A Comparative Evaluation of End-to-End and Pipeline Approaches for Summarisation

Fahime Same
trivago N.V.
fahimeh.same@gmail.com

Saad Mahamood
Shopware
s.mahamood@shopware.com

Srinivas Ramesh Kamath
trivago N.V.
srinik352@gmail.com

Abstract

We describe and evaluate two different architectures for creating book highlights from unstructured data. Given the prevalence of large language models, we examine whether a pipeline-based approach with intermediate steps for text generation is still necessary and whether it continues to offer any benefits over an end-to-end approach. Our comparative evaluations using LLM-as-a-judge across multiple models with different parameter sizes and generation scenarios show that highlights generated by the end-to-end approach are preferred. However, there is a slight but consistent increase in faithfulness for the pipeline-generated highlights when generating at a thematic level. Additionally, our analysis across multiple models shows that while larger models are more faithful, the degree of faithfulness increases when they are used with a pipeline architecture. The findings from our work indicate that whilst there is comparability between the two approaches, the greater faithfulness, controllability, and observability of pipeline-based approaches offer tangible benefits in applied settings.

1 Introduction

Generating accurate and relevant information is essential for a Natural Language Generation (NLG) system that summarises facts. The use of LLMs introduces several problems for applied NLG applications such as the generation of semantically inaccurate output and the omission of content (Huidrom et al., 2024).

Efforts have been made to prevent LLMs from generating divergent information, with approaches that aim to enhance LLM reasoning through reflection and refinement (Shinn et al., 2023; Yan et al., 2024). However, LLMs often fail to adhere to instructions, fail to revise their incorrect predictions, and struggle with knowledge-rich problems (Yan et al., 2024). Most systems using LLMs rely on an end-to-end approach for generation with some attempts to correct or revise divergent information post-generation, despite evidence that rule-based pipeline approaches have consistently shown more semantic faithfulness than both neural non-LLM and LLM-based systems (Huidrom et al., 2024).

Most direct comparisons between the two architectures, however, are based on sequence-to-sequence or LSTM-based models (Castro Ferreira et al., 2019; Moryossef et al., 2019), leaving open several important questions. Modern LLM-based NLG systems vary substantially in model family, parameter scale, and training methodology (Zhao et al., 2026), and it is unclear whether the advantage of a pipeline architecture holds uniformly across these dimensions or whether that advantage increases or diminishes depending on the model used.

Moreover, prior work has generally evaluated generation at a single level of specificity, yet in practice tasks range from producing broad thematic summaries to generating narrower, more specific aspects of a work. These two levels of generation place different demands on content selection: broader thematic summaries may tolerate more abstraction, whereas narrower, more focused summaries may require precise identification and faithful rendering of specific facts, often from sparser source material. It is therefore possible that architectural control matters more for one level than the other.

Parameter scale introduces a further dimension: larger models within the same family are generally expected to produce higher-quality and more faithful output (Wei et al., 2022), but it remains an open question whether this advantage is consistent across architectures or whether the explicit content selection in a pipeline architecture already compensates for some of the weaknesses of smaller models.

In this paper, we investigate how system architecture and generation model characteristics affect the quality and faithfulness of automatically generated

book highlights. Our primary question is whether the architectural difference between an end-to-end system (E2E) and a pipeline system (PIPE) leads to observable differences in divergence from the source material, as well as in overall output quality and user preference.

Beyond architecture, we also examine whether these effects vary depending on the type of generation task. We generate highlights for two types of Knowledge Graph (KG) relations: relation-level highlights target Dublin Core Terms properties (e.g. `dct:subject`), which capture broad thematic categories, and tail-level highlights target specific category nodes (e.g. `cat:Novels_set_in_Europe`), which require more fine-grained, entity-specific content.

In addition, we study whether highlight quality and faithfulness differ across LLM families and parameter sizes, and whether larger models consistently yield higher-quality and more faithful generations. Finally, we ask whether the greater controllability of the PIPE system reduces the impact of model size and family, such that differences between smaller and larger models are less pronounced in the pipeline setting than in the end-to-end setting.[1]

2 Background

Rule-based NLG systems have relied on a data-to-text pipeline architecture (Reiter, 2007) to divide text generation into a series of discrete steps by selecting the most relevant aspects to summarise. However, the lack of generalisability and fluency has led to exploration into neural E2E approaches (Wen et al., 2015; Dušek and Jurčíček, 2016; Mei et al., 2016; Gehrmann et al., 2018). This approach removes the need for intermediate representations, as non-linguistic input is turned into natural language, but at the cost of explainability (Faille et al., 2020).

Attempts were made to combine the strengths of both approaches, with Castro Ferreira et al. (2019) comparing a neural pipeline against an E2E system. The pipeline not only produced better texts but also offered other benefits: explainability, validation, and controllability. Moryossef et al. (2019) also found that in their neural pipeline system, the ability to control the content generation step allowed for an explicit verification step by comparing the entities in the output with those in the content plan.

The common wisdom for language models has been that model performance depends most strongly on the number of model parameters, the size of the dataset, and the amount of compute (Kaplan et al., 2020). For data-to-text generation this relationship is not necessarily clear-cut. Mahapatra and Garain (2024) analysed multiple fine-tuned open models and found that higher-parameter models did not consistently outperform their smaller counterparts across several data-to-text datasets.

Nevertheless, contemporary LLMs have made significant progress in processing longer input contexts that can contain thousands of tokens from input sources such as multiple long documents. However, when answering questions from such long contexts LLMs can exhibit a "lost-in-the-middle" phenomenon, where the performance of the model in terms of question answering is the highest for information present at the beginning or at the end of the input context (Liu et al., 2024). Attempts have been made to mitigate this positional sensitivity in LLMs through techniques such as expanding the context window through the use of a sliding window (Dai et al., 2019; Xiao et al., 2024) or improving how positional information is incorporated into the learning process for transformer models (Su et al., 2024). An alternative approach has been to work around the problem by compressing and segmenting the initial input and presenting only the relevant segment(s) to the LLM for the given query (Chen et al., 2023; Lee et al., 2024).

To bring greater controllability, several systems combine pipeline architectures with LLMs. Avignone et al. (2024) used GPT-2 to lexicalise structured data into product text descriptions, with the input undergoing selection and pre-processing steps prior to generation. Others have focused on general-purpose unsupervised approaches to data-to-text generation with LLMs (Laha et al., 2020), or zero-shot approaches (Kasner and Dusek, 2022) that avoid fine-tuning pre-trained language models and thus over-fitting to a particular benchmark. Hashem et al. (2024) used knowledge graphs to validate the output of large multimodal language models and allow more faithful generation. Common to these systems is the need for discrete steps that separate content selection (what to say) from generation (how to say it).

[1] The datasets, annotations, evaluation code, and prompts from this work are available at https://github.com/fsame/book_summarization_e2e_pipeline.git

3 Research Questions and Hypotheses

In this section, we introduce the research questions and hypotheses underlying this study. We investigate how architectural design, generation level, model family, and model size affect the quality and faithfulness of automatically generated book highlights. Our primary comparison is between an end-to-end and a pipeline architecture, but we also examine whether this comparison changes depending on whether highlights are generated for broader relation-level categories or more specific tail-level targets, and whether it varies across model families and parameter scales.

Because the PIPE architecture explicitly separates content selection from realisation, it offers greater control over what information is verbalised. This may help reduce unsupported content and improve overall usefulness compared with an E2E architecture, which must learn content selection and generation jointly. Based on this, our first research question is: **RQ1: Does system architecture (E2E vs. PIPE) affect overall highlight quality and faithfulness?** We hypothesise that **H1: overall, PIPE produces higher-quality and more faithful highlights than E2E.**

The relative advantage of these architectures may depend on the type of generation task. Relation-level highlights concern broader categories such as theme, author, or genre, whereas tail-level highlights require more specific and fine-grained information. Since tail-level generation places greater demands on content selection, architectural control may be especially important in this setting. Accordingly, our second research question is: **RQ2: Does the effect of architecture differ between relation-level and tail-level generation?** We hypothesise that **H2: the advantage of PIPE over E2E is larger for tail-level generation than for relation-level generation.**

At the same time, model scale may influence both quality and faithfulness. Larger models typically show stronger language generation abilities, better instruction following, and more robust handling of complex input information than smaller models. Given this, our third research question is: **RQ3: How does model size affect highlight quality and faithfulness within a family?** We hypothesise that **H3: within each LLM family, larger models produce higher-quality and more faithful highlights than smaller models.**

Model size may also interact with architecture. Smaller models are more likely to struggle when they must jointly decide what to say and how to say it, as in the E2E setting, whereas the decomposition in PIPE may compensate for some of these limitations. Larger models, by contrast, may already handle this complexity more effectively. Based on this, our fourth research question is: **RQ4: Does the effect of architecture depend on model size?** We hypothesise that **H4: the advantage of PIPE over E2E is larger for smaller models than for larger models.**

Finally, these effects may vary across LLM families, since families differ in training data, alignment strategies, instruction-following behaviour, and stylistic tendencies. Such differences may influence how strongly a model benefits from the additional controllability provided by the pipeline architecture. Therefore, our fifth research question is: **RQ5: Does the effect of architecture vary across LLM families?** We hypothesise that **H5: the extent of the PIPE advantage varies across model families.**

4 System Implementations

We created two comparable systems that use the same input sources: the book metadata, descriptions, and user reviews from the 2018 Amazon review dataset (Ni et al., 2019). Additionally, we used the Amazon Knowledge Graph (KG) dataset (Wang et al., 2024) that defines several relation types for each book. As described in §1, both systems take these sources as input and generate short highlights for each KG relation type, at both the broad thematic level (relation level) and the specific category node level (tail level).

4.1 Data Selection

The input for both systems was limited to books with descriptions of at least 100 characters, at least 10 reviews, and all of the following KG relation types: SUBJECT, AUTHOR, GENRE, PREVIOUSWORK and SUBSEQUENTWORK. This yielded 148 books. For comparability with an earlier evaluation, we further restricted the final selection to 88 books from this sample.

4.2 Models and Comparison Factors

In line with RQ3, our aim is to test the generation of highlights across a variety of open-source models and different parameter sizes.

We selected models from three provider families: OpenAI (`gpt-oss-20b` and `gpt-oss-120b`;

OpenAI 2025b), Meta (`llama-3.1-8b` and `llama-3.1-70b`; Grattafiori et al. 2024), and Qwen (`qwen-3-8b` and `qwen-3-32b`; Team 2025). Within each family, we paired a smaller and a larger model to examine whether parameter scale affects output quality and faithfulness, and whether this size effect interacts with the choice of generation architecture. Table 1 shows the list of models and their corresponding parameter sizes considered for generation with the two architectures (E2E and PIPE).

Model	Parameter Size
GPT-OSS	20bn
GPT-OSS	120bn
Llama_3.1	8bn
Llama_3.1	70bn
Qwen_3	8bn
Qwen_3	32bn

Table 1: Models used for generation (E2E & PIPE).

4.3 E2E Implementation

In the E2E system (Figure 1), we used zero-shot prompting to generate book highlights for each selected book, relation type, and tail node. The prompt assigned a copywriter persona, a summarisation task, and generation criteria. The input included a description and reviews. The output was a JSON array of highlights, each containing a title, text, relation type or tail node, and the sources used to generate that highlight.

4.4 PIPE Implementation

Figure 1 shows the E2E and PIPE architectures. Unlike E2E, PIPE included additional steps before generation, which are described in the following paragraphs.

Data Ingestion and Analysis The description and reviews are first ingested by the data analysis module. Reviews of 25 words or fewer are filtered out, as they potentially lack relevant or detailed information about the book.

Review sentiment analysis was conducted for each review to ensure consistency between the review score and sentiment. Given the large number of reviews (29,414), a two-step process was used. The PIPE system first applies a simple valence-aware sentiment model (Hutto and Gilbert, 2014) to classify the review sentiment, and then uses a more complex RoBERTa-based model (Barbieri et al., 2022) for more complex or edge cases. If the sentiment result matches the score, the review is retained; otherwise, it is discarded (1,091 reviews were removed).

Data Interpretation and Selection Next, each sentence from the description and reviews matched to one or more KG relation types using the LangExtract library (Google, 2025) for structured information extraction with the `gpt-5-thinking-nano` model (OpenAI, 2025a). A one-shot prompting approach was used, pairing instructions with a grounded example and a relation class label for each sentence. Further filtering pruned theme nodes without content, as well as nodes that have content but lack sentiment. The remaining nodes are then ordered as mapped content within one or more product relation types.

Generation of Book Highlights Like the E2E system, the PIPE system used the same prompt and model; the key difference was that only the selected content for each relation node was input to the LLM.

5 Evaluations

We evaluate the generated highlights using LLM-as-a-judge assessments. The evaluation is designed to cover two generation units (relation and tail), and two comparison types (architecture and parameter size).

5.1 Comparison Setup

Pairs are formed within each generation unit (relation-level and tail-level) separately. Within each, we construct two types of matched pairs: architecture pairs, which contrast E2E and PIPE outputs from the same book, model family, and model size; and size pairs, which contrast smaller and larger models from the same book, family, and architecture. We describe how matched pairs are constructed under this design in §5.2.

5.2 Sample Construction

Because our comparisons require matched pairs across architectures and models, we first restricted the 88-book generation pool to those for which both architectures produced highlights for all five relation types (63 books), and then only those for which all twelve architecture–model combinations were available at both the relation and tail levels. This led to a set of 57 books, from which all matched pairs are drawn. We further filtered this set to books

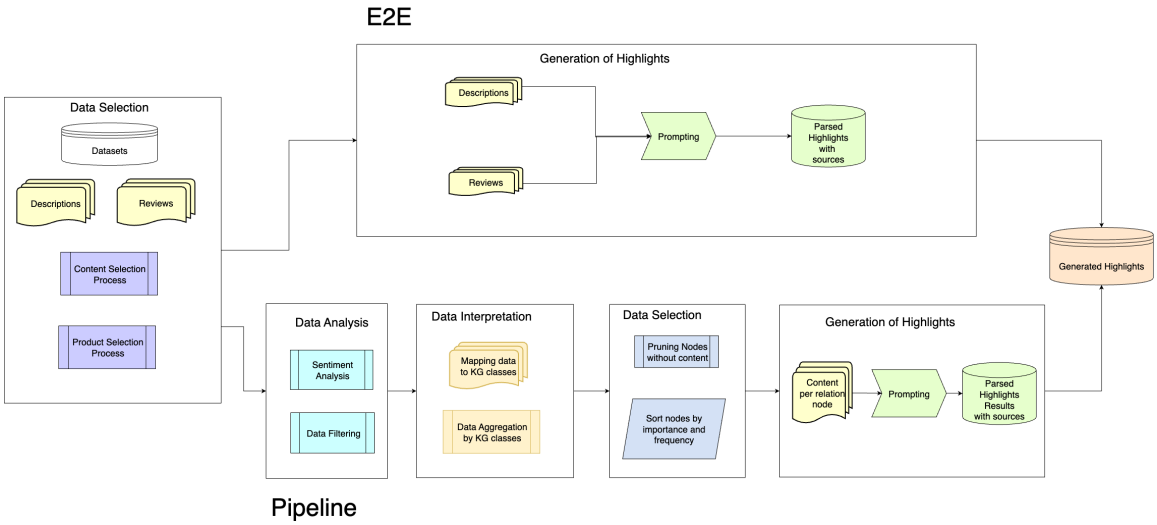

Figure 1: Highlights system architectures for both E2E and PIPE

with between 100 and 1000 reviews, so that each book had sufficient but not overwhelming source material, yielding a final set of 31 books.

Architecture comparisons pair E2E and PIPE outputs generated by the same model (e.g., both produced by Llama-3.1-70B), while size comparisons pair smaller and larger models within the same family and architecture (e.g., Llama-3.1-8B vs. Llama-3.1-70B, both E2E). Pairs are matched on book and relation type at the relation level (e.g., the same book under the *author* relation), and on the normalised tail instance at the tail level (e.g., the tail *Novels set in Edinburgh* for the same book).

Since more than one highlight was generated for each relation type or tail node, we use three different modes for sampling: `convergent`, `divergent`, and `random`. Convergent and divergent pairs are defined using the cosine similarity between the text embeddings of the two highlights in a pair. Convergent pairs contain outputs with relatively high semantic similarity, divergent pairs contain outputs with relatively low semantic similarity, and random pairs provide a random comparison baseline.

5.3 LLM-as-a-Judge Evaluation

Both the faithfulness and preference experiments use the same LLM judge, `gemini-2.5-pro`, in a zero-shot setting with structured JSON outputs. For each item, the judge returns a decision together with a short rationale and, where applicable, a confidence score. The two experiments differ in units of analysis and rubric: preference is assessed per pair under blinded conditions (§5.3.1), whereas faithfulness is assessed per highlight against its source text (§5.3.2).

5.3.1 Preference Judgments Evaluation

For the LLM-as-a-judge preference experiment, same-theme highlight pairs for the same book were compared under blinded conditions, with outputs presented as *Candidate A* and *Candidate B*. Candidate order was randomised to prevent the judge from exploiting position bias. Each pair was rated on six intrinsic criteria: *Informativeness*, *Saliency*, *Fluency/Style*, *Coherence*, *Theme Adherence*, and *Overall Preference*. For each criterion, the judge selected one candidate or indicated *tie* or *neither*. In addition, the judge returned a brief rationale and a confidence score for each criterion and for the overall decision. Table 4 in Appendix A presents the definitions used in the experiment, and Table 5 in Appendix B shows a few sample pairs used in the experiment under different conditions.

5.3.2 Faithfulness Assessment

Unlike the preference experiment, the faithfulness experiment evaluates each highlight individually rather than in pairs. Each generated highlight was assessed against the source text of its book, defined as the concatenation of the book's description and reviews, and the judge was instructed to use only the provided source as evidence. For each highlight, the judge produced four outputs: a binary *factual accuracy* label, a *divergence type* chosen from NONE, HALLUCINATION (Definition: *The highlight introduces at least one unsupported claim that is not established anywhere in the source*), CONTRADICTION (Definition: *The highlight states at least one claim that conflicts with the source*),

	N (dec.)	E2E win rate	95% CrI
Overall	2,938	62.8%***	(61.0, 64.5)
Relation	2,308	65.9%***	(63.9, 67.8)
Tail	630	51.6%	(47.7, 55.5)

Table 2: Architecture comparison (E2E vs. PIPE), overall and by generation level. Significance stars are for a two-sided binomial test against a 50% baseline; 95% CrI from a Beta–binomial model with a uniform prior. *** $p < .001$.

BOTH, and, whenever the divergence type was not NONE, a *severity score* on an integer 1–7 scale indicating how critical the error is and how strongly it affects the reader's understanding. The judge also returned a short rationale grounded in the source text. Faithful paraphrases were accepted, and omissions were not penalised; a highlight was marked inaccurate whenever any material claim was unsupported by or contradicted the source. Table 6 in Appendix B shows some examples and their LLM-as-a-judge annotations.

6 Results

We report results from two LLM-as-a-judge evaluations: a pairwise *preference* assessment (§6.1) and a per-highlight *faithfulness* assessment (§6.2). In both, we first examine the effect of system architecture and its moderation by the generation level (RQ1, RQ2), and then the effects of model family and parameter size, and their interaction with architecture (RQ3–RQ5).

6.1 Preference Judgment Evaluation Results

Contrary to our first hypothesis (H1), the judge preferred E2E outputs over PIPE outputs in 62.8% of decisive pairs (the *Overall Preference* criterion)[2], but, as Table 2 shows, this advantage is almost entirely restricted to relation-level highlights and vanishes at the tail level.

Where does the E2E advantage come from? The overall preference for E2E (62.8% of decisive pairs) shows a strong content-level asymmetry. At the relation level, E2E dominates (65.9%; 95% CrI 63.9–67.8%, posterior mass entirely above parity). At the tail level, where both systems describe an entity at a more granular level, the advantage vanishes: E2E wins 325 of 630 decisive pairs (51.6%; With a 95% CrI 47.7–55.5% that includes 0.5), indistinguishable from chance (binomial $p = 0.45$). A logistic regression confirms this asymmetry: pipeline wins are $1.81\times$ more likely at the tail level than at the relation level ($\hat{\beta} = 0.59$, SE $= 0.09$, $z = 6.52$, $p < 10^{-10}$). Consistent with H2, the E2E advantage is confined to the broader thematic generation task.

What drives the preference? We examine two complementary aspects: how often the two systems are judged equal on each criterion (tie rate), and which criteria actually determine the overall verdict (dominant factors).

As shown in Figure 2, for surface-realisation criteria, the judge overwhelmingly rates the two systems as equal: 78% of all pairs receive a tie on Coherence and 62% on Fluency & Style. Content-selection criteria produce far more decisive judgements — fewer than 3% of pairs are tied on Informativeness — and it is here that E2E holds its largest margins: 74.6% of decisive judgements on Theme Adherence and 65.4% on Informativeness favour E2E. When the judge explicitly names the factor that drove the overall verdict, Informativeness (2,580 pairs) and Saliency (1,647 pairs) dominate, while Fluency and Coherence are rarely cited (471 and 198 pairs).

Taken together, the results suggest that the two architectures produce stylistically comparable output, but that E2E more reliably selects contextually appropriate and thematically grounded content.

Effects of LLM Family and Parameter Size The E2E advantage is not modulated by model scale: win rates are 61.7% against small PIPE outputs and 63.9% against large-model outputs. A logistic regression confirms that the gap between architectures is similar regardless of size (OR $= 0.91$, $p = .22$), contrary to H4.

Consistent with H5, the E2E advantage varies across families (58.7%–68.1%), but the direction is consistent: E2E wins in all three families, most reliably on Theme Adherence (see Appendix C for the full family $\times$ criterion breakdown).

Regarding H3, a modest overall size advantage exists (53.9%), but it is uneven: within GPT-OSS, larger models win consistently across all criteria; within Qwen, the advantage holds overall but not on Theme Adherence; and within Llama, the larger-model advantage is mostly absent, with a striking reversal on Fluency & Style where the smaller model wins 63% of decisive pairs. This

[2] Throughout §6 we report win rates on *decisive* pairs, i.e., pairs in which the judge selected a single winner rather than *tie* or *neither*.

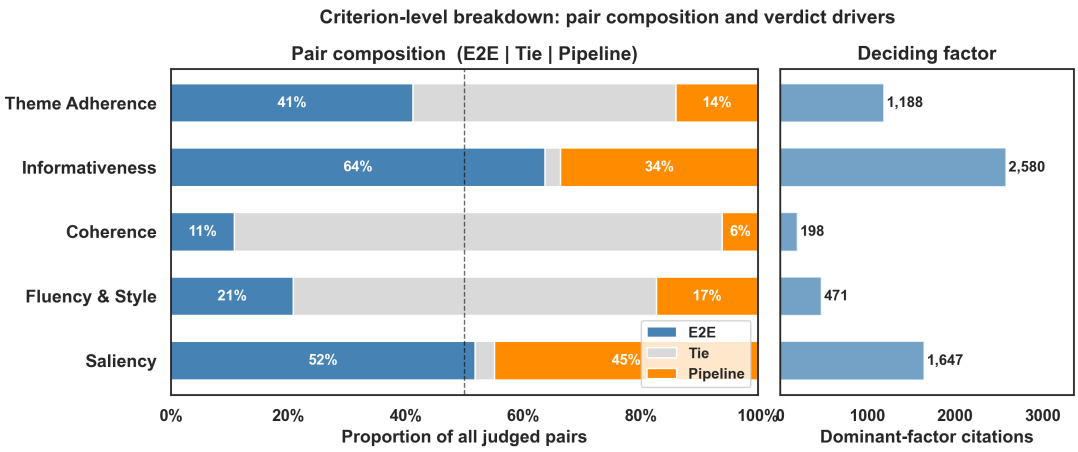

Figure 2: Criterion-level breakdown of judge preferences (E2E vs. PIPE) across all judged pairs. **Left:** proportion of pairs rated as E2E win (blue), tie (grey), or PIPE win (orange) on each evaluation criterion. **Right:** number of times each criterion was cited by the judge as the dominant factor driving its overall preference decision.

Llama-specific pattern might be due to the size-differentiated post-training pipeline described by Grattafiori et al. (2024), where the 8B and 70B models receive different synthetic training data and iterative style-steering, which may produce divergent stylistic outputs independently of model size. The comparison is further complicated by the non-comparable configurations across families (parameter ratios 4×–9×; context windows 32K vs. 128K).

6.2 Faithfulness Evaluation Results

Pipeline is slightly more faithful overall. Across 5,488 judged outputs (1,814 E2E and 3,674 PIPE)[3], PIPE outputs are rated fully faithful 77.4% of the time, compared with 73.8% for E2E. A chi-squared test of independence on the architecture × divergence-type table confirms that the two architectures produce significantly different distributions of error types ($\chi^2(3) = 18.58$, $p < .001$). The gap is small but consistent, and it is entirely driven by hallucination: E2E outputs are 1.36× more likely to hallucinate than PIPE outputs (OR $= 1.36$, $p < 0.001$), while contradiction rates are virtually identical across the two architectures ($p = 0.52$). The lower hallucination rate in PIPE may reflect the fact that decomposing the generation task into explicit retrieval and generation steps gives the model less opportunity to drift from the source material. Table 3 shows the full breakdown by content level.

[3]The imbalance arises because the PIPE system produced a larger and more diverse set of highlights overall, resulting in more unique outputs after deduplication by highlight identity.

	Faithful		Hallucination		Contradiction	
	E2E	Pipe	E2E	Pipe	E2E	Pipe
Overall	73.8	77.4**	13.9	10.1**	10.5	11.0
Relation	69.7	75.6***	17.0	11.6***	11.0	11.0
Tail	87.0	85.3	4.0	3.4	8.7	10.8

Table 3: Faithfulness rates (%) by architecture and content level. Significance markers indicate a reliable difference between E2E and Pipeline within that row (logistic regression): $^{**}p < .01$, $^{***}p < .001$. The architecture × level interaction is significant ($p = .025$).

The gap disappears at the tail level. At the relation level, the architecture effect is clear: PIPE achieves a 75.6% faithfulness rate versus 69.7% for E2E, with the difference concentrated in hallucination (17.0% for E2E vs. 11.6% for PIPE). At the tail level, however, the gap narrows and reverses: E2E is marginally more faithful (87.0%) than PIPE (85.3%). A logistic regression confirms that the architecture effect is significantly stronger at the relation level than at the tail level ($p = 0.025$), consistent with H2. A likely explanation is that tail generation targets a specific entity whose relevant properties are already localised in the retrieved context, leaving less room for hallucination regardless of how the generation is structured.

Size and family effects. We next ask whether faithfulness varies with model size and family, and whether these factors interact with architecture. Larger models are more faithful within GPT-OSS (63.2% → 71.7%) and Llama (81.4% → 90.1%), both gaining roughly 8 points from small to large (see Figure 3), but not within Qwen, where the two sizes are essentially equal (77.8% vs. 76.8%).

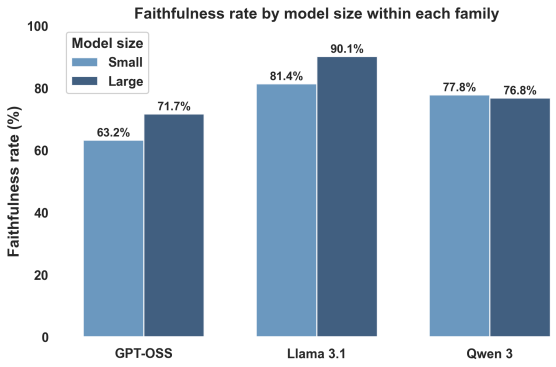

Figure 3: Faithful rate by model size within each family. GPT-OSS and Llama show a clear size benefit; Qwen does not.

Looking at how the architecture effect varies with model size and family (RQ4–5), neither factor significantly changes it overall. The pipeline advantage is slightly larger for large models (+5.6 points) than for small ones (+2.2). Across families, the direction of the architecture effect is broadly consistent at the relation level, but tail-level estimates are unreliable due to sparse coverage in some cells; the full breakdown is in Appendix D, Figure 5. In other words, whichever model size or family that is used, E2E outputs are somewhat more likely to hallucinate than PIPE outputs by the same margin. The more striking result is that Llama is substantially more faithful than GPT-OSS regardless of architecture or size (OR $= 2.83$, $p < 0.001$), a gap that reflects model family characteristics rather than architecture or size.

7 Discussion

Preference and faithfulness pull in different directions. E2E outputs are more preferred but less faithful. At the relation level, judges favour their thematic scope and content selection, yet these same outputs hallucinate more. This is not entirely surprising: generating broad, engaging highlights likely requires drawing on knowledge beyond what the retrieved context provides, which is exactly what pipeline decomposition is designed to prevent. What is perhaps more surprising is that architecture affected hallucination rates but not contradiction rates. It might be the case that pipeline changes what the model draws on, not how carefully it reads what it has.

At the tail level, architecture does not matter. For entity-specific generation, both preference and faithfulness are nearly identical across architectures. The retrieved context is narrow enough that both systems stay close to it, and judges cannot consistently tell them apart. This suggests the architecture choice is most consequential for relation-level highlights, and less so once the generation task is tightly constrained by a specific entity. More broadly, summarisation work often treats the task at a single level of abstraction without separating settings where the source material is rich from settings where it is sparse. Our results suggest that conclusions about which architecture is better should be stated relative to the specificity of the generation target, not as a single global ranking.

Model family matters more than architecture. Llama is substantially more faithful than GPT-OSS regardless of architecture or size. This gap is larger than any architecture effect in the data. Whatever drives it (instruction tuning, context utilisation, alignment), it is not something that switching from E2E to PIPE can replicate. In practice, choosing the right base model may have more impact on faithfulness than choosing the right system design.

8 Conclusion

We examined how architecture, model family, and parameter size shape the quality and faithfulness of LLM-generated book highlights. Our LLM-as-a-judge evaluations showed that pipeline outputs are more faithful while end-to-end outputs are more preferred, with both effects concentrated on broader, thematic generation tasks and absent for more specific, entity-level tasks, where the two architectures converge on both dimensions. Scale improves faithfulness within GPT-OSS and Llama but not Qwen, and model family is a stronger predictor of faithfulness than either architecture or size. Contrary to our expectation, switching to pipeline does not narrow the gap between smaller and larger models, nor between model families: the architecture effect is similar regardless of model size or family. Together, these results suggest that end-to-end and pipeline generation involve a genuine trade-off: end-to-end outputs are more preferred, whereas pipeline outputs are more faithful, and that this trade-off is most consequential for broader thematic generation tasks rather than entity-specific ones.

Limitations

The evaluation relies on an LLM-as-a-judge setup for both preference and faithfulness judgements.

Beyond a potential bias towards longer or more elaborated outputs, the judge model may have inherent preferences that are not fully aligned with human judgement. For example, it may favour outputs that resemble its own generation style. Comparing LLM-based evaluations against human annotations would help establish how much these biases affect the conclusions.

The study covers a single domain (book descriptions and reviews), which limits how far the findings generalise. Knowledge graphs for other domains may have different relation structures, retrieval properties, and levels of source sparsity, all of which could shift the balance between E2E and PIPE generation. Extending the evaluation to other domains would help clarify which findings are domain-specific and which are more general.

Model configurations are not matched across families for size comparisons: the parameter ratios and context windows differ substantially between GPT-OSS, Llama, and Qwen. This makes it difficult to attribute cross-family differences in faithfulness to scale alone, as opposed to other architectural and training differences. A more controlled comparison, holding context window and parameter count constant across families, would allow stronger conclusions about scale effects.

The preference judgment dataset includes divergent, convergent, and random pairs, designed to test whether sampling strategy affects the results. Due to space constraints, we do not analyse these sub-conditions here; differences across pair types, as well as individual book-level variation, are left for future work.

Finally, tail-level analyses are based on substantially fewer samples than relation-level ones, and some sub-group cells have sparse coverage.

References

Andrea Avignone, Alessandro Fiori, Silvia Chiusano, Giuseppe Rizzo, and 1 others. 2024. From product sheet to text and video: A nlg pipeline to transform structured data into comprehensive descriptions. In *Proceedings of the 32nd Symposium of Advanced Database Systems, Villasimius, Italy, June 23rd to 26th, 2024*, volume 3741, pages 271–280. CEUR-WS.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *Preprint*, arXiv:2310.05029.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics.

Juliette Faille, Albert Gatt, and Claire Gardent. 2020. The natural language pipeline, neural text generation and explainability. In *2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pages 16–21, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Google. 2025. Langextract. `https://github.com/google/langextract`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Tahsina Hashem, Weiqing Wang, Derry Tanti Wijaya, Mohammed Eunus Ali, and Yuan-Fang Li. 2024. Generating faithful and salient text from multimodal

data. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 646–662, Tokyo, Japan. Association for Computational Linguistics.

Rudali Huidrom, Anya Belz, and Michela Lorandi. 2024. Differences in semantic errors made by different types of data-to-text systems. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 609–621, Tokyo, Japan. Association for Computational Linguistics.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Anirban Laha, Parag Jain, Abhijit Mishra, and Karthik Sankaranarayanan. 2020. Scalable micro-planned generation of discourse from structured data. *Computational Linguistics*, 45(4):737–763.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26396–26415. PMLR.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Joy Mahapatra and Utpal Garain. 2024. Impact of model size on fine-tuned llm performance in data-to-text generation: A state-of-the-art investigation. *Preprint*, arXiv:2407.14088.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. What to talk about and how? selective generation using LSTMs with coarse-to-fine alignment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 720–730, San Diego, California. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Improving quality and efficiency in plan-based neural data-to-text generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 377–382, Tokyo, Japan. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2025a. Gpt-5 system card.

OpenAI. 2025b. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*, pages 97–104, Saarbrücken, Germany. DFKI GmbH.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yuhan Wang, Qing Xie, Mengzi Tang, Lin Li, Jingling Yuan, and Yongjian Liu. 2024. Amazon-kg: A knowledge graph enhanced cross-domain recommendation dataset. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 123–130, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.

Hanqi Yan, Qinglin Zhu, Xinyu Wang, Lin Gui, and Yulan He. 2024. Mirror: Multiple-perspective self-reflection method for knowledge-rich reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7086–7103, Bangkok, Thailand. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2026. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Preference Judgment Criteria

Table 4 lists the criteria and their definitions as presented to the judge in the preference judgment experiment.

B Example Outputs from the LLM-as-a-Judge Experiments

Tables 5 and 6 show example outputs from the two LLM-as-a-judge experiments described in §5.3.

Table 5 presents one example pair per sample type from the preference judgment experiment, restricted to convergent pairs. Each row shows the two candidate highlights presented to the judge (blinded), the overall winner and its resolved source (E2E or PIPE; small or large), the criteria that drove the decision, and the judge's rationale.

Table 6 presents one example per combination of divergence type, content level, model size, and system architecture from the faithfulness experiment. Each row shows the generated highlight, the judge's verdict (*none* = fully faithful, *hallucination*, *contradiction*, or *both*), the assigned severity, and the judge's rationale.

C Family and Criterion Breakdown

Figure 4 shows the E2E win rate (top) and large-model win rate (bottom) broken down by model family and evaluation criterion.

D Faithfulness Gap by Model Family and Content Level

Figure 5 shows the difference in faithfulness rate between PIPE and E2E for each combination of model family and content level. Green cells indicate a pipeline advantage; red cells indicate an E2E advantage. The Qwen tail-level cell should be interpreted with caution due to sparse PIPE output coverage at that condition.

Criterion	Definition & Question
Informativeness	How much useful, concrete, book-specific information the candidate conveys for the stated theme. Prefer the candidate that provides more meaningful detail rather than vague or generic wording, but do not reward unsupported specificity. *Which candidate conveys more useful, book-specific information for the stated theme?*
Saliency	How well the candidate surfaces a point that would stand out to a reader and help them decide whether the book is worth attention. Prefer the candidate that highlights a more compelling or decision-relevant selling point, without rewarding hype alone. *Which candidate highlights a more compelling and decision-relevant selling point for a reader?*
Fluency & Style	How natural, polished, and readable the candidate is. Prefer grammatical, idiomatic, concise, and well-phrased text, and penalize awkward wording, repetition, malformed syntax, or obvious style issues. *Which candidate is more natural, polished, and readable?*
Coherence	How logically organized and internally consistent the candidate is. Prefer the candidate whose claims fit together cleanly and are easy to follow, and penalize contradictions, abrupt jumps, unclear referents, or confusing structure. *Which candidate is more logically organized and internally consistent?*
Theme Adherence	How well the candidate stays focused on the intended theme instead of drifting to another aspect of the book. Prefer the candidate that clearly addresses the provided theme, and penalize off-theme details, mixed themes, or weak connection to the requested aspect. *Which candidate stays more clearly focused on the intended theme?*
Overall Preference	If only one of the two candidates could be shown for the given theme, which one should be selected? This judgment should be based on the full rubric rather than on any single criterion alone. *If you could show only one of the two candidates for this theme, which one would you choose?*

Table 4: Criteria and definitions used in the pairwise preference LLM-as-a-judge experiment.

Metadata	Highlight A	Highlight B	Winner	Dominant Factors	Rationale (shortened)
Type: *relation* **Comparison:** *Architecture* **Book:** *The Heart of the Matter* **Theme:** *previousWork*	**PIPE:** Like other Greene novels, Heart of the Matter follows settings in foreign times and places.	**E2E:**Preceded by 'A Burnt Out Case,' sharing Greene's focus on moral ambiguity and colonial tensions.	B (E2E)	informativeness, saliency	B names a specific prior work and highlights salient themes that help a reader decide. A is too vague.
Type: *relation* **Comparison:** *Size* **Book:** *Under the Banner of Heaven* **Theme:** *author*	**Small:** Author Jon Krakauer examines the connection between religion and violence in his book.	**Large:** Jon Krakauer is a gifted writer, known for his meticulous research and engaging storytelling.	B (large)	theme_adherence, informative-ness	B addresses the author theme directly. A describes book content but says nothing about the author.
Type: *tail* **Comparison:** *architecture* **Book:** The Terminal Man **Theme:** Michael Crichton	**E2E**: Provides an early glimpse into Crichton's writing style and his ability to craft engaging stories.	**PIPE**: Crichton knows how to build suspense and develop characters, making the reader invested in the outcome.	B (pipe.)	informativeness, saliency	B gives specific stylistic details (suspense, character development); A is more generic.
Type: *tail* **Comparison:** *size* **Book:** Espresso Tales **Theme:** serial publication	**Large:** Originally published as daily installments in *The Scotsman*, blending comedy and character-driven storytelling.	**Small:** Published as a serial novel, with short chapters that mirror the original newspaper format.	A (large)	informativeness	A names the specific publication and connects format to content. B is correct but too generic.

Table 5: Example convergent preference pairs, one per sample type. W. = overall winner (resolved label in parentheses).

Metadata	Highlight	Div.	Sev.	Rationale
Level: *relation* **Size:** *large* **Sys.:** *pipeline* **Book:** *Sphere* **Theme:** *subsequentWork*	Fans can expect a high-octane movie version, packed with nonstop action, drama, and tension, delivering the novel's adventurous spirit to the screen.	both	5.0	The highlight claims the film adaptation is action-packed and delivers the novel's spirit. The source does not support the 'nonstop action' claim, and numerous reviews directly contradict the idea that the film was a successful adaptation, with one calling it a 'disgrace to the book' and another stating it was 'nowhere near as good' as the book.
Level: *tail* **Size:** *small* **Sys.:** *e2e* **Book:** *Tatiana and Alexander* **Theme:** *nan*	First installment of a critically acclaimed trilogy, setting the stage for the next two novels.	both	6.0	The highlight incorrectly identifies this book as the 'First installment' of a trilogy; the source text repeatedly describes it as the second book and a sequel to 'The Bronze Horseman'. Consequently, the claim it 'sets the stage for the next two novels' is also a contradiction, as only one book follows. The term 'critically acclaimed' is a hallucination, as it is unsupported by the provided user reviews.
Level: *relation* **Size:** *large* **Sys.:** *pipeline* **Book:** *Bloodsucking Fiends* **Theme:** *previousWork*	Readers note this novel picks up after Moore's cult classic Bloodsucking Fiends, delivering the same off-beat humor and supernatural chaos fans love.	contradiction	5.0	The highlight claims the novel is a 'Follow-up to Bloodsucking Fiends.' The source material repeatedly identifies 'Bloodsucking Fiends' as the first book in the series, not a sequel.
Level: *relation* **Size:** *small* **Sys.:** *e2e* **Book:** *Guards! Guards!* **Theme:** *previousWork*	Follows earlier works like 'The Color of Magic' and 'The Light Fantastic' in the series.	contradiction	5.0	The highlight claims the book is a prequel to Discworld, but multiple parts of the source text state that it is the eighth book in the series, not a prequel.
Level: *tail* **Size:** *large* **Sys.:** *e2e* **Book:** *The Terminal Man* **Theme:** *American science fiction novels*	Reviews praise Crichton's foresight on brain-computer interfaces, marking the novel as a cornerstone of American sci-fi literature that still resonates.	hallucination	4.0	The highlight claims the novel is considered a 'cornerstone of American sci-fi literature,' but the source text does not support this. While some reviews praise its foresight, many others describe it as an early, flawed, or non-essential work compared to Crichton's other novels, making this claim an unsupported overstatement.
Level: *relation* **Size:** *large* **Sys.:** *e2e* **Book:** *Until I Find You* **Theme:** *author*	John Irving draws from his life as a child of a single mother and his passion for organ music.	hallucination	4.0	The source text confirms that Irving drew on his personal experience of not knowing his biological father. However, while organ music is a major theme in the novel, the source does not state that this is a personal passion of the author. The passion for organ music is attributed to a character in the book, not to Irving himself.

Table 6: Example outputs from the faithfulness evaluation. Each row shows the generated highlight, the judge's verdict (*none* = faithful, *hallucination*, *contradiction*, or *both*), severity, and the judge's rationale.

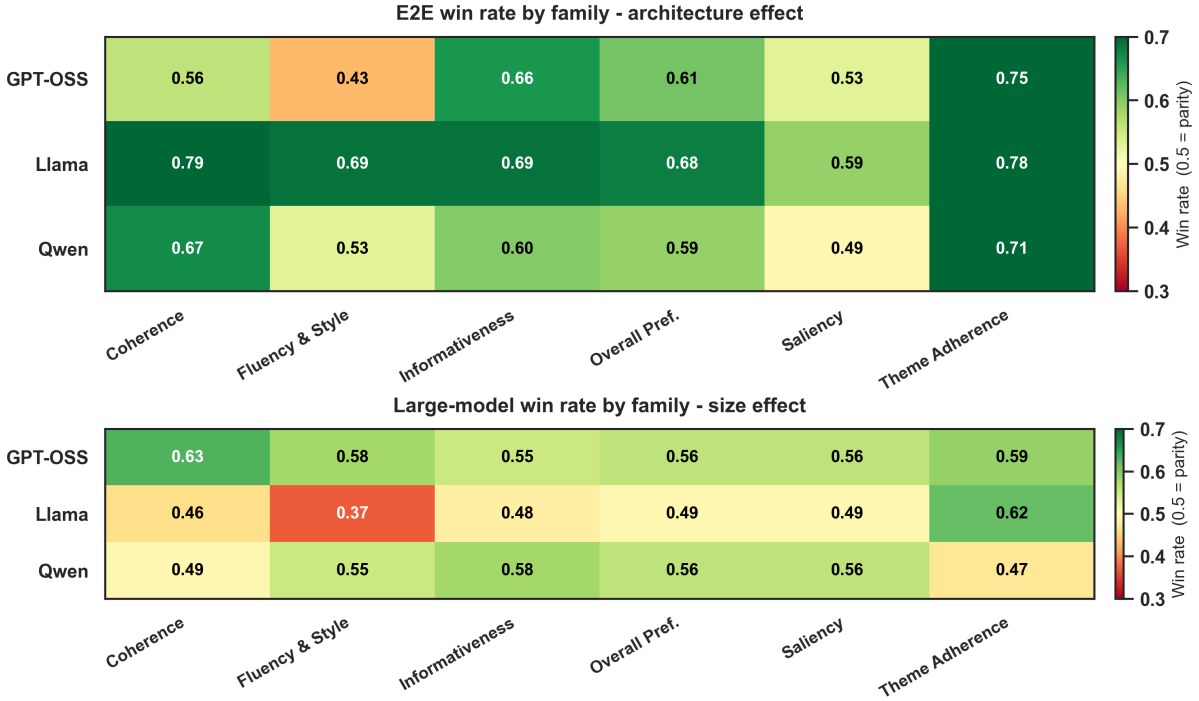

Figure 4: **Top (architecture effect):** E2E win rate by model family and evaluation criterion. Green cells indicate that E2E is preferred on that criterion within that family; red cells indicate that PIPE is preferred.

Bottom (size effect): Large-model win rate by family and criterion. Green cells indicate that the larger model is preferred within a family; red cells indicate that the smaller model is preferred. This panel should be interpreted with caution: parameter ratios (4×–9×) and context windows (32K vs. 128K) are not matched across families.

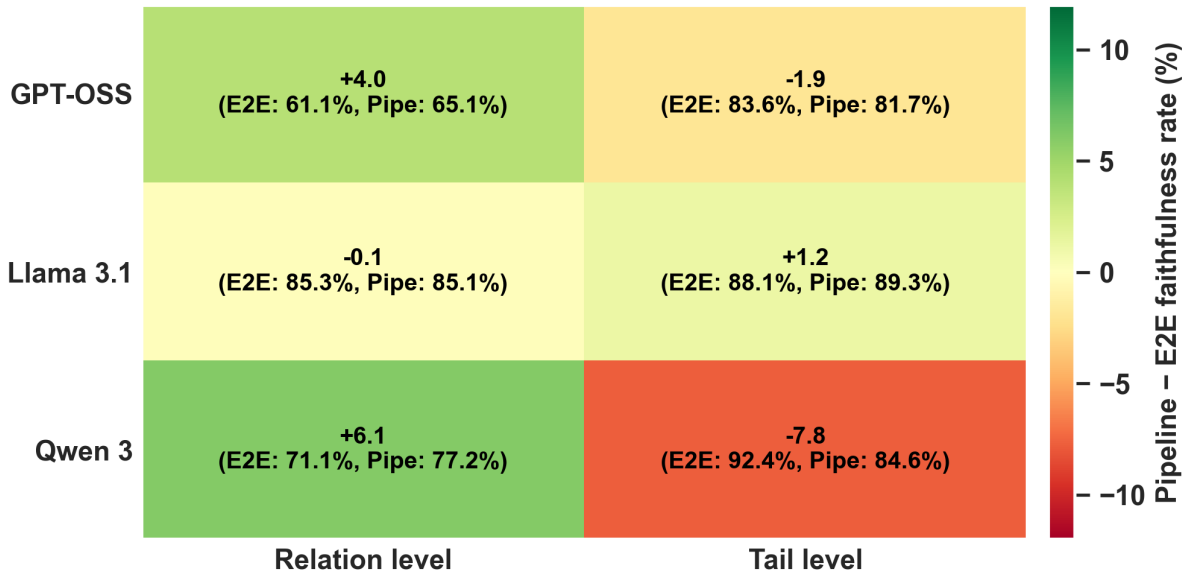

Figure 5: Pipeline minus E2E faithful rate by model family and content level. Green indicates pipeline is more faithful; red indicates E2E is more faithful.