction p $\rightarrow$ g correctly fails to entail (the partial is missing content) but carries only $1 - \alpha = 0.15$ of the weight. CAP therefore inherits the high gold-side entailment and ranks partial answers near the top group rather than mid-taxonomy.

Overinclusive-valid scores too low. overinclusive-valid is the mirror case. The answer contains the gold plus additional valid content, so p $\rightarrow$ g entails strongly–but this is the down-weighted direction. The dominant g $\rightarrow$ p direction sees gold-plus-extras and assigns mostly neutral mass, since the extras are not supported by the gold. The asymmetric weighting amplifies this neutral signal and pushes overinclusive-valid below partial, inverting the taxonomy on this specific axis.

Overinclusive-invalid is positioned correctly. In contrast to the two cases above, overinclusive-invalid behaves as expected. Its mean (38.46 synthetic, 38.22 annotator) sits below overinclusive-valid because the invalid extras introduce contradiction mass that the entailment + $\lambda\cdot$ neutral score does not reward, and clearly above invalid and contradictory because the correct content is still recognized. CAP separates "correct with false extras" from "correct with valid extras" and from "fully invalid" in the expected order. The two overinclusive classes are difficult for surface-similarity baselines (Tab. 4, OV > OI column), but they are not where CAP itself fails.

Class	Class-specific instruction
exact	Reproduce the gold answer verbatim.
equivalent	Express the same meaning as the gold answer using different wording. Do not omit or add information.
alternative-correct	Produce a different but equally valid answer to the question. The answer must be factually correct and must not be entailed by the gold answer in either direction.
partial	Reproduce a strict subset of the information in the gold answer. Do not add information beyond what the gold states.
overinclusive-valid	Include the full content of the gold answer and add one short piece of additional information that is factually correct and relevant to the question.
overinclusive-invalid	Include the full content of the gold answer and add one short piece of additional information that is factually incorrect or unsupported.
invalid	Produce a plausible-sounding answer to the question that is factually wrong. Do not directly contradict the gold answer.
contradictory	Produce an answer that directly contradicts the gold answer or rejects the premise of the question.

Table 18: Class-specific instructions used to instantiate the candidate-answer generation prompt. The {class_instruction} placeholder in the template is filled with the row matching the target class.