Logic-Level Evaluation of Logical Table-to-Text Generation

Lena Trigg and Ahsan Bilal and Dean F. Hougen
School of Computer Science, University of Oklahoma
Norman, OK, USA
lena.trigg@ou.edu, ahsan.bilal-1@ou.edu, hougen@ou.edu

Abstract

Logical Table-to-Text (LT2T) generation aims to produce natural-language sentences that are logically faithful to structured tabular data. While recent Large Language Models (LLMs) show high performance on aggregate fidelity metrics, these scores provide only a coarse view of performance, obscuring specific logic-type reasoning failures and models' meta-logical awareness. We propose an operation-aware diagnostic framework that evaluates four core competencies: (1) Logical Form (LF) execution accuracy, (2) fidelity of LF-conditioned generation, (3) logic-type identification, and (4) LF-free generation.

We apply this framework to a suite of frontier LLMs and perform fine-grained analysis across logic types such as aggregation, ordinal, and superlative reasoning. Our results show that LT2T fidelity assessment can be unstable; the choice of verifier and logic type can substantially alter conclusions and model rankings. Crucially, we identify a meta-logical gap: models often generate faithful statements while failing to identify the underlying operation.

1 Introduction

Logical Table-to-Text (LT2T) generation aims to produce natural-language descriptions that are both fluent and logically faithful to structured tabular data. Unlike surface-level data-to-text generation, LT2T requires explicit reasoning over table entries, including comparison, aggregation, counting, and arithmetic operations. Ensuring logical fidelity is critical for real-world applications such as scientific reporting, financial summarization, and decision support, where even minor reasoning errors can lead to misleading or incorrect conclusions.

Recent advances in large language models (LLMs) have led to substantial improvements across a wide range of natural language processing tasks, including reasoning-intensive generation. LLMs demonstrate strong performance in LT2T settings through in-context learning and chain-of-thought prompting (Zhao et al., 2023b), often achieving high scores on existing logical fidelity metrics without fine-tuning. However, these gains are typically reported at an aggregate level, providing limited insight into the models' underlying reasoning behavior or their performance on specific logical operations.

We argue that relying solely on aggregate scores obscures critical failure modes. In practical applications such as data journalism or dashboard narration, users depend on the reliability of specific operations, such as identifying the largest value or calculating counts. A model may achieve a high average score yet remain fundamentally unreliable for specific, decision-critical logic types such as Ordinals or Aggregation. Furthermore, existing evaluations fail to reveal the source of these errors: is it a failure in grounding, operator selection, or linguistic realization? Without this granularity, we lack the diagnostic clarity needed to build more robust systems or implement targeted fixes.

In this work, we move beyond aggregate evaluation to present an operation-aware diagnostic analysis of LLM-based LT2T systems. We systematically investigate model behavior across diverse logical operations through four research questions: **(RQ1) Execution:** How accurately can models execute logical forms against tables? **(RQ2) Fidelity:** How faithfully do they realize a given LF into a natural language sentence? **(RQ3) Identification:** Can models recognize the logic type implied by a table-statement pair? **(RQ4) Generalization:** Can models generate a novel, entailed sentence of the same logic type without an LF? We translate these questions into a suite of targeted tasks and evaluate models per logic type to uncover operation-specific strengths, failure modes, and scaling trends that are invisible under aggregate fidelity scores.

Our operation-aware analysis shows that high

Proceedings of the 30th Conference on Computational Natural Language Learning, pages 677–691

aggregate LT2T scores can mask important weaknesses. We highlight recurring phenomena: (1) a meta-logical gap, where models produce outputs judged faithful while struggling to identify the underlying operation; (2) metric instability, where conclusions and model rankings vary substantially with verifier choice. By decomposing the task into specific competencies, our framework provides a more granular evaluation, exposing the disconnect between a system's linguistic fluency and its underlying logical reasoning.

2 Related Work

Logical Table-to-Text Generation. LT2T generation extends standard table-to-text tasks by requiring models to realize explicit logical reasoning, such as comparison or counting, as faithful natural language statements. Following the taxonomy of Trigg and Hougen (2025), existing approaches address this requirement either by explicitly conditioning on symbolic structure, such as logical forms (Chen et al., 2020b) or programs executed over the table (Zhao et al., 2023a), or by implicitly inducing reasoning behaviors through pretraining (Liu et al., 2022) or distillation (Yang et al., 2024). More recently, studies on LT2T with LLMs report strong overall performance (Zhao et al., 2023b). However, aggregate metrics can mask sharp operation-specific failures; systematic, operation-aware analysis by logic type remains underexplored.

Evaluating Logical Fidelity in Table Reasoning. A significant body of evaluation work in table reasoning and LT2T treats fidelity as table entailment. In this paradigm, a model predicts whether a generated statement is entailed by the table, using the entailment rate as a proxy for logical fidelity. Examples include NLI-Acc (Chen et al., 2020a) and table-aware variants like TAPAS-Acc and TAPEX-Acc (Liu et al., 2022). While these approaches are robust to linguistic paraphrasing, they overlook fine-grained operator or argument errors and remain sensitive to how the table is linearized.

Complementarily, semantic-parsing-based evaluation (Chen et al., 2020a) attempts to map the generated sentence back to a formal program or an LF. By executing this LF against the table, this method enforces a stricter notion of program recoverability. However, this approach is inherently brittle; it frequently penalizes valid linguistic variations and depends heavily on the parser's ability to resolve ambiguity. Recognizing that these verifier families capture distinct dimensions of correctness, our study provides operation-level diagnostics to pinpoint where models, and the verifiers themselves, succeed or fail.

3 Task Setup and Problem Formulation

This section formally introduces the problem, logic types, and tasks the models are to complete.

3.1 Tables and Logical Forms

We study *LT2T* generation, where inputs are semi-structured tables, and outputs are natural language statements that are logically entailed by a table.

Let a table be denoted as $T = \{ T_{i,j} \mid 1 \leq i \leq R_T,\ 1 \leq j \leq C_T \}$, where R_T and C_T are the number of rows and columns, respectively, and each cell entry $T_{i,j}$ may contain a word, a number, a phrase, or even an entire sentence.

A *logical form* (LF), denoted p, represents the multi-step reasoning required to derive an answer from the table. Executing p on T yields a final value v such as a scalar or boolean.

3.2 Logic-Type Taxonomy

To support operation-level analysis, we assign each LF/sentence to a *logic type* $l \in L$, where L consists of *Count*, *Aggregation*, *Superlative*, *Ordinal*, *Comparative*, *Majority*, and *Only*.

3.3 Task Definitions

To answer our research questions, we design an operation-aware diagnostic framework consisting of four complementary tasks: (1) LF execution and evidence grounding, (2) LF-conditioned generation fidelity, (3) logic-type recognition, and (4) LF-free type-controlled generation.

Figure 1 provides a running example that illustrates the input and expected output for each task. The same table and logical form are used across the figure to show how the framework decomposes LT2T evaluation into separate reasoning abilities. The figure shows how the inputs are transformed into four different expected model outputs: an evidence-grounded execution trace for Task 1, an LF-conditioned natural-language text for Task 2, a predicted logic-type label for Task 3, and a new text without an LF for Task 4.

By presenting the tasks side by side, the example illustrates how our framework separates different aspects of LT2T reasoning: execution, realization, operation recognition, and LF-free generalization.

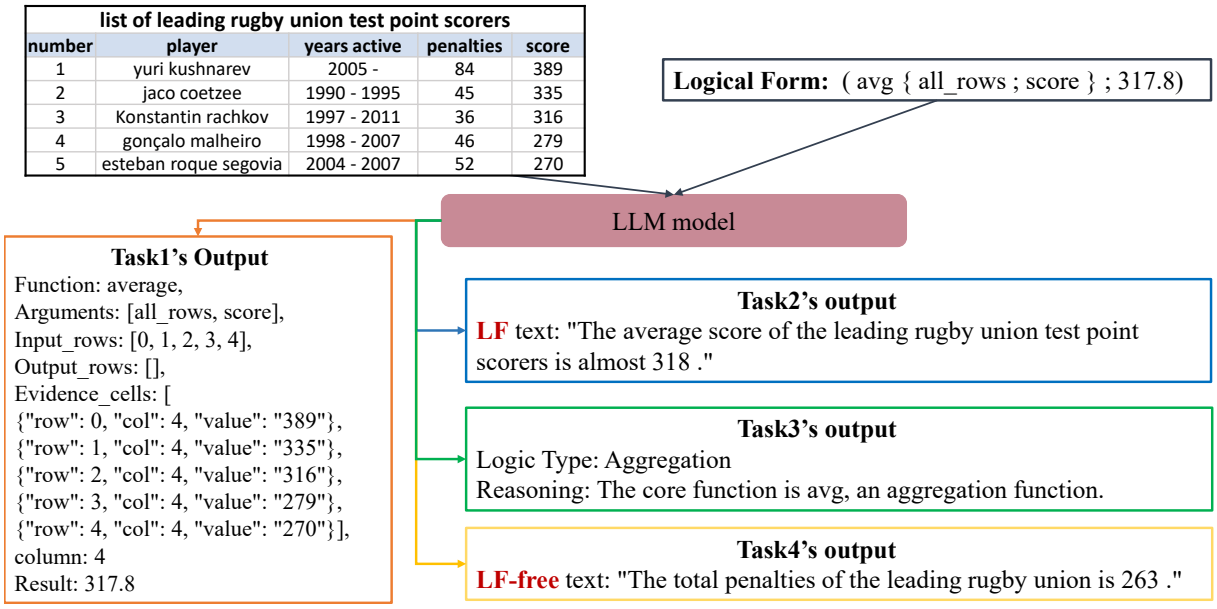

Figure 1: Running example for the four LT2T diagnostic tasks.

These dimensions are usually hidden by a single aggregate fidelity score.

Task 1 (Execution Trace): This task examines a model's ability to execute an LF and ground each intermediate step in table evidence. Given a table and a logical form, the model outputs a nested execution tree τ aligned with the LF structure. For each operator node, the trace records the function name, its arguments, including nested subcalls, intermediate results, the input/output row sets, and the evidence cells used. This task captures *where* the model sources its evidence and enables step-level error attribution, such as incorrect filter conditions versus aggregating over the wrong column. For example, for the LF in Figure 1, the model should identify all table rows, select the score column, extract the score values, compute their average, and return the final result together with the evidence cells used in the computation.

Task 2 (Entailed Claim Generation): This task measures a model's ability to generate an entailed logical sentence when an LF is available. Given a table and a logical form, the model generates a single natural-language sentence y that expresses the final result of executing p on T. This task evaluates whether a model can faithfully *realize* a given LF in text by expressing the computed value(s) while respecting the intended operator semantics and arguments (columns, rows, constants), without introducing unsupported facts. As depicted in Figure 1, a valid output is: "The average score of the leading rugby union test point scorers is almost 318."

Task 3 (Logic-Type Identification): Given the table, and the generated statement in Task 2, the model predicts a logic-type label $l \in L$ and provides a short rationale for the prediction. This evaluates operation recognition, such as *Ordinal* vs. *Superlative*, not captured by standard fidelity scores. For the example in Figure 1, the correct logic type is Aggregation because the central operation is computing an average.

Task 4 (LF-Free Claim Generation): Given the table, the model must generate a new entailed sentence y' grounded in a different part of the table, without access to an LF. This task tests whether the model can generate an entailed sentence without explicit guidance from an LF. As illustrated in Figure 1, a valid LF-free output is: "The total penalties of the leading rugby union is 263."

3.4 What These Tasks Enable

Together, the four tasks separate failure sources that are blurred by aggregate metrics: (1) execution/evidence errors, (2) LF-following realization errors, (3) logic-type recognition errors, and (4) realization without LF errors.

This formulation supports operation-level evaluation that can attribute errors to specific evidence rather than reporting only end-to-end correctness.

Logic type	Execution failure pattern (frontier models)
Aggregation	Average mis-calculation (wrong mean or precision) Sum mis-calculation Rounding / threshold tolerance (e.g., 21.57 vs 21.43) Row-filtering issue (extra or missing rows before aggregation)
Comparative	Numeric-diff mismatch (diff correct but format/string off) Unsupported operation/format (date arithmetic, mixed units) Unit-token mismatch ("6 points" vs "6") Composite-logic failure (sub-clause in an AND/OR chain flips the whole result)
Count	Under-count (qualifying rows missed) Over-count (extra rows counted) Row-filtering issue (substring filters too broad/narrow) Parser / format mismatch (numbers with commas, "15 July" vs "July 15")
Majority	Fuzzy vs. strict equality (`most_eq` vs `most_str_eq`) Off-by-one majority threshold (predicate true, model says false) Comparator-swap / argument-swap ("home" vs "away")
Ordinal	Fuzzy vs. strict equality (`eq` vs `str_eq`) Wrong ordinal index / tie handling (`nth`, `argmin`/`argmax`) Parser / format mismatch
Superlative	Fuzzy vs. strict equality Row-filtering issue (pre-filter too loose)
Only	Row-filtering issue (returns 0 or >1 rows) Fuzzy vs. strict equality

Table 1: Most common LF execution failure patterns aggregated across frontier models, grouped by logic type.

4 Experiments and Results

This section covers the test set, metrics, and results on all research questions.

4.1 Test Set

We build a 700-instance test set by sampling from the Logic2Text development and test splits, comprising 100 instances for each logic type. Model outputs are grouped by their gold logic type, and results are reported separately for every type.

4.2 Automated Logical Fidelity Metrics

We evaluate sentence-level logical fidelity using two common families of automatic metrics discussed in Section 2: (1) Entailment-based verification, including NLI-Acc, TAPAS-Acc, and TAPEX-Acc. (2) Parsing-based execution: Sp-Acc. Details are provided in Appendix A.

4.3 Evaluation Results

We present results organized by our research questions, corresponding to Tasks 1- 4. Unless stated otherwise, we report the score of the automated metrics (1) per logic type; (2) the macro-average across logic types for each model to summarize overall capability; (3) the cross-model mean for each logic type to characterize type difficulty.

RQ1: Logical Form Execution Accuracy Our first research question asks: *How well can models execute a gold LF against an input table?* We answer this in two stages, both relying on a trace-level comparison between the model-generated output and the dataset's reference execution.

(1) **Per-operation accuracy:** We evaluate LF execution accuracy using a flexible execution-based comparison rather than exact trace matching. A model output is accepted when it reaches the correct final result and follows the same execution flow even if some operator labels differ from the gold trace. For example, for filtering operations, we accept `filter_eq` in place of `filter_str_eq` when the selected operation produces the correct filtered rows and final answer.

Table 7 (Appendix B) presents the ability of models to *execute* LFs against tables. Overall, Frontier-scale models achieve higher accuracy, at 90.4%, compared with 86.1% for Small/Distilled models; however, performance is not uniform across logic types. In particular, both groups exhibit their weakest results on *Aggregation* (Frontier: 85.4%, Small/Distilled: 78.2%), suggesting that multi-step numerical computation remains fragile even for high-capacity models. In addition, logic-specific errors in Table 1 underscore the prevalence of com-

putational errors in these cases. In contrast, *Majority*, *Superlative*, and *Only* tend to be more reliable across both tiers (Frontier: 94.6%, 92.8%, 92.6%; Small/Distilled: 91.6%, 89.2%, 88.4%), consistent with these operators often reducing to simpler combinations of filtering and counting/extrema. Notably, scaling benefits are not evenly distributed: the frontier–small gap is largest for *Aggregation* and *Comparative*, 7 points for each, while it is smaller for *Majority* and *Superlative*.

(2) **Failure-pattern taxonomy:** To move beyond binary metrics, we leverage the tree-structured nature of execution traces, which capture operators, arguments, and intermediate row/column subsets, to perform fine-grained mismatch attribution. Through manual inspection of Deepseek-v3.2 mismatches and LLM-assisted summarization for other frontier models, we cluster these errors into the taxonomy presented in Table 1.

Across all evaluated models, failures separate into cross-cutting and logic-specific categories. Cross-cutting errors, such as incorrect row filtering and inconsistencies in fuzzy versus strict string matching, occur across most operations and constitute the majority of the error mass. In contrast, logic-specific errors stem from operator-sensitive pitfalls, such as miscalculating averages during aggregation. This taxonomy clarifies why executions fail and motivates straightforward mitigations, including value normalization, tolerant numeric comparison, and explicit operator disambiguation.

RQ2: LF-to-Text Realization Fidelity (Task 2) Our second research question investigates: *How well can LLMs generate sentences that are both logically entailed by the table and faithful to the provided LF?* To investigate this, first, we apply automatic logical fidelity metrics to address the former (Tables 2-3). In these tables, the section background colors distinguish evaluators (NLI-Acc, TAPAS-Acc, TAPEX-Acc), bold indicates the strongest score within each comparison block, red highlights the weakest average logic type, and green and tan row shading emphasize the selected base model vs. the distilled version in the comparison. Second, because these metrics primarily capture entailment rather than LF compliance, we additionally conduct a human LF-adherence evaluation to assess whether models truly realize the intended LF (Table 4).

1) Overall Trends Across Evaluators: TAPEX-Acc is generally the most optimistic, NLI-Acc occupies a middle ground, and TAPAS-Acc is more conservative, especially on count-based structural logic. SP-Acc is much lower overall, reflecting the stricter nature of parsing-and-execution evaluation.

2) Logic-Type Difficulty: Task 2 performance varies sharply across logic types. Although absolute accuracies depend on the verifier, difficulty ordering across logic types is almost stable.

Ordinal is consistently the hardest type, exhibiting the lowest average performance across NLI-Acc, TAPAS-Acc, and TAPEX-Acc. Based on this and failure patterns in Table 1, we conclude that ranking and ordering constraints remain fragile, as models frequently struggle with incorrect indexing, tie-breaking, or parser mismatches.

Superlative and Only are the easiest types. Across the same evaluators, these logic types achieve the highest accuracies. These operations map naturally to common lexical patterns such as highest, lowest, and only, making LF realization comparatively easier.

Comparative, Majority, and Count occupy a middle tier. *Comparative* is generally strong under table-aware evaluators, while errors stem from format-sensitive issues such as numeric–string mismatches or unit inconsistencies. *Majority* is often moderate to strong but susceptible to fuzzy vs. strict equality, quantifier realization, and scope errors. *Count* typically performs in the middle, with common errors arising from counting the wrong subset due to missing or incorrectly realized filters.

SP-Acc highlights aggregation-specific brittleness. This aligns with the inherent challenges of semantic parsing under linguistic variation: aggregation statements permit diverse paraphrases, such as *total*, *sum*, or *average*, where minor mis-parsing of a rounding threshold or a row-filter can flip the execution outcome even when the generated sentence remains table-consistent.

3) Frontier vs. Small/Distilled models: Performance gaps are not uniform across evaluators, models, or logic types. Using cross-model group averages, Frontier models lead on *Comparative*, *Only*, and *Superlative*, with margins of 2 to 11 points. In contrast, Small/Distilled models remain competitive with and sometimes surpass Frontier peers on *Aggregation* under TAPAS/TAPEX-Acc, and on *Ordinal* under NLI-Acc and TAPAS-Acc. SP-Acc amplifies nearly all gaps, especially for *Aggregation*, where strict parsing penalizes smaller checkpoints more sharply.

Model	Aggregation	Comparative	Count	Majority	Ordinal	Superlative	Only	Average
NLI-Acc								
Llama3.1-405B	59.0	83.0	76.0	80.0	49.0	90.0	86.0	75.0
Qwen3-235B-A22B	60.0	80.0	75.0	78.0	43.0	**96.0**	92.0	75.0
Deepseek-v3.2	62.0	85.0	68.0	79.0	55.0	90.0	90.0	76.0
GPT5.1	60.0	**86.0**	77.0	76.0	50.0	91.0	**94.0**	76.0
GPT4.1	58.0	**86.0**	**85.0**	81.0	45.0	94.0	91.0	**77.0**
Average (Frontier)	60.0	84.0	76.0	79.0	48.0	92.0	91.0	76.0
Llama3.2-3B	67.0	69.0	55.0	81.0	**68.0**	82.0	53.0	68.0
Gemma3-27B-IT	43.0	73.0	61.0	74.0	52.0	87.0	83.0	68.0
Deepseek-R1-Distill-Qwen32B	43.0	78.0	63.0	77.0	41.0	95.0	87.0	69.0
Deepseek-R1-Distill-Llama-70B	48.0	81.0	63.0	77.0	42.0	92.0	86.0	70.0
Llama3.3-70B	41.0	80.0	80.0	80.0	43.0	91.0	88.0	72.0
Qwen3-8B	49.0	74.0	74.0	82.0	41.0	94.0	87.0	72.0
Gemma3-12B-IT	55.0	72.0	78.0	83.0	46.0	91.0	85.0	73.0
Llama3.1-8B	58.0	85.0	69.0	79.0	64.0	84.0	73.0	73.0
Qwen3-32B	59.0	80.0	73.0	82.0	40.0	**96.0**	90.0	74.0
Gemma3-4B-IT	**69.0**	82.0	79.0	**86.0**	**68.0**	84.0	84.0	**79.0**
Average (Small/Distilled)	53.0	77.0	70.0	80.0	51.0	90.0	82.0	72.0
TAPAS-Acc								
GPT5.1	77.0	87.0	66.0	74.0	40.0	82.0	83.0	73.0
Deepseek-v3.2	79.0	86.0	61.0	76.0	44.0	84.0	87.0	74.0
Qwen3-235B-A22B	81.0	87.0	66.0	72.0	37.0	85.0	**91.0**	74.0
Llama3.1-405B	75.0	87.0	62.0	**78.0**	44.0	89.0	90.0	75.0
GPT4.1	80.0	**88.0**	67.0	75.0	41.0	89.0	89.0	**76.0**
Average (Frontier)	78.0	87.0	64.0	75.0	41.0	86.0	88.0	74.0
Llama3.2-3B	70.0	51.0	70.0	68.0	42.0	68.0	63.0	62.0
Llama3.3-70B	60.0	86.0	61.0	72.0	34.0	84.0	87.0	69.0
Deepseek-R1-Distill-Llama70B	79.0	82.0	53.0	75.0	35.0	78.0	88.0	70.0
Deepseek-R1-Distill-Qwen32B	79.0	79.0	58.0	73.0	37.0	79.0	84.0	70.0
Gemma3-4B-IT	81.0	72.0	**77.0**	72.0	53.0	75.0	76.0	72.0
Llama3.1-8b	**82.0**	77.0	65.0	69.0	46.0	90.0	76.0	72.0
Gemma3-12B-IT	80.0	78.0	69.0	76.0	46.0	79.0	81.0	73.0
Gemma3-27B-IT	74.0	82.0	66.0	68.0	**56.0**	79.0	85.0	73.0
Qwen3-8B	79.0	84.0	65.0	76.0	39.0	88.0	86.0	74.0
Qwen3-32B	81.0	84.0	**77.0**	71.0	42.0	**93.0**	86.0	**76.0**
Average (Small/Distilled)	77.0	78.0	66.0	72.0	43.0	81.0	81.0	71.0
TAPEX-Acc								
GPT5.1	73.0	85.0	76.0	74.0	50.0	81.0	83.0	75.0
Deepseek-v3.2	85.0	93.0	77.0	80.0	53.0	84.0	86.0	80.0
Qwen3-235B-A22B	86.0	**94.0**	81.0	73.0	50.0	84.0	**90.0**	80.0
Llama3.1-405B	87.0	87.0	**85.0**	80.0	57.0	88.0	88.0	**82.0**
GPT4.1	85.0	92.0	83.0	**82.0**	54.0	88.0	87.0	**82.0**
Average (Frontier)	83.0	90.0	80.0	78.0	53.0	85.0	87.0	79.0
Llama3.2-3B	74.0	48.0	64.0	65.0	45.0	71.0	55.0	60.0
Deepseek-R1-Distill-Qwen32B	93.0	81.0	64.0	76.0	47.0	74.0	84.0	74.0
Gemma3-4B-IT	84.0	73.0	78.0	72.0	**58.0**	78.0	74.0	74.0
Llama3.3-70B	68.0	83.0	70.0	78.0	49.0	84.0	**90.0**	75.0
Deepseek-R1-Distill-Llama70B	93.0	85.0	63.0	83.0	44.0	75.0	86.0	76.0
Gemma3-27B-IT	80.0	87.0	70.0	80.0	55.0	79.0	86.0	77.0
Llama3.1-8B	90.0	83.0	76.0	78.0	49.0	**94.0**	73.0	78.0
Qwen3-8B	88.0	87.0	74.0	79.0	51.0	89.0	86.0	79.0
Qwen3-32B	**93.0**	78.0	77.0	81.0	52.0	90.0	84.0	79.0
Gemma3-12B-IT	93.0	89.0	80.0	81.0	57.0	78.0	81.0	80.0
Average (Small/Distilled)	87.0	79.0	72.0	77.0	51.0	81.0	80.0	75.0

Table 2: Automated logical-fidelity evaluation on Logic2Text (higher is better).

Distilled variants do not yield consistent gains. For example, the distilled Qwen-32B model matches its base counterpart on *Aggregation* (93%) but drops by 3 to 13 points on *Comparative*, *Count*, and *Majority*, ultimately lowering its macro-average across NLI-based metrics. Similarly, while distillation strongly boosts *Aggregation* performance from 68% to 93% for Llama3.3-70B, this variant simultaneously loses points on *Count* and *Superlative*, resulting in only a marginal shift in its overall average.

Smaller models are not universally worse. They can be competitive on specific operators and under entailment-oriented metrics. For instance, *Gemma3-4B-IT* matches or exceeds frontier averages on NLI-Acc for several logic types. This suggests that targeted instruction tuning can deliver high logical fidelity even at very low parameter counts.

4) Metric Disagreement and Implications: Evaluator choice fundamentally shifts the narrative regarding logical fidelity. While TAPAS-Acc and TAPEX-Acc remain closely aligned, NLI-Acc shows moderate alignment, and SP-Acc diverges significantly. These discrepancies stem from core differences in evaluation targets. NLI, TAPAS, and TAPEX primarily test entailment-like correctness, while SP-Acc additionally tests whether statements can be mapped back into executable programs.

Because SP-Acc relies on a semantic parser, it is highly sensitive to surface realization and paraphrasing. Semantically accurate statements may still fail SP-Acc if they deviate from the parser's canonical templates. This sensitivity is most pronounced in *Aggregation* and *Comparative* categories, where phrasing is naturally diverse. Conversely, categories with lexically constrained realizations, such as *Superlative* and *Only*, show higher cross-metric agreement. Section 5 analyzes metric disagreements in more detail.

5) Human LF-Adherence Evaluation To probe whether Task 2 outputs faithfully realize the provided LFs, we conduct a focused human study on *Gemma3-27B-IT*. This model sits near the overall mean performance of both the *Frontier* and *Small/Distilled* groups under the more closely aligned table-based verifiers (TAPAS/TAPEX).

For each logic type, we annotate 40 model outputs sampled from three strata whenever possible: (1) *Agreement-Entailed (E/E)*: 15 instances where both verifiers predict entailed, (2) *Agreement-Refuted (R/R)*: 15 instances where both verifiers predict not-entailed/refused, and (3) *Disagreement*: 10 instances where TAPAS and TAPEX disagree. This stratified design allows us to measure LF-adherence while diagnosing whether verifier outcomes reflect true generation errors or evaluator noise (see Appendix C for details).

The human evaluator assigns one of four labels: **A** (Adheres), **O** (Off-LF but entailed), **N** (Not-entailed), and **U** (Unclear). Table 4 summarizes the results.

High LF adherence overall. Five of seven logic types exhibit almost 90% adherence; the exceptions are *Count* (50%) and *Superlative* (72%).

Verifier refutations often hide false negatives. In the *refuted-by-both* subset, almost 45% of cases are still rated **A** by human (73% for Aggregation, 100% for Majority, and at least 47% for Ordinal).

Dual acceptance is highly reliable. When both verifiers accept a claim, this indicates that combined acceptance is a strong indicator of LF faithfulness.

Task 2 confirms that LF-conditioned generation remains challenging and operation-dependent. While automatic verifiers (NLI/TAPAS/TAPEX) provide a useful entailment baseline, human inspection reveals that many *refutations* stem from evaluator limitations rather than model errors. Conversely, when verifiers agree on entailment, the outputs are consistently LF-faithful. Single aggregate scores thus conflate (1) genuine realization errors, (2) Off-LF yet entailed outputs, and (3) verifier false rejections, underscoring the need for more nuanced, operation-aware evaluation.

RQ3: Logic-Type Prediction (Task 3) Table 5 reports per-type and overall accuracy for logic-type prediction. Overall, most models achieve high accuracy on *Comparative* and *Majority* (with the exception of the smallest models), but struggle on *Ordinal*, *Aggregation*, and *Superlative*.

Figure 2 in Appendix D shows the confusion matrix and reveals systematic confusions. In particular, *Comparative* is over-predicted: instances from *Aggregation*, *Ordinal*, and *Superlative* are frequently misclassified as *Comparative*, indicating that models tend to default to a simple comparison between values even when the logic requires more complex operations like summing or ranking.

RQ4: LF-Free Claim Generation (Task 4) Task 4 evaluates whether models can generate a

Model	Aggregation	Comparative	Count	Majority	Ordinal	Superlative	Only	Average
SP-Acc								
Deepseek-v3.2	24.0	48.0	59.0	67.0	59.0	**83.0**	69.0	58.0
Llama3.1-405B	25.0	41.0	54.0	66.0	67.0	81.0	81.0	59.0
Qwen3-235B-A22B	18.0	42.0	62.0	69.0	66.0	80.0	73.0	59.0
GPT4.1	24.0	50.0	61.0	65.0	64.0	81.0	81.0	61.0
GPT5.1	44.0	**57.0**	61.0	**76.0**	**74.0**	81.0	77.0	**67.0**
Average (Frontier)	27.0	48.0	59.0	69.0	66.0	81.0	76.0	61.0
Llama3.2-3B	20.0	44.0	37.0	57.0	48.0	62.0	69.0	48.0
Gemma3-4B-IT	24.0	55.0	52.0	54.0	65.0	71.0	72.0	56.0
Gemma3-12B-IT	22.0	39.0	57.0	63.0	64.0	75.0	74.0	56.0
Llama3.1-8B	**32.0**	44.0	52.0	65.0	58.0	68.0	80.0	57.0
Qwen3-32B	30.0	41.0	57.0	68.0	57.0	78.0	73.0	58.0
Gemma3-27B-IT	30.0	43.0	57.0	55.0	65.0	**83.0**	76.0	58.0
Deepseek-R1-Distill-Llama70B	29.0	42.0	58.0	70.0	58.0	76.0	81.0	59.0
Qwen3-8B	30.0	42.0	53.0	63.0	68.0	82.0	79.0	60.0
Llama3.3-70B	41.0	41.0	60.0	64.0	59.0	79.0	80.0	60.0
Deepseek-R1-Distill-Qwen32B	29.0	49.0	**65.0**	73.0	59.0	78.0	**84.0**	62.0
Average (Small/Distilled)	29.0	44.0	55.0	63.0	60.0	75.0	77.0	56.0

Table 3: Automated logical-fidelity evaluation (SP-Acc) on Logic2Text.

Logic	Entailed$_{both}$	Refuted$_{both}$	Refuted$_{TAPAS}$	Refuted$_{TAPEX}$	A	N	O	U
Aggregation	15	15	7	3	36	3	1	0
Comparative	16	7	11	6	36	2	0	2
Count	15	15	5	5	20	16	3	1
Majority	15	13	6	6	40	0	0	0
Ordinal	15	15	5	5	32	5	2	1
Superlative	16	16	4	4	29	10	1	0
Only	17	6	9	8	35	5	0	0

Table 4: LF-adherence investigation by a human study.

new entailed claim *without* access to an LF. Tables 8 and 9 (Appendix E) report sentence-level correctness for Task 4 using the same automatic fidelity metrics.

Impact of Removing Logical Forms. Tables 2–3 (NLI/TAPAS/TAPEX) and Tables 8–9 (SP-Acc) contrast logical-fidelity scores *with* versus *without* the gold LF. ***Removing the LF consistently degrades performance***. The effect is most pronounced for TAPAS/TAPEX and for smaller or distilled models.

Frontier models. Macro-average TAPAS/TAPEX-Acc falls from 74.0 to 69.0 and from 79.0 to 73.0, respectively. SP-Acc drops from 61.0 to 56.0.

Small/Distilled models. The decline is sharper: TAPAS-Acc and TAPEX-Acc fall by 10 and 11 points, respectively, while SP-Acc decreases by 4.

Logic-level Impacts The LF yields the largest gains for operations that impose explicit constraints, notably *Majority*, *Superlative*, and *Only*. Without the LF, *Majority* accuracy collapses for frontier models (TAPAS: 75.0 to 46.0; TAPEX: 78.0 to 50.0), implying that unconstrained models struggle to preserve majority-style evidence requirements. Under SP-Acc, the steepest drops occur for *Superlative* (frontier: –19 pts; small/distilled: –23 pts) and *Only* (frontier: –9 pts; small/distilled: –15 pts). Interestingly, *Ordinal* sometimes *improves* when the LF is removed: for frontier models, TAPAS rises by 24 pts and TAPEX by 17 pts, consistent with models producing easier-but-entailed statements when not required to follow the target operation.

Overall, LFs provide strong operator-level guidance, with the greatest impact on capacity-limited models and on operations requiring non-trivial constraints.

5 Discussion

Metric Disagreement Analysis We identify, for each NLI-based verifier, the set of samples it flags as *not entailed* (i.e., *refused*), and then compute pairwise overlaps between these refusal sets to ob-

Model	Ordinal	Superlative	Aggregation	Count	Only	Majority	Comparative	Overall
GPT5.1	29.0	49.0	38.0	35.0	69.0	100	100	60.0
Qwen3-235B-A22B	53.0	67.0	56.0	70.0	40.0	99.0	86.0	67.3
Llama3.1-405B	17.0	38.0	70.0	100	95.0	97.0	100	73.9
GPT4.1	48.0	93.0	86.0	98.0	99.0	99.0	96.0	88.0
Deepseek-v3.2	93.0	93.0	98.0	77.0	95.0	95.0	97.0	93.0
Average	48.0	68.0	69.6	76.0	79.6	98.0	96.0	76.0
Llama3.2-3B	17.0	15.0	3.0	16.0	53.0	8.0	91.0	29.0
Gemma3-4B-IT	1.0	0.0	95.0	9.0	23.0	4.0	81.0	30.0
Gemma3-27-IT	0.0	0.0	5.0	11.0	33.0	100	100	36.0
Llama3.1-8B	0.0	0.0	0.0	92.0	80.0	74.0	98.0	49.0
Qwen3-8B	0.0	26.0	0.0	44.0	98.0	84.0	98.0	50.0
Llama3.3-70B	1.0	22.0	11.0	49.0	76.0	100	100	51.0
Gemma3-12B-IT	18.0	45.0	18.0	41.0	39.0	99.0	100	51.0
Qwen3-32B	21.0	45.0	38.0	45.0	91.0	100	96.0	62.0
Deepseek-R1-Distill-Llama70B	76.0	85.0	74.0	84.0	60.0	100	100	83.0
Deepseek-R1-Distill-Qwen32B	71.0	80.0	68.0	79.0	59.0	100	100	80.0
Average	21.0	32.0	31.0	47.0	61.0	77.0	96.0	52.0

Table 5: Accuracy of models in logic type prediction.

Logic type	NLI–TAPAS	NLI–TAPEX	TAPAS–TAPEX
Ordinal	0.54	**0.61**	0.38
Majority	0.58	**0.62**	0.37
Count	0.70	**0.73**	0.59
Aggregation	0.61	**0.75**	0.61
Only	0.70	**0.76**	0.51
Comparative	0.74	**0.79**	0.58
Superlative	0.79	**0.82**	0.48
Overall Avg	0.67	**0.73**	0.50

Table 6: Average pairwise disagreement rates across logic types (averaged over models).

tain *disagreement rates*. Table 10 in Appendix F reports the detailed pairwise disagreement results for each logic type, and Table 6 shows the average results. The main takeaways are:

NLI-Acc diverges most from table-based verifiers. Across logic types, NLI-Acc shows substantially higher disagreement with TAPEX-Acc (avg 0.73) and TAPAS-Acc (avg 0.67) than TAPAS-Acc and TAPEX-Acc disagree with each other (avg 0.50). This gap suggests that NLI models operate on a fundamentally different definition of entailment than dedicated table-based verifiers. The significant disagreement between NLI-Acc and the table-centric models indicates that NLI processes logic through a linguistic lens rather than a tabular one. TAPAS and TAPEX show relatively high agreement, likely because they are architecturally optimized for structured data.

Disagreement is operation-dependent. The largest gaps appear for *Superlative* and *Comparative* (NLI-Acc vs. TAPEX-Acc: 0.82/0.79), followed by *Only* and *Aggregation*, while *Majority* and *Ordinal* exhibit the lowest cross-metric disagreement. Therefore, we can conclude that disagreement spikes on logic types that require precise numeric computation, comparison, and selection.

6 Conclusion

This paper presents an operation-aware framework that decomposes LT2T into four distinct competencies. Our evaluation of 700 Logic2Text instances reveals that aggregate metrics are misleading: verifier choice significantly shifts results, and a pervasive meta-logical gap exists where models fail to align realization with execution. With specific bottlenecks identified in ordinals and aggregation, and human studies revealing high verifier error rates, we conclude that operation-level diagnostics are necessary to move beyond surface-level fluency toward verifiable logical reliability.

Limitations

Our study evaluates models under a single prompting configuration (one-shot). While this setting enables controlled comparisons, performance and error patterns may change under alternative prompting strategies, such as using more in-context examples, instruction refinements, or chain-of-thought style reasoning. Exploring these prompting variations is an important direction for future work. Our human study is also constrained (single annotator, limited sample size, and one representative model), which may limit generalizability and prevent reporting robust inter-annotator agreement.

References

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Hanwen Zha, Xiyou Zhou, Yunkai Zhang, Sairam Sundaresan, and William Yang Wang. 2020b. Logic2Text: High-fidelity natural language generation from logical forms. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2096–2111, Online. Association for Computational Linguistics.

Ao Liu, Haoyu Dong, Naoaki Okazaki, Shi Han, and Dongmei Zhang. 2022. PLOG: Table-to-logic pretraining for logical table-to-text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5531–5546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lena Trigg and Dean F. Hougen. 2025. Logical table-to-text generation: Challenges, methods, and reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1663–1677, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao, and Chenghua Lin. 2024. Effective distillation of table-based reasoning ability from LLMs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5538–5550, Torino, Italia. ELRA and ICCL.

Xueliang Zhao, Tingchen Fu, Lemao Liu, Lingpeng Kong, Shuming Shi, and Rui Yan. 2023a. SORTIE: Dependency-aware symbolic reasoning for logical data-to-text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11247–11266, Toronto, Canada. Association for Computational Linguistics.

Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023b. Investigating table-to-text generation capabilities of large language models in real-world information seeking scenarios. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 160–175, Singapore. Association for Computational Linguistics.

A Automatic Evaluation Metrics

Automatic evaluation metrics are calculated using official code releases:

NLI-Acc and SP-Acc: https://github.com/wenhuchen/LogicNLG.

TAPAS-Acc and TAPEX-Acc: https://github.com/microsoft/PLOG.

B Logical Form Execution Accuracy

Table 7 presents the ability of models to execute LFs against tables.

C Human Evaluation Details

To complement automatic verifier-based metrics, we conduct a targeted human evaluation designed to be lightweight yet diagnostically informative. We sample 40 instances per logic type (7 types; 280 total) using three strata: Agreement–Entailed (E/E), Agreement–Refuted (R/R), and Disagreement. Each instance includes the input table, the model-generated claim (Task 2 output), the reference LF, and its natural-language interpretation. A graduate student annotator voluntarily participated and provided informed consent before labeling the samples. The evaluator judged LF-adherence of each instance by answering: *Does the generated claim faithfully realize the provided logical form when grounded in the table?* The annotator selects one of:

- *Adheres (A):* The sentence realizes the logical form completely and correctly, and it is entailed.
- *Not-entailed (N):* The sentence cannot be verified as true from the table.
- *Off-LF but entailed (O):* The sentence is supported by the table, but expresses a different fact than the LF, such as a different operator, column, subset/filter, or missing/extra constraint.
- *Unclear (U):* The sentence is too vague or underspecified to verify.

We introduce the *Verifier False Negative Rate (VFNR)* to quantify how often verifiers jointly reject claims that a human still deems LF-faithful. The value is computed in Equation 1:

$$\mathrm{VFNR} = \frac{A - \mathrm{Ent}_{\mathrm{both}} - \mathrm{Ref}_{\mathrm{TAPAS}} - \mathrm{Ref}_{\mathrm{TAPEX}}}{\mathrm{Ref}_{\mathrm{both}}} \quad (1)$$

where A is the total number of *LF-Adherence and Entailed* approved by the human evaluator, $\mathrm{Ent}_{\mathrm{both}}$ is the total number of cases accepted as entailed by both automatic verifiers, $\mathrm{Ref}_{\mathrm{both}}$ is the total number of cases rejected by both automatic verifiers, $\mathrm{Ref}_{\mathrm{TAPAS}}$ is the total number of cases rejected by TAPAS, and $\mathrm{Ref}_{\mathrm{TAPEX}}$ is the total number of cases rejected by TAPEX.

A high VFNR indicates that dual-metric rejection is overly aggressive, hiding many genuine, human-approved outputs. (IRB was not required.)

D Confusion Matrix Heatmap

As illustrated in Figure 2, the *Comparative* class is consistently over-predicted. A closer inspection of model rationales reveals a systematic error: many models focus on the *final boolean check* in the logical form (LF)—which indeed compares a derived value to the gold reference—while ignoring the *primary* operation used to *produce* that value. For example, consider the instance

> **Sentence generated by the model:** "The average race rank for is approximately 12.56."
> **Logic string:** `round_eq { avg { all_rows ; rank } ; 12.56 } = true`

The correct logic type is *Aggregation* because the central action is computing an *average* over all rows. However, several models label it as *Comparative*, explaining that "the logic string checks if the average of the *rank* column is approximately equal to 12.56." In effect, the model attends to the `round_eq` *comparison* at the end of the LF rather than the `avg` operation that dominates the reasoning chain.

This *last-step bias* appears across diverse architectures and accounts for a substantial fraction of false *Comparative* predictions. Accordingly, we believe that the over-prediction of *Comparative* is *not* an inherent weakness of current LLMs, but a resolvable prompting artifact, and this motivates future work on prompt engineering and rationale supervision for logic-type classification.

E Results of Task 4

Tables 8 and 9 show the result of Task 4.

F Detailed Pairwise Disagreement Matrix

We discuss results of Table 10 in the following:

Model	Aggregation	Count	Ordinal	Comparative	Only	Superlative	Majority	Average
GPT5.1	71.0	84.0	67.0	84.0	83.0	85.0	91.0	80.7
DeepSeek-V3.2	92.0	87.0	88.0	91.0	94.0	92.0	92.0	90.9
Qwen3-235B-A22B	88.0	87.0	**95.0**	**98.0**	92.0	92.0	94.0	92.3
Llama3.1-405B	82.0	**92.0**	91.0	**98.0**	**97.0**	**98.0**	**100**	**94.0**
GPT4.1	**94.0**	90.0	87.0	97.0	**97.0**	97.0	96.0	**94.0**
Average (Frontier)	85.4	88.0	85.6	93.6	92.6	92.8	94.6	90.4
Gemma3-12B-IT	34.0	78.0	81.0	66.0	81.0	81.0	81.0	71.7
DeepSeek-R1-Distill-Qwen32B	81.0	69.0	74.0	81.0	74.0	72.0	78.0	75.6
DeepSeek-R1-Distill-Llama70B	85.0	66.0	77.0	83.0	85.0	80.0	87.0	80.4
Llama3.3-70B	66.0	74.0	78.0	94.0	93.0	91.0	98.0	84.9
Gemma3-27B-IT	82.0	75.0	80.0	93.0	93.0	84.0	**99.0**	86.6
Llama3.2-3B	93.0	**97.0**	93.0	69.0	80.0	**99.0**	90.0	88.7
Qwen3-8B	77.0	87.0	93.0	96.0	92.0	91.0	93.0	89.9
Llama3.1-8B	74.0	91.0	**99.0**	91.0	95.0	97.0	97.0	92.0
Qwen3-32B	90.0	94.0	96.0	97.0	92.0	98.0	97.0	94.9
Gemma3-4B-IT	**100**	94.0	92.0	**98.0**	**99.0**	**99.0**	96.0	**96.9**
Avg (Small/Distilled)	78.2	82.5	86.3	86.8	88.4	89.2	91.6	86.1

Table 7: The results of Task 1 measuring the models' accuracies to execute logical forms.

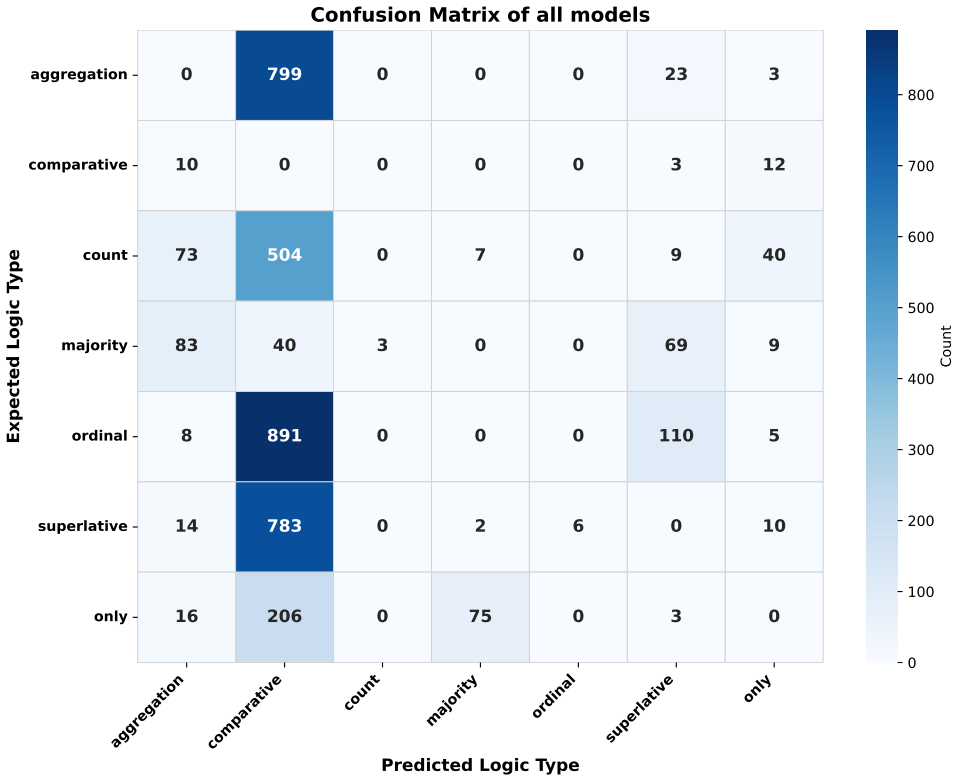

Figure 2: Confusion matrix heatmap of all models.

Model-specific trends. Looking at individual models, we can see which ones cause the most confusion among the evaluators:

Most Contentious: Models like Llama3.3-70B and GPT5.1 often trigger the highest disagreement rates, reaching up to 0.97 in *Superlative* logic for Llama3.3-70B. This may imply these models produce *edge case* answers that different evaluators interpret in opposite ways.

Most Consistent: Smaller or older models like Llama3.1-8B generally show lower disagreement rates across the board compared to the frontier models.

Aggregation anomaly. The data highlights that *Aggregation* is a specific failure point for table-based reasoning. While TAPAS-Acc and TAPEX-Acc generally align (0.50 avg), their disagreement jumps to 0.61 for aggregation tasks. This suggests that even table-specialized verifiers handle aggregation differently, yielding higher disagreement.

Model	Aggregation	Comparative	Count	Majority	Ordinal	Superlative	Only	Average
NLI-Acc								
Deepseek-v3.2	68.0	82.0	64.0	68.0	56.0	88.0	79.0	72.0
Llama3.1-405B	57.0	81.0	82.0	64.0	81.0	91.0	70.0	75.0
GPT5.1	61.0	86.0	81.0	66.0	73.0	85.0	85.0	77.0
Qwen3-235B-A22B	72.0	83.0	79.0	67.0	76.0	89.0	83.0	78.0
GPT4.1	71.0	88.0	85.0	68.0	68.0	95.0	79.0	79.0
Average (Frontier)	65.8	84.0	78.2	66.6	70.8	89.6	79.2	76.2
Llama3.2-3B	72.0	66.0	60.0	73.0	60.0	73.0	60.0	66.0
Gemma3-27B-IT	47.0	74.0	65.0	57.0	77.0	71.0	70.0	66.0
Llama3.1-8B	47.0	80.0	71.0	68.0	58.0	82.0	65.0	67.0
Deepseek-R1-Distill-Qwen32B	38.0	73.0	60.0	73.0	65.0	86.0	84.0	68.0
Deepseek-R1-Distill-Llama70B	43.0	78.0	66.0	67.0	74.0	82.0	77.0	69.0
Gemma3-12B-IT	65.0	75.0	72.0	69.0	68.0	82.0	79.0	73.0
Qwen3-8B	65.0	75.0	72.0	79.0	39.0	91.0	91.0	73.0
Gemma3-4B-IT	68.0	73.0	80.0	68.0	77.0	79.0	78.0	75.0
Qwen3-32B	70.0	79.0	71.0	78.0	53.0	88.0	83.0	75.0
Llama3.3-70B	59.0	79.0	82.0	70.0	85.0	89.0	67.0	76.0
Average (Small/Distilled)	57.4	75.2	69.9	70.2	65.6	82.3	75.4	70.8
TAPAS-Acc								
Deepseek-v3.2	83.0	83.0	58.0	43.0	49.0	75.0	67.0	66.0
GPT5.1	67.0	88.0	66.0	37.0	66.0	79.0	77.0	69.0
Llama3.1-405B	83.0	85.0	63.0	45.0	71.0	82.0	53.0	69.0
Qwen3-235B-A22B	82.0	82.0	61.0	55.0	72.0	77.0	70.0	71.0
GPT4.1	79.0	85.0	66.0	51.0	67.0	82.0	72.0	72.0
Average (Frontier)	78.8	84.6	62.8	46.2	65.0	79.0	67.8	69.4
Llama3.1-8B	44.0	75.0	52.0	29.0	47.0	48.0	51.0	49.0
Llama3.2-3B	52.0	54.0	53.0	48.0	36.0	63.0	58.0	52.0
Deepseek-R1-Distill-Llama70B	72.0	70.0	44.0	44.0	51.0	53.0	64.0	57.0
Deepseek-R1-Distill-Qwen32B	68.0	74.0	54.0	45.0	45.0	52.0	66.0	58.0
Gemma3-27B-IT	61.0	76.0	62.0	52.0	57.0	63.0	54.0	61.0
Gemma3-12B-IT	79.0	72.0	61.0	52.0	58.0	55.0	55.0	62.0
Gemma3-4B-IT	71.0	65.0	67.0	50.0	63.0	62.0	67.0	64.0
Llama3.3-70B	82.0	73.0	69.0	47.0	68.0	75.0	58.0	67.0
Qwen3-8B	87.0	81.0	65.0	60.0	37.0	76.0	80.0	69.0
Qwen3-32B	87.0	75.0	59.0	60.0	52.0	73.0	81.0	70.0
Average (Small/Distilled)	70.3	71.5	58.6	48.7	51.4	62.0	63.4	60.9
TAPEX-Acc								
Deepseek-v3.2	90.0	79.0	66.0	49.0	54.0	80.0	66.0	69.0
Llama3.1-405B	89.0	85.0	83.0	49.0	76.0	79.0	44.0	72.0
GPT5.1	69.0	87.0	81.0	49.0	74.0	80.0	81.0	74.0
GPT4.1	82.0	89.0	84.0	44.0	76.0	83.0	65.0	75.0
Qwen3-235B-A22B	93.0	79.0	75.0	61.0	70.0	83.0	62.0	75.0
Average (Frontier)	84.6	83.8	77.8	50.4	70.0	81.0	63.6	73.0
Llama3.1-8B	39.0	81.0	50.0	29.0	50.0	51.0	44.0	49.0
Llama3.2-3B	61.0	51.0	59.0	51.0	41.0	64.0	46.0	53.0
Deepseek-R1-Distill-Llama70B	91.0	64.0	60.0	41.0	58.0	62.0	51.0	61.0
Deepseek-R1-Distill-Qwen32B	95.0	67.0	65.0	40.0	49.0	58.0	53.0	61.0
Gemma3-4B-IT	79.0	61.0	64.0	54.0	62.0	66.0	68.0	65.0
Gemma3-27B-IT	77.0	77.0	70.0	54.0	67.0	73.0	50.0	67.0
Gemma3-12B-IT	90.0	74.0	73.0	56.0	65.0	66.0	53.0	68.0
Llama3.3-70B	87.0	74.0	74.0	46.0	79.0	73.0	54.0	70.0
Qwen3-8B	92.0	78.0	68.0	62.0	55.0	77.0	74.0	72.0
Qwen3-32B	91.0	68.0	69.0	70.0	56.0	80.0	85.0	74.0
Average (Small/Distilled)	80.2	69.5	65.2	50.3	58.2	67.0	57.8	64.0

Table 8: Task 4: automated logical-fidelity evaluation on Logic2Text *without* LFs (higher is better).

Model	Aggregation	Comparative	Count	Majority	Ordinal	Superlative	Only	Average
SP-Acc								
Llama3.1-405B	16.0	41.0	53.0	67.0	65.0	59.0	57.0	51.0
Deepseek-v3.2	20.0	51.0	61.0	70.0	60.0	64.0	61.0	55.0
GPT4.1	22.0	52.0	52.0	72.0	74.0	64.0	75.0	59.0
Qwen-3-235B	25.0	45.0	60.0	73.0	71.0	72.0	73.0	60.0
GPT5.1	36.0	54.0	51.0	76.0	71.0	58.0	75.0	60.0
Average (Frontier)	23.8	48.6	55.4	71.6	68.2	63.4	68.2	57.0
Llama3.2-3B	33.0	40.0	44.0	58.0	48.0	54.0	46.0	46.0
Gemma3-12B-IT	22.0	35.0	63.0	71.0	58.0	39.0	61.0	50.0
Gemma3-27B-IT	28.0	44.0	49.0	69.0	62.0	49.0	53.0	51.0
Deepseek-R1-Distill-Llama70B	15.0	44.0	47.0	69.0	66.0	55.0	65.0	51.0
Llama3.3-70B	20.0	43.0	56.0	76.0	68.0	49.0	55.0	52.0
Deepseek-R1-Distill-Qwen32B	11.0	53.0	50.0	66.0	64.0	49.0	76.0	53.0
Gemma3-4B-IT	32.0	50.0	51.0	56.0	70.0	56.0	56.0	53.0
Llama3.1-8B	57.0	46.0	36.0	62.0	53.0	52.0	64.0	53.0
Qwen-3-8B	23.0	51.0	43.0	64.0	65.0	54.0	68.0	53.0
Qwen-3-32B	29.0	48.0	52.0	69.0	62.0	66.0	76.0	57.0
Average (Small/Distilled)	27.0	45.4	48.9	66.0	61.6	52.3	62.0	51.9

Table 9: Task 4: SP-Acc on Logic2Text *without* provided LFs.

Model	Ordinal	Majority	Aggregation	Count	Only	Comparative	Superlative	Average
Pairwise disagreement rate between NLI-Acc and TAPEX-Acc								
Gemma3-27B-IT	0.60	0.49	0.59	0.65	0.75	0.80	0.76	0.66
Llama3.1-8B	0.56	0.67	0.47	0.75	0.68	0.78	0.88	0.68
Llama3.2-3B	0.66	0.71	0.81	0.63	0.59	0.66	0.76	0.69
Deepseek-R1-Distill-Qwen32B	0.53	0.69	0.92	0.59	0.83	0.68	0.71	0.71
Deepseek-R1-Distill-Llama70B	0.67	0.62	0.88	0.66	0.74	0.78	0.65	0.71
Gemma3-12B-IT	0.57	0.61	0.82	0.72	0.83	0.72	0.73	0.71
GPT4.1	0.56	0.58	0.62	0.76	0.81	0.79	0.84	0.71
Qwen-3-32B	0.51	0.69	0.88	0.71	0.75	0.74	0.72	0.71
Deepseek-v3.2	0.39	0.62	0.83	0.69	0.78	0.82	0.90	0.72
Qwen-3-235B	0.54	0.43	0.79	0.82	0.75	0.85	0.83	0.72
Gemma3-4B-IT	0.78	0.65	0.69	0.72	0.74	0.69	0.88	0.74
GPT5.1	0.71	0.69	0.51	0.77	0.90	0.96	0.91	0.78
Llama3.3-70B	0.94	0.55	0.71	0.84	0.75	0.82	0.97	0.80
Llama3.1-405B	0.74	0.57	0.88	0.79	0.66	0.94	0.93	0.79
Qwen-3-8B	0.42	0.77	0.87	0.79	0.87	0.84	0.77	0.76
Average	0.61	0.62	0.75	0.73	0.76	0.79	0.82	0.73
Pairwise disagreement rate between NLI-Acc and TAPAS-Acc								
Gemma3-27B-IT	0.60	0.43	0.41	0.60	0.73	0.81	0.71	0.61
Llama3.2-3B	0.57	0.70	0.60	0.68	0.50	0.67	0.72	0.63
Deepseek-R1-Distill-Qwen32B	0.53	0.63	0.63	0.49	0.81	0.66	0.75	0.64
Llama3.1-8B	0.52	0.64	0.40	0.72	0.71	0.64	0.83	0.64
Deepseek-v3.2	0.35	0.52	0.68	0.67	0.71	0.70	0.88	0.64
Qwen-3-32B	0.33	0.60	0.70	0.72	0.74	0.76	0.66	0.64
Qwen-3-235B	0.47	0.47	0.65	0.67	0.66	0.75	0.83	0.64
Deepseek-R1-Distill-Llama70B	0.62	0.55	0.66	0.63	0.73	0.75	0.62	0.65
Qwen-3-8B	0.26	0.67	0.70	0.70	0.83	0.71	0.78	0.66
GPT4.1	0.59	0.50	0.61	0.83	0.60	0.88	0.79	0.69
GPT5.1	0.51	0.61	0.50	0.80	0.81	0.87	0.76	0.69
Llama3.1-405B	0.50	0.53	0.72	0.78	0.57	0.79	0.92	0.69
Gemma3-4B-IT	0.77	0.67	0.66	0.77	0.74	0.75	0.79	0.73
Gemma3-12B-IT	0.65	0.61	0.67	0.80	0.80	0.64	0.79	0.71
Llama3.3-70B	0.82	0.59	0.60	0.71	0.56	0.77	0.97	0.72
Average	0.54	0.58	0.61	0.70	0.70	0.74	0.79	0.67
Pairwise disagreement rate between TAPAS-Acc and TAPEX-Acc								
Llama3.1-8B	0.34	0.21	0.31	0.47	0.36	0.48	0.42	0.37
Deepseek-R1-Distill-Llama70B	0.26	0.25	0.78	0.46	0.56	0.42	0.31	0.43
Gemma3-4B-IT	0.30	0.33	0.45	0.60	0.50	0.40	0.44	0.43
Deepseek-R1-Distill-Qwen32B	0.23	0.35	0.84	0.51	0.49	0.50	0.26	0.45
Llama3.2-3B	0.33	0.43	0.62	0.40	0.40	0.45	0.51	0.45
Gemma3-27B-IT	0.42	0.38	0.52	0.52	0.34	0.53	0.51	0.46
Llama3.3-70B	0.53	0.30	0.52	0.61	0.56	0.49	0.56	0.51
Llama3.1-405B	0.44	0.42	0.78	0.62	0.34	0.57	0.44	0.52
Gemma3-12B-IT	0.46	0.49	0.65	0.60	0.47	0.54	0.45	0.52
Deepseek-v3.2	0.25	0.40	0.65	0.60	0.57	0.64	0.64	0.54
Qwen-3-235B	0.39	0.35	0.75	0.67	0.55	0.70	0.40	0.54
GPT4.1	0.42	0.33	0.50	0.78	0.63	0.70	0.60	0.57
GPT5.1	0.50	0.37	0.44	0.71	0.69	0.86	0.59	0.56
Qwen-3-8B	0.34	0.45	0.60	0.62	0.58	0.72	0.58	0.56
Qwen-3-32B	0.49	0.52	0.76	0.65	0.63	0.64	0.43	0.59
Average	0.38	0.37	0.61	0.59	0.51	0.58	0.48	0.50

Table 10: Pairwise disagreement across evaluator pairs (NLI–TAPAS, NLI–TAPEX, TAPAS–TAPEX).