# ErrorBench: Fine-Grained Error Analysis of Multi-Family LLMs in Data-to-Text Generation

Soumya Bharadwaj
*Department of Computer Science and Engineering*
*Indian Institute of Technology Guwahati, India*
soumya.bharadwaj@iitg.ac.in

Ashish Anand
*Department of Computer Science and Engineering*
*Indian Institute of Technology Guwahati, India*
anand.ashish@iitg.ac.in

***Abstract*—Large language models (LLMs) increasingly power Data-to-Text (D2T) generation, yet their reliability varies widely across architectures, training regimes, and parameter scales. In this work, we present the first cross-family, multi-scale analysis of fine-grained generation errors in D2T outputs. We rigorously evaluate 27 variants from nine major LLM families (Meta, OpenAI, Google, Alibaba, DeepSeek, Mistral AI, Microsoft, Technology Innovation Institute (TII), and SmolLM) using triple-to-text setting with structured knowledge derived from DBPedia. We manually annotate model outputs with a 10-category span-level taxonomy, enabling rich error profiling beyond standard semantic faithfulness metrics, resulting in a reusable span-annotated error dataset. Our findings reveal that while model scale generally improves quality, this trend is highly non-uniform; certain intermediate-scale models exhibit catastrophic failures due to Prompt Echo. We identify Addition and Omission as the dominant global errors, and show that families like Mistral and Qwen-3 are the most reliable architectures for structured D2T. Complementing these findings, prompt ablation studies demonstrate that stricter instructional constraints can reduce specific errors in capable models but are ineffective against persistent failures in lower-capacity architectures, highlighting the primacy of model capacity over prompt design. Overall, this study quantifies distinctive error fingerprints across model lineages, offering actionable guidance for building more dependable structured-data generation systems. The dataset is publicly available at: https://huggingface.co/datasets/soumyaBharadwaj/ErrorBench.**

***Index Terms*—Data-To-Text Generation, Semantic Errors, LLM Families**

## I. Introduction

Data-to-text generation has been extensively studied across rule-based, template-based, and neural generative models, with an increasing shift toward large language models (LLMs) as general-purpose generators [1]. Prior evaluations [2] have primarily focused on semantic adequacy, factuality, and overall fluency, often using human judgments or n-gram based metrics. However, existing research emphasizes that system outputs contain diverse error types (e.g., semantic [3], [4], syntactic, structural [5], pragmatic) and that narrowly focused evaluation underestimates the true failure modes of data-driven Natural Language Generation (NLG) systems.

The recent study [4], conduct a fine-grained semantic error analysis across 15 data-to-text systems using WebNLG. Their work identifies and annotates three categories of semantic deviations: omission, addition, and repetition, applying a span-based annotation scheme to both rule-based and neural systems, including two untuned LLMs. Their findings highlight systematic differences in semantic robustness across system families and offer valuable insights into content selection and over-generation behaviors.


Fig. 1: Example of span-level error annotation showing a Llama2-7B output with multiple simultaneous errors: Partial Entity Mismatch, Relation Ambiguity, Addition, and Spell/-Format errors, alongside the grounded input tuple.

Despite its significance, this prior work has three notable limitations. First, the error taxonomy is restricted to semantic deviations, excluding other common and practically consequential error types such as formatting mistakes, syntactic breakdowns, incoherent sentences, orthographic errors, or structural mismatches and they further limit analysis to omission errors despite data-to-text generation requiring equal fidelity to both entities and relations. Second, the evaluation covers only a limited set of LLMs, and does not reflect the capabilities or failure modes of modern state-of-the-art model families such as Qwen-3 [6], DeepSeek-R1 [7], Gemma-3 [8], or Phi-3 [9]. Third, the study is tied to a single dataset and domain (WebNLG [10]), limiting generalizability and the ability to evaluate robustness across diverse inputs or formatting demands, especially the fine-grained relational complexity inherent in sources like DBPedia.

We address these limitations by presenting the first cross-family, multi-scale error analysis. Using rigorous span-level annotation technique, we offer a high-resolution view of LLM failures. As shown in Fig 1, this method precisely marks multiple concurrent errors (semantic, ambiguity, format) in generated outputs. Our contributions are as follows:

- **Fine-Grained Span-Level Error Taxonomy and Quantitative Metrics.** We employ a 10-category, span-level error framework that extends prior semantic analyses by explicitly bifurcating omission into *Entity Omission*

and *Relation Omission*. We further introduce the *Total Error Span Rate (TESR)* and *Generation Quality Index (GQI)* metrics. This framework captures detailed error signatures (semantic, syntactic, coherence, formatting) that remain invisible under semantic-only evaluation.

- **Dissection of Scaling Anomalies and Architectural Reliability.** We analyze 9 LLM families with three scale variants each on 224 sentences covering 224 unique relations, enabling a broad evaluation of diverse relation semantics, and show that scaling behavior in D2T is highly non-uniform.
- **Reusable Span-Annotated Error Dataset.** Our annotation effort yields a human-annotated error dataset (≈2.5k annotated sentences) supporting future benchmarking, automatic error detection, and studies of robustness across diverse DBPedia relation and entity types.
- **Prompt Ablation as a Diagnostic for Architectural Fidelity Limits.** We conduct a targeted prompt ablation across three prompt configurations to systematically evaluate the effect of instructional constraints on error behavior in models exhibiting high error rates.

Our studies show that fine-grained error behavior in D2T generation varies substantially across model families and scales, with several dominant failure modes persisting despite improvements in overall quality. To support reproducibility and enable future research on error-aware evaluation, we make our dataset publicly available.[1]

## II. Related Work

**Semantic Error Evaluation in Data-to-Text Generation** Early NLG evaluation work [1], [5] focused on sentence level metrics or human Likert ratings, which often fail to capture fine-grained semantic inconsistencies. More recent studies [3], [4] emphasize word-span or entity-level annotation of errors. They analyze semantic errors across structured datasets, finding symbolic models more faithful and LLMs prone to overgeneration, but consider only limited LLMs and no scale or family variation.

**Hallucinations, Omissions, and Relationship Errors in LLMs** Hallucination and factual inconsistency in LLMs are widely studied [11], especially in open-ended QA and summarization. However, D2T hallucinations differ because the input is fully structured any deviation can be precisely identified. Recent studies [12] analyse D2T hallucinations using alignment scoring between inputs and generated text. Although informative, these works do not perform detailed manual error annotation nor study multi-model LLM behaviour.

**Impact of Model Scaling and Architecture** LLM scaling laws have demonstrated consistent improvements in fluency and reasoning with model size [13], but recent evidence [14] suggests that factual grounding does not always improve proportionally. Existing comparative [4] analyses typically cover two or three models rather than full families. To the best of our knowledge, no prior work has conducted a large-scale, multi-family, multi-parameter evaluation of LLMs in the Data-to-Text setting; our work addresses this gap by presenting a systematic and fine-grained analysis of semantic errors across multiple LLM families.

[1]https://huggingface.co/datasets/soumyaBharadwaj/ErrorBench

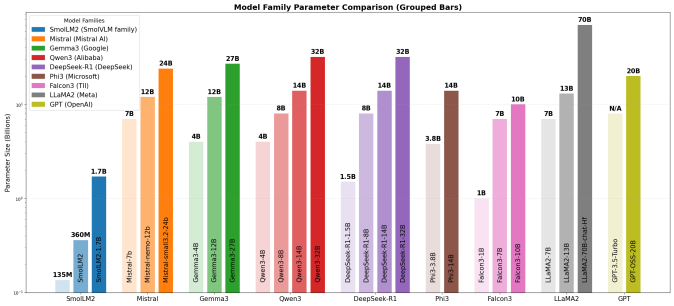



Fig. 2: Overview of model families and parameter scales considered in our comparative error analysis. For each family, colors transition from light to dark, representing lower to higher parameter models

## III. Methodology

Our objective is to evaluate the faithfulness and linguistic quality of D2T generation from structured knowledge triples produced by diverse LLMs. We use structured triples from the DBPedia [15] knowledge base, whose rich relational variety and entity heterogeneity enable fine-grained semantic error analysis in a controlled generation setting.

### A. Data and Task

We construct our input tuples from a subset of the DBPedia-based dataset introduced in [16]. From this resource, we select $N$ = 224 samples such that each sample represents a unique relation type, ensuring broad relational coverage and diversity. This design allows us to systematically examine how different LLM families behave across heterogeneous relation semantics in the D2T setting, rather than overfitting to a narrow or repetitive relational distribution.

*1) Task Definition:* The D2T task converts the structured input into a single fluent and faithful natural language sentence. Each tuple is structured as $(E1, E1_{TYPE}, R, E2, E2_{TYPE})$, where $E1$ and $E2$ are entities, $E1_{TYPE}$, $E2_{TYPE}$ are entity types restricted to set $T$={Person, Location, Organization}, and $R$ is the specified relation between $E1$ and $E2$. The goal is to generate an output sentence that accurately verbalizes the $R$ relationship between $E1$ and $E2$, while strictly adhering to the entities and types provided in the input tuple.

### B. LLM Variants and Scaling

In our work, we systematically investigate the impact of architectural family and parameter scale on D2T performance. Our evaluation set, denoted as $\mathcal{M}$, comprises 27 distinct LLM variants from 9 major architectural families (e.g., Llama-2, Mistral, Gemma, GPT, etc.) as shown in Fig 2. For analysis, we refer to any model variant in this set as $M_i$, where $i = 1 \cdots 27$, and $M_i \in \mathcal{M}$. The models were chosen to cover three distinct scaling tiers within each family where available: (i)

*Low Scale (L)* Typically $<$ 8 billion parameters. (ii) *Medium Scale (M)* Typically 8$-$14 billion parameters. (iii) *High Scale (H)* Typically $\geq$ 24 billion parameters.

The following models and their versions are used: *SmolLM2 (SmolVLM Family)*, SmolLM2-135M, SmolLM2-360M, SmolLM2-1.7B [17]. *Mistral (Mistral AI)* [18], Mistral-7b, Mistral-nemo-12b, Mistral-small3.2-24b. *Gemma 3 (Google)* [8], Gemma3-4B, Gemma3-12B, Gemma3-27B. *Qwen 3 (Alibaba)* [6], Qwen3-4B, Qwen3-8B, Qwen3-14B and Qwen3-32B. *DeepSeek-R1 (DeepSeek)* [7], DeepSeek-R1-1.5B, DeepSeek-R1-8B, DeepSeek-R1-14B and DeepSeek-R1-32B. *Phi-3 (Microsoft)* [9], Phi-3-3.8B, Phi-3-14B. *Falcon 3 (Technology Innovation Institute (TII) in Abu Dhabi)* [19], Falcon3-1B, Falcon3-7B, Falcon3-10B. *LLaMA-2 (Meta)* [20], LLaMA-2-7B, LLaMA-2-13B, LLaMA-2-chat-hf-70B. *GPT Family (OpenAI)*, GPT-3.5-Turbo [21], GPT-OSS-20B [22]. All the models are accessed via Ollama API.[2]

### C. Data-to-Text Generation

Using the structured triples defined in Section III-A1, we generated one sentence per tuple with each of the model variants $M_i$ described in Section III-B. This resulted in a total corpus of $N \times |\mathcal{M}|$ = 224×27 = 6,048 generated sentences. All models were evaluated under identical conditions using a fixed instruction, referred to as the Standard Guidelines (SG) prompt, to generate a single, fluent sentence from the provided tuple. This control ensures observed performance differences are due to architecture and scale, not prompt engineering variations which is further studied in Section IV-E. We adopt a one-shot prompting strategy rather than zero-shot prompting, as prior work shows that minimal in-context examples improve stability and reliability in table-to-text and data-to-text generation tasks [23]. The SG prompt template is as follows.

**Generation Prompt**

Given a tuple of the form (Entity1, Entity1 Type, Relation between the entities, Entity2, Entity2 Type), generate a sentence using the information given in the tuple.

Example:
**Tuple:** *("San Giovanni Rotondo","Location","saint","John the Baptist","Person")*
**Sentence:** *John the Baptist was a saint who lived in the small village of San Giovanni Rotondo.*

Now, for the given tuple, generate a single complex, diverse sentence in a similar manner.

**Guidelines:** 1. Do not include entity types in the generated sentence.
2. Do not give explanations or reasoning.
3. The sentence should be syntactically different from the example.
4. Use synonyms of the relation rather than repeating the same wording.
5. Keep the sentence complex and natural, ensuring the relation is still clear but not overly obvious.
6. If entity details are provided in parentheses, use them to enrich the sentence.
Given tuple: sample_tuple

Sentence:

### D. Fine Grained Error Categories

To assess the linguistic quality and faithfulness of the generated sentences, we developed a comprehensive, manual error taxonomy. Traditional surface-level metrics (e.g., BLEU, ROUGE) are known to fail at capturing semantic or logical incoherence in Data-to-Text tasks. Our taxonomy classifies generation errors into 10 distinct, fine-grained categories:

*1) Entity Omission (E1 / E2):* The required entity specified in the input tuple does not appear anywhere in the generated sentence. This includes cases where the entity is entirely dropped or replaced with unrelated content.

*2) Relation Omission:* The sentence fails to express the semantic relation provided in the tuple. Although the entities might be present, the relational meaning is missing, under-specified, or not conveyed at all.

*3) Repetition:* Unnecessary repetition of words, phrases, or entities within the sentence, typically caused by decoding loops or over-generation. This includes consecutive repeated tokens ("is is") as well as more distributed redundancies.

*4) Addition:* The sentence introduces factual content that is not present in the input tuple and is not inferable from it [4]. This includes hallucinated details such as dates, numbers, extra properties, expanded descriptions, or new entities introduced without grounding.

*5) Spelling/Format Drift (Drift):* Surface-level text quality issues including typos, malformed tokens, irregular punctuation, broken words, leftover artifacts from prompt formatting, or unnatural token boundaries. These issues do not necessarily change meaning but degrade linguistic fidelity.

*6) Prompt Echo:* Fragments of the original instruction, system prompt, or the model's internal reasoning (e.g., step-by-step thinking) appear verbatim in the output. This indicates prompt leakage or incomplete grounding. Such leakage may appear mid-sentence or towards the end, suggesting failures in generation control and clean termination.

*7) Relation Ambiguity:* The relation is technically expressed but in a vague, underspecified, or semantically ambiguous way that does not clearly reflect the intended tuple meaning. The expression of the relation becomes unclear, misleading, or pragmatically incoherent.

*8) Entity Type Change:* The entity appears in the generated sentence but is described with an incorrect semantic type compared to the tuple (e.g., a location described as a person, or an organization treated as a place). This captures mismatches in ontological role or category.

*9) Incoherence:* This represents a severe class of generation failure, capturing a fundamental breakdown in the sentence's meaning and logical flow. The sentence, despite containing required entities and relations, is globally nonsensical, contradictory to the input, or semantically meaningless. It renders the overall utterance unusable due to critical failures in making no sense or factual adherence.

*10) Partial Entity Mismatch:* The entity appears in the sentence but only partially e.g., ("New York" instead of "New York City") or shortened/altered forms that distort the intended referent. This error captures incomplete realizations of the target entity without full omission.

Table I illustrates the practical application of our taxonomy by presenting representative examples for each error category.

### E. Annotation Guidelines

The 6,048 generated sentences underwent rigorous manual, span-level error annotation using the BRAT rapid annotation

[2] https://oLlama.com/

TABLE I: Overview of 10 fine-grained error categories with examples from nine LLM families.

| Type of Error | Family | Tuple and Sentence |
|---|---|---|
| Entity Omission | Mistral | **(James River Gorge, Location, lowestPlace, *James River*, Location)**<br>The breathtaking James River Gorge is where the waterway plunges into its deepest point, carving a dramatic fissure amidst the verdant landscape. |
| Relation Omission | SmolLM | **(Enver Čolaković, Person, restingPlace, Zagreb, Location)**<br>According to a lesser-known historical significance that Enver Čolaković has faced today, our city, Zagreb, stands as an exemplary example of resilience and the enduring spirit of its people. |
| Partial Entity Match | Falcon | **(*Mark Battles Fly America 1*, Organization, distributingCompany, Universal Music Group, Organization)**<br>Mark Battles finds its distributing company in the renowned Universal Music Group. |
| Prompt Echo | LLaMA | **(Sam Nujoma, Person, placeOfBirth, Namibia, Location)**<br>Sam Nujoma's birthplace is Namibia, where his life journey began. Here's how we break down sentence... |
| Addition | Gemma | **(Navika Kotia, Person, stateOfOrigin, India, Location)**<br>Born in Mumbai, Maharashtra, in 1990, Navika Kotia's heritage is deeply rooted in India. |
| Entity Type Mismatch | DeepSeek | **Roman Polom, Person, team, Czech Republic national under-18 football team, Organization)**<br>Roman Polom was a team that began playing football and became the Czech Republic under-18 national football team. Santito led them toward victory in their regional tournament. |
| Spelling/Format Drift | Qwen | **(Wellingsbüttel, Location, administrativeDistrict, Wandsbek, Location)**<br>Wellingsbüttel, an administrative district located within Wandsbek, 是个显著的地区坐标。 |
| Repetition | Phi | **(Harland Svare, Person, highSchool, North Kitsap School District, Organization)**<br>Born and greatly influenced by his early education at North Kitsap High School, Harland Svare later made a notable impact as part of the esteemed institution known as the North Kitsap School District. |
| Relation Ambiguity | GPT | **(National Archives of Sweden, Organization, headquarter, Sweden, Location)**<br>Situated in Sweden, the National Archives of Sweden serve as country's principal custodian of records. |
| Incoherence | DeepSeek | **(Eorodale, Location, lieutenancyArea, Outer Hebrides, Location)**<br>Eorodale moved via lieutenancyArea through Relation in Outer Hebbles. |

tool [24]. To ensure strict internal consistency and taxonomic precision, the annotation was performed by a Single Expert Annotator holding a Master's degree, following a blind annotation protocol in which the annotator had no knowledge of the generating LLM family or parameter scale. Unlike crowdsourced paradigms, which may struggle with subtle distinctions in fine-grained taxonomies (e.g., *Relation Ambiguity* vs. *Incoherence*), an expert-driven approach was chosen to ensure consistent interpretation and uniform guideline application. For each instance, the annotator was presented with the LLM-generated sentence alongside the original DBPedia input tuple. The annotation procedure followed a three-step workflow:

1) **Span Identification**: The annotator highlighted the *minimal* erroneous span within the sentence that contained a discrepancy.
2) **Error Categorization:** Identified spans were assigned to one of the 10 predefined error categories (Section III-D)
3) **Omission Marking:** Cases where the LLM failed to verbalize an entity or relationship from the tuple were explicitly marked using placeholder tokens: `[MISSING_E1]`, `[MISSING_E2]`, and `[MISSING_RELATION]`.

### F. Evaluation Metrics

The performance of each LLM variant $M_i \in \mathcal{M}$ is evaluated using two metrics motivated by the fine-grained, span-level annotations described in Section III-E. We use $N = 224$ defined in Section III-A to denote the total number of sentences generated per model variant.

*1) Total Error Span Rate (TESR):* The TESR quantifies the density of generation failures, defined as the average number of annotated error spans produced by a model per generated sentence. The Total Error Spans is the raw count of all marked spans across the 10 categories (Section III-D). Lower values indicate superior performance in minimizing localized errors.

$$\text{TESR}_{M_i} = \frac{\text{Total Error Spans Annotated for } M_i}{N}$$

*2) Generation Quality Index (GQI):* The GQI measures the overall success rate of a model, it is defined as the percentage of error-free sentences produced. A sentence is considered *error-free* if it contains zero annotated error spans. This metric reflects the model's ability to generate a completely faithful and fluent output adhering to all constraints. Higher values indicate higher generation quality.

$$\text{GQI}_{M_i} = \frac{\text{Error-Free Sentences Generated by } M_i}{N} \times 100\%$$

Notably, TESR and GQI capture complementary characteristics of generation quality, with TESR reflecting the density of localized errors and GQI measuring the model's ability to produce entirely error-free outputs.

## IV. Results And Observations

We present the empirical findings from our error analysis of 27 LLM variants. Results are reported using the GQI and TESR, covering overall architectural performance, model scaling effects, and fine-grained error distribution.

### A. Overall Performance and the Impact of Scaling

Table II showcases the raw counts, GQI, and TESR for all models. The data overwhelmingly supports a strong positive correlation between model scale and D2T generation quality, although the trend is highly non-linear and family-dependent. As shown in Fig 3, the overall trend demonstrates that medium

TABLE II: Consolidated Performance and Error Profile of 27 LLM Variants for D2T Generation using SG III-C. The evaluation covers 6,048 generated outputs, identifying total of **4,732 error spans** distributed across **2,557 error-containing sentences**.

| Model Family | Model | Entity Omission | Relation Omission | Addition | Repetition | Drift | Prompt Echo | Relation Ambiguity | Entity Type Changed | Semantic Incoherence | Partial Entity | #Total Errors | TESR ↓ | #Total Error-Free | GQI (%) ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Gemma3 (Google) | Gemma3-4b **(L)** | 1 | 5 | 85 | 0 | 8 | 2 | 13 | 1 | 2 | 0 | 117 | 0.522 | 131 | 58.48 |
| | Gemma3-12b **(M)** | 0 | 2 | 26 | 0 | 0 | 0 | 26 | 1 | 0 | 2 | 57 | 0.254 | 171 | 76.34 |
| | Gemma3-27b **(H)** | 0 | 3 | 66 | 0 | 0 | 0 | 11 | 0 | 0 | 3 | 83 | 0.371 | 157 | 70.09 |
| Deepseek-r1 (DeepSeek) | Deepseek-r1-1.5b **(L)** | 94 | 95 | 104 | 10 | 144 | 38 | 31 | 19 | 38 | 39 | 612 | 2.732 | 14 | 6.25 |
| | Deepseek-r1-8b | 5 | 6 | 36 | 0 | 27 | 0 | 13 | 0 | 3 | 6 | 96 | 0.429 | 150 | 66.96 |
| | Deepseek-r1-14b **(M)** | 2 | 3 | 21 | 1 | 7 | 15 | 24 | 0 | 2 | 1 | 76 | 0.339 | 159 | 70.98 |
| | Deepseek-r1-32b **(H)** | 2 | 1 | 35 | 0 | 5 | 6 | 29 | 0 | 0 | 1 | 79 | 0.353 | 163 | 72.77 |
| Falcon3 (TII) | Falcon3-1b **(L)** | 33 | 14 | 107 | 1 | 50 | 6 | 0 | 9 | 30 | 10 | 260 | 1.161 | 89 | 39.73 |
| | Falcon3-7b **(M)** | 0 | 2 | 28 | 0 | 9 | 0 | 3 | 0 | 1 | 4 | 47 | 0.21 | 183 | 81.7 |
| | Falcon3-10b **(H)** | 0 | 2 | 16 | 0 | 3 | 0 | 6 | 0 | 1 | 4 | 32 | 0.143 | 194 | 86.61 |
| Llama2 (Meta) | Llama2-7B **(L)** | 1 | 11 | 44 | 1 | 10 | 59 | 5 | 2 | 15 | 2 | 150 | 0.67 | 114 | 50.89 |
| | Llama2-13B **(M)** | 0 | 3 | 59 | 1 | 5 | 206 | 4 | 1 | 6 | 4 | 289 | 1.29 | 38 | 16.96 |
| | Llama2-70B **(H)** | 45 | 12 | 73 | 1 | 4 | 0 | 13 | 2 | 0 | 6 | 156 | 0.696 | 145 | 64.73 |
| Mistral (Mistral AI) | Mistral-7B **(L)** | 1 | 2 | 39 | 0 | 5 | 0 | 9 | 0 | 3 | 1 | 60 | 0.268 | 174 | 77.68 |
| | Mistral-12B **(M)** | 6 | 2 | 46 | 0 | 5 | 0 | 13 | 0 | 2 | 16 | 90 | 0.402 | 156 | 69.64 |
| | Mistral-24B **(H)** | 0 | 1 | 22 | 0 | 4 | 1 | 15 | 0 | 2 | 1 | 46 | 0.205 | 185 | 82.59 |
| Phi3 (Microsoft) | Phi3-3.8B **(L)** | 26 | 0 | 149 | 1 | 38 | 20 | 18 | 3 | 7 | 31 | 293 | 1.308 | 61 | 27.23 |
| | Phi3-14B **(M)** | 1 | 18 | 47 | 0 | 16 | 17 | 19 | 0 | 7 | 22 | 147 | 0.656 | 121 | 54.02 |
| Qwen3 (Alibaba) | Qwen3-4B **(L)** | 10 | 8 | 42 | 1 | 2 | 20 | 16 | 0 | 4 | 2 | 105 | 0.469 | 145 | 64.73 |
| | Qwen3-8B | 0 | 0 | 22 | 0 | 1 | 0 | 13 | 0 | 3 | 0 | 39 | 0.174 | 189 | 84.38 |
| | Qwen3-12B **(M)** | 1 | 9 | 28 | 1 | 2 | 0 | 48 | 0 | 0 | 2 | 91 | 0.406 | 142 | 63.39 |
| | Qwen3-32B **(H)** | 0 | 3 | 37 | 0 | 1 | 3 | 21 | 0 | 0 | 1 | 66 | 0.295 | 166 | 74.11 |
| SmolLM2 (SmolVLM) | SmolLM2-135M **(L)** | 432 | 218 | 15 | 1 | 2 | 233 | 0 | 1 | 10 | 2 | 914 | 4.08 | 0 | 0 |
| | SmolLM2-350M **(M)** | 94 | 59 | 112 | 5 | 56 | 46 | 11 | 7 | 27 | 26 | 443 | 1.978 | 39 | 17.41 |
| | SmolLM2-1.7B **(H)** | 21 | 8 | 73 | 1 | 25 | 8 | 5 | 4 | 9 | 9 | 163 | 0.728 | 113 | 50.45 |
| GPT (OpenAI) | GPT-OSS-20B **(M)** | 0 | 0 | 68 | 0 | 112 | 1 | 11 | 0 | 0 | 0 | 192 | 0.857 | 100 | 44.64 |
| | GPT-3.5-Turbo **(H)** | 0 | 2 | 7 | 3 | 4 | 0 | 5 | 0 | 8 | 0 | 29 | 0.129 | 196 | 87.5 |
| Total | | 775 | 489 | 1407 | 28 | 545 | 681 | 382 | 50 | 180 | 195 | 4732 | - | 3495 | - |

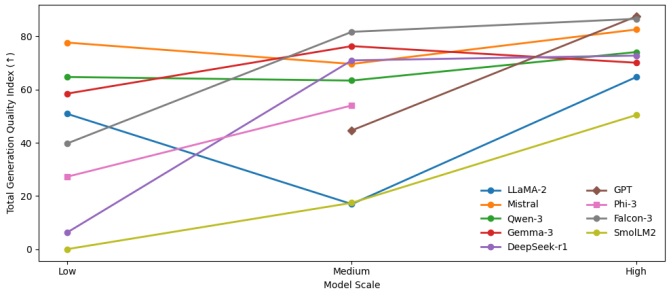



Fig. 3: Impact of Model Scaling on GQI. Generation Quality Index (GQI↑) plotted against model scale (L, M, H) for all LLM families.

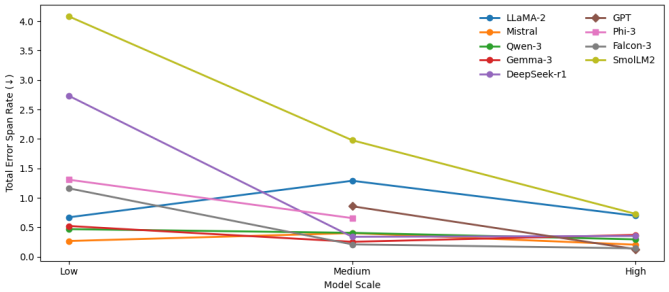



Fig. 4: Impact of Model Scaling on TESR. Total Error Span Rate (TESR↓) plotted against model scale (L, M, H) for all LLM families.

(M) and high (H) complexity models consistently outperform low (L) complexity models. Specifically, the average GQI increases sharply from the L-tier to the M-tier, indicating that crossing a certain parameter count threshold significantly improves the ability to produce error-free sentences. Exceptionally, The Llama2 family exhibits an anomalous dip in GQI at the M-tier (Llama2-13B: 16.96%), primarily due to a massive spike in Prompt Echo errors (206 instances, see Table II). This demonstrates that model scale alone does not guarantee robustness against instruction following failures.

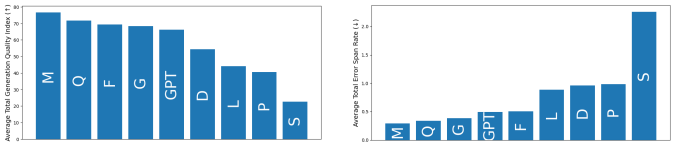
Fig 4 confirms the scaling trend inversely as TESR decreases as model size increases. High-tier models like Falcon3-10B (0.143) and GPT-3.5-Turbo (0.129) demonstrate the highest efficiency, generating an average of less than 0.15 errors per sentence. Conversely, the largest models in the dataset often exhibit performance comparable to, or slightly worse than, their M-tier counterparts (e.g., Gemma3-27b has a higher TESR than Gemma3-12b).

### *B. Model Family Rate and Efficiency*

To isolate architectural strengths from mere scale effects, we compare the average performance across the nine model families. When analyzing model family performance, Fig 5 (i) (for GQI) and Fig 5 (ii) (for TESR) demonstrate that the Mistral (Avg-GQI = 76.63%) and Qwen-3 (Avg-GQI = 71.65%) architectures emerged as the most reliable, demonstrating the highest average success rates across their scale tiers, with the Falcon-3 family also performing strongly (Avg-GQI = 69.35%). Furthermore, these families also displayed the highest efficiency, as measured by the Total Error Span Rate, with Mistral (Avg-TESR = 0.292) and Qwen-3 (Avg-TESR = 0.336) leading the group. This dual leadership in both success rate and efficiency validates their overall architectural superiority for structured Data-to-Text generation. Conversely, the smallest and most compact models, notably

TABLE III: Consolidated Performance and Error Profile of Selected LLM Variants across Three Distinct Prompt Strategies: No-Guidelines (NG), Standard Guidelines (SG), and Strict Fidelity (FG). The ablation evaluates $N = 224$ tuples per model to isolate the impact of instruction complexity on specific error modes.

| Prompt Strategies | Model | Entity Omission | Relation Omission | Addition | Repetition | Drift | Prompt Echo | Relation Ambiguity | Entity Type Changed | Semantic Incoherence | Partial Entity | #Total Errors | TESR ↓ | #Total Error-Free | GQI (%) ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No-Guidelines Prompt (NG) | Deepseek-r1-1.5b **(L)** | 74 | 53 | 65 | 5 | 115 | 108 | 5 | 9 | 21 | 10 | 465 | 2.076 | 34 | 15.18 |
| | Falcon3-1b | 38 | 4 | 98 | 4 | 29 | 3 | 3 | 0 | 6 | 5 | 190 | 0.848 | 102 | 45.54 |
| | SmolLM2-135M **(M)** | 442 | 223 | 0 | 0 | 2 | 108 | 0 | 0 | 0 | 0 | 775 | 3.46 | 0 | 0 |
| Standard Guidelines Prompt (SG) | Deepseek-r1-1.5b **(L)** | 94 | 95 | 104 | 10 | 144 | 38 | 31 | 19 | 38 | 39 | 612 | 2.732 | 14 | 6.25 |
| | Falcon3-1b **(L)** | 33 | 14 | 107 | 1 | 50 | 6 | 0 | 9 | 30 | 10 | 260 | 1.161 | 89 | 39.73 |
| | SmolLM2-135M **(L)** | 432 | 218 | 15 | 1 | 2 | 233 | 0 | 1 | 10 | 2 | 914 | 4.08 | 0 | 0 |
| Strict Guidelines Prompt (FG) | Deepseek-r1-1.5b **(L)** | 54 | 53 | 56 | 6 | 112 | 31 | 9 | 10 | 20 | 17 | 368 | 1.643 | 39 | 17.41 |
| | Falcon3-1b | 14 | 8 | 88 | 5 | 56 | 11 | 1 | 3 | 19 | 13 | 218 | 0.973 | 80 | 35.71 |
| | SmolLM2-135M **(M)** | 430 | 224 | 6 | 1 | 1 | 231 | 0 | 0 | 0 | 1 | 894 | 3.991 | 1 | 0.45 |
| Total | | 1611 | 892 | 539 | 33 | 511 | 769 | 49 | 51 | 144 | 97 | 4696 | - | 1656 | - |




(i) (ii)

Fig. 5: Mean (i) GQI and (ii) TESR scores across LLM families, aggregated over model scales. (L: LLaMA-2, M: Mistral, D: DeepSeek-r1, G: Gemma-3, P: Phi-3, F: Falcon-3, S: SmolLM2, Q: Qwen-3)

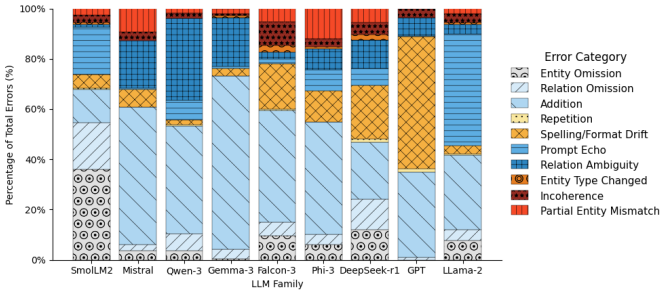



Fig. 6: Normalized distribution of fine-grained error categories across LLM families, revealing dominant architectural failure patterns.

SmolLM2, demonstrated the poorest performance (Avg-GQI = 22.62%, Avg-TESR = 2.262), highlighting a critical trade-off in highly size-constrained models, which often prioritize minimal parameter count over linguistic fidelity, resulting in highly erroneous output.

### C. Fine-Grained Error Distribution Analysis

Analysis of the 4732 total annotated error spans, visually broken down in Fig 6, reveals the dominant failure modes in LLM-based D2T generation. The most frequent error type globally is Addition (1,407 instances, or 29.7% of all errors), followed by Entity Omission (775 instances, 16.4%). The persistence of Addition across all families, including high-performing ones (e.g., Mistral), suggests that the tendency to hallucinate ungrounded details is a fundamental weakness in zero-shot D2T. Conversely, while Omission is a common global error, it is heavily concentrated in the smallest models (SmolLM2 contributing 547 total omissions), highlighting that adherence to input constraints is the primary weakness of the lowest-scale models.

The third most frequent global failure is Prompt Echo (681 instances, 14.4%), which is highly concentrated and catastrophic for specific variants like Llama2-13B (206 instances) and SmolLM2-135M (233 instances), demonstrating architectural susceptibility to instruction-adherence failures. Errors related to meaning and structure, such as Incoherence (180 instances), occur less frequently, suggesting that increased model scale is generally effective at mitigating severe linguistic meaninglessness.

### D. Qualitative Observations

Qualitative analysis revealed specific, unique failure modes across the LLM variants. The smallest variant, SmolLM2-135M (L), exhibited catastrophic instruction-following failures, frequently outputting irrelevant Python code or internal reasoning and ignoring the input tuple. This behavior led to its high Omission count and zero error-free generations (GQI of 0%). Distinct surface-level issues were also observed: GPT-OSS-20B (M) frequently generated unneeded whitespace breaks resulting in high Drift errors, while Qwen-3-4B (L) was the only model to produce unreadable (NaN) sentences in five instances. Regarding linguistic style, while the best M and H models (Qwen, DeepSeek, GPT-3.5-Turbo) achieved high fidelity and good realization of the tuple, other variants like Falcon often lacked creativity, simply realizing the relations without adding stylistic features, demonstrating a common trade-off between fidelity and linguistic diversity. Finally, Qwen and DeepSeek occasionally produced $< think > \cdots < /think >$ reasoning traces, which were removed during preprocessing and excluded from error annotation, as they reflect internal reasoning rather than generation errors.

### E. Prompt Ablation Studies

To further investigate the impact of instructional constraints on model performance, we conduct a targeted ablation study on a subset of models selected based on the error profiles observed in Table II. We focus on models exhibiting high rates of Entity Omission, Relation Omission, Relation Ambiguity, Incoherence, and Addition, as these errors directly violate the core D2T objective of faithfully verbalizing the input tuple into a coherent sentence. Based on these criteria, three representative models are further evaluated: Deepseek-r1-1.5b (L), Falcon3-1b (L), and SmolLM2-135M (L). Apart from

SG Prompt (Section III-C), two additional distinct prompt categories are devised:

1) **No Guideline (NG)**: The prompt from SG is utilized without any qualitative instructions to establish a baseline and measure the specific impact of instructional guidance.
2) **Strict Fidelity Guideline (FG)**: Building upon SG, we append two rigorous constraints designed to mitigate Entity Omission and Relation Omission in the D2T generation process.

> 1. You must include Entity1, Entity2, and the specific Relationship between them in the sentence you generate.
> 2. Do not introduce any external information, or facts not present in the tuple.

As shown in Table III, the Strict Fidelity strategy successfully mitigates semantic noise, reducing Relation Ambiguity in DeepSeek-R1-1.5B by 71%, but fails to correct severe Entity Omission rates in SmolLM2-135M, which remains nearly stagnant ($\approx$430 errors) across all prompt variations. Similarly, Addition persists as the dominant failure mode for Falcon3-1B regardless of strict negative constraints. This architectural intransigence demonstrates that while prompt engineering can refine the stylistic precision of reasoning-capable models, it cannot override the fundamental capacity deficits and hallucination propensities inherent to lower-parameter architectures, confirming that these failures are ultimately a function of model scale rather than instruction complexity.

## V. CONCLUSION

We present the first cross-family, multi-scale evaluation of Data-to-Text (D2T) generation using a novel 10-category span-level error taxonomy to assess 27 LLM variants on diverse DBPedia triples, yielding a reusable human-annotated benchmark of $\approx$2.5k sentences containing $\approx$4.7k fine-grained errors, enabling detailed analysis across model scales. This reveals non-uniform scaling behavior where smaller variants can outperform larger ones, confirming the architectural strength of families such as Mistral and Qwen-3 and exposing dominant global errors and catastrophic failure modes. The ablation studies show that stricter prompts help capable models but fail to fix dominant errors in smaller ones, indicating that these limitations stem from model capacity rather than instruction design. Ultimately, this work and the resulting span-annotated dataset provide a critical benchmark for the community, facilitating future research in automatic error detection and the development of dependable structured-data generation systems.

## ACKNOWLEDGMENT

The authors used Grammarly for grammar checking and minor language refinement to improve the clarity and readability.

## REFERENCES

[1] C. Dong, Y. Li, H. Gong, M. Chen, J. Li, Y. Shen, and M. Yang, "A survey of natural language generation," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–38, 2022.

[2] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," *arXiv e-prints*, pp. arXiv–2006, 2020.

[3] Z. Kasner and O. Dusek, "Beyond traditional benchmarks: Analyzing behaviors of open LLMs on data-to-text generation," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 12 045–12 072.

[4] R. Huidrom, A. Belz, and M. Lorandi, "Differences in semantic errors made by different types of data-to-text systems," in *Proceedings of the 17th International Natural Language Generation Conference*. Association for Computational Linguistics, 2024, pp. 609–621.

[5] C. Thomson and E. Reiter, "A gold standard methodology for evaluating accuracy in data-to-text systems," in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, pp. 158–168.

[6] A. Yang, A. Li, B. Yang, B. Zhang *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[7] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[8] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard *et al.*, "Gemma 3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2503.19786

[9] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," 2024. [Online]. Available: https://arxiv.org/abs/2404.14219

[10] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "The WebNLG challenge: Generating text from RDF data," in *Proceedings of the 10th International Conference on Natural Language Generation*. Association for Computational Linguistics, Sep. 2017.

[11] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM computing surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[12] O. Dušek and Z. Kasner, "Evaluating semantic accuracy of data-to-text generation with natural language inference," in *Proceedings of the 13th International Conference on Natural Language Generation*. Association for Computational Linguistics, 2020, pp. 131–137.

[13] J. Mahapatra and U. Garain, "Impact of model size on fine-tuned llm performance in data-to-text generation: A state-of-the-art investigation," *arXiv preprint arXiv:2407.14088*, 2024.

[14] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022, pp. 30 016–30 030.

[15] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web Journal*, 2015.

[16] A. Parekh, A. Anand, and A. Awekar, "Taxonomical hierarchy of canonicalized relations from multiple knowledge bases," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. Association for Computing Machinery, 2020, p. 200–203.

[17] L. B. Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček, A. P. Lajarín, V. Srivastav *et al.*, "Smollm2: When smol goes big–data-centric training of a small language model," *arXiv preprint arXiv:2502.02737*, 2025.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *ArXiv*, vol. abs/2310.06825, 2023.

[19] F.-L. Team, "The falcon 3 family of open models," December 2024. [Online]. Available: https://huggingface.co/blog/falcon3

[20] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[21] O. Team, "Gpt-3.5 turbo," 2023. [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5

[22] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao *et al.*, "gpt-oss-120b & gpt-oss-20b model card," *arXiv preprint arXiv:2508.10925*, 2025.

[23] S. Iravani, T. Conrad *et al.*, "Towards more effective table-to-text generation: Assessing in-context learning and self-evaluation with open-source models," *arXiv preprint arXiv:2410.12878*, 2024.

[24] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "brat: a web-based tool for NLP-assisted text annotation," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 102–107.