RetroEval 2026

Proceedings of the 1st Symposium on Natural Language Generation Evaluations

Aberdeen, Scotland, United Kingdom

June 1-2, 2026

The RetroEval organizers gratefully acknowledge the support from the following sponsors.

Sponsored by

Supported by

Endorsed by

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-436-1

Preface

We are excited to present the Proceedings of the 1st Symposium on Natural Language Generation Evaluations (RetroEval 2026). This special symposium celebrates the career and accomplishments of Prof. Ehud Reiter (University of Aberdeen) by looking at evaluation practices in the field of Natural Language Generation past, present, and future.

The first RetroEval symposium will take place from 1-2 June, in Aberdeen, United Kingdom, endorsed by the Association of Computational Linguistics Special Interest Group in Generation (ACL SIGGEN). We thank the local organisation team, including Arpita Gado, Mengxuan Sun, Yujun Wang, and Jakub Zbrzezny; the symposium would not have been possible without their efforts. We also thank Debbie Meharg, the head of the Department of Computing Science at the University of Aberdeen, for arranging financial support from the department for this event; and we thank Chrissy Sanachan from the Scottish Informatics and Computer Science Alliance (SICSA) for logistical support.

In addition to a number of papers that were selected following an open call and a light review process, we are excited to present three keynotes, which discuss pertinent issues in evaluation within Natural Language Processing:

- Sina Zarrieß, Universität Bielefeld, Germany
- Beatrice Alex, Heriot-Watt University, United Kingdom
- Albert Gatt, Universiteit Utrecht, the Netherlands

Taken jointly, the contributions to RetroEval illustrate how Ehud has helped to make our field what it is today, both in academia and in industry. They highlight Ehud's role as an enthusiastic inspirator and a kind mentor, and the role he has played as a builder and evaluator of NLG systems. Last but not least, they highlight the way in which Ehud has made the NLG community, and the NLP community more broadly, aware of the many pitfalls that can stand in the way of truly scientific system evaluation. It is to his work on evaluation that this symposium is devoted.

Your RetroEval 2026 Organizers,
Saad Mahamood, David M. Howcroft, Kees van Deemter, Adarsa Sivaprasad, Barkavi Sundararajan, Jose Maria Alonso-Moral, and Simone Balloccu

Programme Committee

Jose Maria Alonso-Moral, CiTIUS, University of Santiago de Compostela
Simone Balloccu, TU Darmstadt
Alberto Bugarín-Diz, CiTIUS, University of Santiago de Compostela
Silvia Casola, Ludwig Maximilian University of Munich
Ondřej Dušek, Charles University
Albert Gatt, Universiteit Utrecht
Dimitra Gkatzia, Napier University
David M. Howcroft, University of Aberdeen
Mateusz Lango, Charles University
Chenghua Lin, University of Manchester
Saad Mahamood, Shopware
Patrícia Schmidtová, Charles University
Kees van Deemter, Universiteit Utrecht
Mengxuan Sun, University of Aberdeen
Craig Thomson, Dublin City University

Table of Contents

Decomposition Does Not Help: Evidence from Semantic Clustering in LLM-based Causal Graph Discovery

Nikolay Babakov, Alberto Bugarín-Diz
CiTIUS-Centro Singular de Investigación
en Tecnoloxías Intelixentes
Universidade de Santiago de Compostela
{nikolay.babakov, alberto.bugarin.diz}@usc.es

Abstract

Recent advances in large language models (LLMs) have enabled their application to non-traditional tasks such as causal graph construction, a key component of reasoning frameworks, including Bayesian Networks. The most effective existing approaches rely on direct prompting, where an LLM generates a complete graph from a full set of variables in a single step. However, the performance of such methods degrades as the number of graph nodes increases. To address this limitation, we explore a divide-and-conquer alternative based on semantic clustering. Node representations are first embedded and clustered, after which subgraphs are constructed independently for each cluster using LLM prompting. The resulting subgraphs are then merged pairwise into a global graph.

Contrary to our expectations, this approach leads to a substantial degradation in performance compared to direct prompting baselines, as measured by Structural Hamming Distance (SHD). We attribute this to the misalignment between semantic similarity and causal structure, as well as error propagation during subgraph merging. We report these negative results to highlight the limitations of decomposition strategies in LLM-based causal graphs construction.

1 Introduction

The growing capabilities of large language models (LLMs) have expanded their applications into domains not traditionally associated with natural language processing, including education (Kasneci et al., 2023) and programming (Guo et al., 2024). One such emerging application is causal graph (CG) construction, a key component of probabilistic reasoning frameworks like Bayesian Networks (BNs) (Koller, 2009). Causal graph discovery (CGD) has traditionally been addressed either through data-driven structure learning algorithms (Kitson et al., 2023) or through expert elicitation, where domain knowledge is used to define causal relationships (Nyberg et al., 2022) . In contrast to these approaches, recent work demonstrates that LLMs can infer causal structure directly from textual descriptions of variables, effectively enabling CGD without explicit data or expert input (Wang et al., 2024; Chen et al., 2024; Wan et al., 2024).

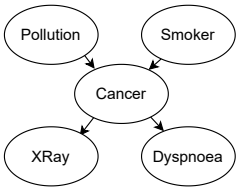

Figure 1: Causal Graph of the BN related to the lung cancer problem (Korb and Nicholson, 2010).

CG is a directed acyclic graph (DAG) that illustrates variables and their causal dependencies. Consider the example shown in Figure 1, which depicts a CG of a simple BN (Korb and Nicholson, 2010). This BN models a hypothetical scenario involving potential causes (e.g., Pollution and Smoker) and effects (e.g., X-Ray results and Dyspnoea) of Lung Cancer.

The growing attention to LLM-based causal graph construction has led to the development of dedicated evaluation benchmarks, such as CausalGraphBench introduced by Babakov et al. (2025b), enabling systematic comparisons of different approaches. In addition to establishing a unified evaluation setting, this work demonstrates that simple direct prompting strategies, where an LLM is asked to reconstruct a full graph from a list of nodes in a single step, perform on par with more elaborate multi-step methods that incorporate additional reasoning or constraints. At the same time, all evaluated approaches exhibit a substantial decline in

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 1–7

performance as graph size increases, identifying scalability as a central challenge in LLM-based CGD (Babakov et al., 2025b).

This limitation motivates the exploration of decomposition strategies that break the task into smaller, more manageable subproblems. In this work, a divide-and-conquer approach based on semantic node clustering is investigated. Given textual descriptions of variables, nodes are embedded into a vector space, dimensionality is reduced, and clusters are formed using a pipeline inspired by topic modelling techniques. Each cluster is then processed independently by an LLM to construct a subgraph, after which the resulting subgraphs are iteratively merged in a pairwise manner to produce a global causal graph.

The approach is evaluated using the aforementioned CausalGraphBench benchmark. Contrary to expectations, decomposition via semantic clustering results in substantial performance degradation compared to direct prompting baselines. These results suggest that semantic similarity between node descriptions does not align well with underlying causal structure and that errors introduced during subgraph construction and merging accumulate in the final graph.

By reporting these negative findings, this work aims to contribute to a better understanding of the limitations of decomposition strategies in LLM-based structured prediction tasks and to inform future research on scalable approaches to causal graph construction.

2 Related works

LLMs have been explored for a variety of graph-related tasks, including connectivity, cycle detection, shortest path, and topological ordering (Wang et al., 2024; Chen et al., 2024).

In the context of causal graph construction, existing approaches can be broadly divided into two categories. The first combines LLMs with traditional data-driven methods (Ban et al., 2023a; Long et al., 2023a). The second category relies on LLMs to construct causal graphs directly, with methods differing mainly in how they query the model. One group of LLM-only methods makes exhaustive queries, like all possible pairs, triplets, or other combinations of nodes, resulting in a significant number of queries necessary for reconstruction of one CG (Cohrs et al., 2024; Zhang et al., 2024; Vashishtha et al., 2023; Long et al., 2023b; Kıcıman et al., 2023; Feng et al., 2024; Darvariu et al., 2024; Zhou et al., 2024). In contrast, minimal-query approaches aim to construct the full graph with fewer interactions while preserving a more global view of the structure, including iterative graph construction, structured multi-step prompting, and ensemble-style aggregation of independently generated graphs (Jiralerspong et al., 2024; Ban et al., 2023b; Babakov et al., 2025a; Zhang et al., 2025).

3 Experimental setup

3.1 Dataset

The experiments are conducted using the CausalGraphBench benchmark (Babakov et al., 2025b), which comprises 35 causal graphs derived from both publicly available repositories and academic papers. The benchmark includes graphs of varying sizes, with a median of 16 nodes and 21 edges. Each graph is accompanied by structured metadata, including a textual description of the graph's purpose, the associated knowledge domain, a dictionary of node descriptions clarifying variable semantics, and the ground-truth graph structure.

3.2 Methodology of the experiments

Two approaches to CG construction are compared: a baseline method and a cluster-based decomposition method.

The baseline follows a direct zero-shot prompting strategy, where the LLM is provided with the full list of clearly defined node names and asked to generate the complete causal graph in a single step.

The cluster-based method introduces a multi-step pipeline to decompose the task into smaller subproblems, motivated by scalability challenges observed in prior work. The approach follows a subset of steps inspired by the BERTopic framework (Grootendorst, 2022). First, node descriptions are embedded into a vector space using sentence embedding models; two variants are considered: MiniLM[1] and Gemma[2]. Second, dimensionality reduction is applied using UMAP (McInnes et al., 2018; McInnes et al., 2018). Third, clustering is performed using HDBSCAN (McInnes et al., 2017). The hyperparameters for these stages are adopted based on recommendations from BERTopic: minimum cluster size of 2, UMAP with

[1] huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[2] huggingface.co/google/embeddinggemma-300m

15 neighbours (capped at the number of nodes minus one), 2 output components, cosine distance metric, and HDBSCAN with Euclidean distance and 'excess of mass' cluster selection.

After clustering, each cluster is processed independently by the LLM using the same prompting strategy as in the baseline, producing subgraphs. Clusters containing a single node are preserved without modification. The resulting subgraphs are then combined through a pairwise merging procedure: all pairs of subgraphs (or a restricted subset based on nearest cluster centroids) are presented to the LLM, which is queried to infer cross-cluster connections. The final causal graph is constructed by aggregating intra-cluster subgraphs and inter-cluster edges. Two merging strategies are explored: considering all possible subgraph pairs and restricting merging to the top-k nearest clusters (with $k \in 3, 5$) based on cosine distance between cluster centroids. The specific prompts used for LLM querying are shown in the Appendix.

Experiments are conducted using both proprietary and open-source LLMs. The proprietary models include GPT-5.4 (2026-03-05)[3] and GPT-5.2 (2025-12-11)[4]. Open-source models include GPT-OSS-120b (OpenAI, 2025) and GLM-5 (GLM-5-Team et al., 2026).

3.3 Evaluation

We evaluate the quality of the LLM-generated CGs using Structural Hamming Distance (SHD), a widely used measure for evaluating graph discovery algorithms (Tsamardinos et al., 2006). Lower SHD values indicate higher-quality graphs. SHD is calculated as the total number of operations (addition, removal, or reversal of edge directions) required to transform the generated graph into the target graph. Incorrectly oriented edges, where the cause and effect are reversed, are penalised as two errors. To make comparisons across CGs of varying sizes more meaningful, we report SHD normalised by the node count in the actual CG. We used causal discovery toolbox[5] for SHD calculations.

4 Contamination Analysis

LLMs may have prior exposure to some CGs, leading to artificially improved performance (Tu et al., 2023; Tamkin et al., 2021; Sainz et al., 2023). To mitigate this, a contamination detection procedure based on Babakov et al. (2025a) is applied. Each model is first prompted to reconstruct the set of nodes of a CG using only its metadata (paper and, when available, source URL). Exact recovery of nodes in both number and semantic meaning is treated as a signal of potential contamination.

CG name	LLM	SHD/nodes
alarm	GPT-5.4	0.41
cancer	GLM-5	0
	GPT-5.2	0.2
	GPT-5.4	0
	GPT-OSS-120b	0
coma	GPT-5.4	0
covid	GPT-5.4	0.2
sachs	GLM-5	0.73
	GPT-5.4	0.73

Table 1: Results of the second step of contamination analysis - for CGs that are potentially contaminated (i.e., LLM can produce an accurate list of nodes relying solely on paper name or URL), LLM is also queried to generate a corresponding CG.

Using this procedure, such signals are observed for several models, including GLM-5 (*sachs*, *cancer*), GPT-5.2 (*cancer*), GPT-5.4 (*sachs*, *cancer*, *alarm*, *covid*, *coma*), and GPT-OSS-120b (*cancer*). These CGs are further tested by prompting the corresponding models to reconstruct their structure from the generated nodes. The resulting graphs are compared to the ground truth using normalized SHD (Table 1).

The results show that, although multiple CGs have perfectly reconstructed node sets, only a subset can be accurately recovered at the structural level. In particular, *cancer* and *coma* are reconstructed with a zero error by at least one model, indicating strong prior exposure. These two CGs are therefore excluded from further experiments to ensure fair evaluation.

5 Experimental Results

The experiments are conducted by applying both the baseline and the cluster-based methods to all CGs that were not excluded during the contamination analysis (Section 4). The cluster-based method is evaluated with two different embedding models (Section 3.2).

Table 2 presents the results for the all-vs-all merging strategy, where the LLM is queried to merge every possible pair of subgraphs. The results show a substantial increase in SHD compared to

[3] openai.com/index/introducing-gpt-5-4/
[4] openai.com/index/introducing-gpt-5-2/
[5] github.com/ElementAI/causal_discovery_toolbox

Mehtod/LLM	GPT-5.4	GPT-5.2	GPT-OSS-120b	GLM-5
Baseline	1.71	1.67	1.68	1.57
Cluster (MiniLM)	3.04	3.52	3.26	3.09
Cluster (Gemma)	2.74	3.10	3.00	2.81

Table 2: SHD normalized by nodes count for baseline experiments and cluster-based experiments, conditioned on clusters from MiniLM and Gemma encoder models.

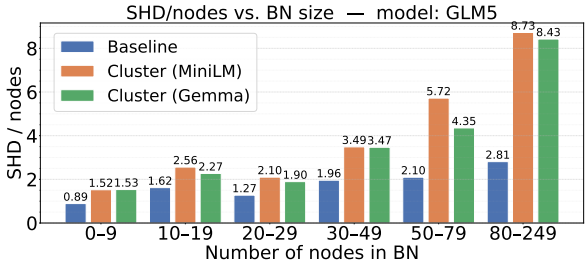

Figure 2: Normalised SHD for CGs of different sizes resulting from application of baslein and cluster-based method with GLM-5 model.

the baseline across all models, indicating that this approach is ineffective for CGD. This trend is further illustrated in Figure 2, where the degradation in performance is observed even for small graphs. As the number of nodes increases, the negative effect becomes significantly more pronounced, highlighting the poor scalability of the cluster-based decomposition under exhaustive merging.

An alternative merging strategy, in which only the top-3 or top-5 nearest cluster pairs (based on centroid proximity) are considered instead of all possible pairs, also fails to yield meaningful improvements. As shown in Appendix Table 3, restricting the merging process does not significantly reduce SHD, indicating that limiting inter-cluster interactions is insufficient to overcome the limitations of the cluster-based approach.

6 Discussion

The contamination analysis reveals that, although several widely known CGs have node sets that are clearly recognised by the models, this knowledge does not reliably translate into accurate reconstruction of the underlying graph structure. Even when node names are perfectly recovered, the corresponding causal relationships are often not. This suggests that causal graph reconstruction is not a well-internalised capability of LLMs. While some models (e.g., GPT-5.x) support multimodal inputs, this does not imply effective retention or use of structured graph knowledge. These findings support the validity of the CausalGraphBench benchmark, as the task does not reduce to memorisation, and highlight the importance of contamination checks for reliable evaluation.

The main experimental results demonstrate that the proposed clustering-based decomposition does not improve causal graph construction and, in fact, leads to substantial performance degradation. While the approach was intended to simplify the task and improve scalability for larger graphs, the opposite effect is observed: errors increase significantly as graph size grows. A likely explanation is that semantic clustering of node descriptions does not correspond to the underlying causal structure. As a result, important cross-cluster dependencies are lost, and the subsequent merging process introduces additional inconsistencies, ultimately leading to higher reconstruction error compared to direct prompting.

7 Conclusion

This work investigates a clustering-based decomposition strategy for LLM-driven causal graph construction and finds that, contrary to expectations, it consistently degrades performance compared to direct prompting. The results show that semantic node grouping does not align with the causal structure, and decomposition introduces errors that accumulate during graph merging, particularly in larger graphs. These findings highlight the limitations of naïve divide-and-conquer approaches for structured reasoning with LLMs and suggest that preserving global context is critical for accurate causal graph discovery.

Acknowledgments

The first author would like to express sincere gratitude to Ehud Reiter for his guidance and co-supervision during the PhD studies at the University of Santiago de Compostela. This work is a direct continuation of prior joint research on the application of LLMs to causal graph construction, and several ideas explored in this paper, including

the contamination analysis procedure, originated from discussions within that collaboration.

This paper is part of the R+D+i projects PID2023-149549NB-I00 and PID2023-149959OA-I00 funded by MCIN/AEI/10.13039/501100011033/ and by "ERDF a way of making Europe". The support of the Galician Ministry for Education, Universities and Professional Training and "ERDF A way of making Europe" is also acknowledged through the grant "Centro de investigación de Galicia accreditation 2024-2027 ED431G-2023/04".

References

Nikolay Babakov, Ehud Reiter, and Alberto Bugarín. 2025a. Scalability of Bayesian network structure elicitation with large language models: a novel methodology and comparative analysis. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10685–10711, Abu Dhabi, UAE. Association for Computational Linguistics.

Nikolay Babakov, Ehud Reiter, and Alberto Bugarín-Diz. 2025b. CausalGraphBench: a benchmark for evaluating language models capabilities of causal graph discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 240–258, Vienna, Austria. Association for Computational Linguistics.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. 2023a. Causal structure learning supervised by Large Language Model. *arXiv preprint arXiv:2311.11689*.

Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. 2023b. From query tools to causal architects: Harnessing Large Language Models for advanced causal discovery from data. *arXiv preprint arXiv:2306.16902*.

Sirui Chen, Mengying Xu, Kun Wang, Xingyu Zeng, Rui Zhao, Shengjie Zhao, and Chaochao Lu. 2024. CLEAR: Can language models really understand causal graphs? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6247–6265, Miami, Florida, USA.

Kai-Hendrik Cohrs, Gherardo Varando, Emiliano Diaz, Vasileios Sitokonstantinou, and Gustau Camps-Valls. 2024. Large Language Models for constrained-based causal discovery. *arXiv preprint arXiv:2406.07378*.

Victor-Alexandru Darvariu, Stephen Hailes, and Mirco Musolesi. 2024. Large Language Models are effective priors for causal graph discovery. *arXiv preprint arXiv:2405.13551*.

Tao Feng, Lizhen Qu, Niket Tandon, Zhuang Li, Xiaoxi Kang, and Gholamreza Haffari. 2024. From pretraining corpora to Large Language Models: What factors influence LLM performance in causal discovery tasks? *arXiv preprint arXiv:2407.19638*.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, and 168 others. 2026. Glm-5: from vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, and 1 others. 2024. Deepseek-coder: When the Large Language Model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient causal graph discovery using Large Language Models. *arXiv preprint arXiv:2402.01207*.

Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, and 4 others. 2023. ChatGPT for good? on opportunities and challenges of Large Language Models for education. *Learning and Individual Differences*, 103:102274.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2023. Causal reasoning and Large Language Models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*.

Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. 2023. A survey of Bayesian Network structure learning. *Artificial Intelligence Review*, pages 1–94.

Daphane Koller. 2009. Probabilistic graphical models: Principles and techniques.

Kevin B Korb and Ann E Nicholson. 2010. *Bayesian artificial intelligence*. CRC press.

Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. 2023a. Causal discovery with language models as imperfect experts. *arXiv preprint arXiv:2307.02390*.

Stephanie Long, Tibor Schuster, and Alexandre Piché. 2023b. Can Large Language Models build causal graphs? *arXiv preprint arXiv:2303.05279*.

L. McInnes, J. Healy, and J. Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861.

Erik P. Nyberg, Ann E. Nicholson, Kevin B. Korb, Michael Wybrow, Ingrid Zukerman, Steven Mascaro, Shreshth Thakur, Abraham Oshni Alvandi, Jeff Riley, Ross Pearson, Shane Morris, Matthieu Herrmann, A.K.M. Azad, Fergus Bolger, Ulrike Hahn, and David Lagnado. 2022. BARD: A structured technique for group elicitation of Bayesian Networks to support analytic reasoning. *Risk Analysis*, 42(6):1155–1178.

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

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. 2021. Understanding the capabilities, limitations, and societal impact of Large Language Models. *arXiv preprint arXiv:2102.02503*.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing Bayesian Network structure learning algorithm. *Machine learning*, 65:31–78.

Ruibo Tu, Chao Ma, and Cheng Zhang. 2023. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis. *arXiv preprint arXiv:2301.13819*.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. 2023. Causal inference using LLM-guided discovery. *arXiv preprint arXiv:2310.15117*.

Guangya Wan, Yuqi Wu, Mengxuan Hu, Zhixuan Chu, and Sheng Li. 2024. Bridging causal discovery and Large Language Models: A comprehensive survey of integrative approaches and future directions. *arXiv preprint arXiv:2402.11068*.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36.

Yinghuan Zhang, Yufei Zhang, Parisa Kordjamshidi, and Zijun Cui. 2025. Bayesian network structure discovery using large language models. *arXiv preprint arXiv:2511.00574*.

Yuzhe Zhang, Yipeng Zhang, Yidong Gan, Lina Yao, and Chen Wang. 2024. Causal graph discovery with retrieval-augmented generation based Large Language Models. *arXiv preprint arXiv:2402.15301*.

Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Causalbench: A comprehensive benchmark for causal learning capability of Large Language Models. *arXiv preprint arXiv:2404.06349*.

A Appendix

MiniLM				
Method/LLM	GPT-5.4	GPT-5.2	GPT-OSS-120b	GLM-5
Baseline (top-3)	1.95	1.90	1.87	1.80
Cluster (top-3)	2.52	2.71	2.54	2.45
Baseline (top-5)	2.03	2.00	2.02	2.02
Cluster (top-5)	3.19	3.51	3.36	3.16

Gemma				
Method/LLM	GPT-5.4	GPT-5.2	GPT-OSS-120b	GLM-5
Baseline (top-3)	1.84	1.79	1.76	1.67
Cluster (top-3)	2.16	2.25	2.37	2.17
Baseline (top-5)	1.74	1.71	1.75	1.71
Cluster (top-5)	2.39	2.67	2.66	2.54

Table 3: SHD/nodes (Structural Hamming Distance normalised by node count; lower is better) for the cluster-based approach under two inter-cluster joining strategies (top-3 and top-5 nearest-neighbour pairs) and two node encoders (MiniLM, Gemma). For each joining mode, the baseline is averaged over the same subset of CGs that satisfies the minimum cluster-count requirement for that mode, ensuring a fair comparison, e.g., for top-3 only CGs with 4 or more clusters from the corresponding encoder model are included.

A.1 Baseline prompt

This prompt was used both for a direct baseline LLM query and for querying a subgraph with a node count of more than 3.

You are an expert on {domain}. You are constructing the Bayesian Network aimed to fulfill the following task: {task}. To construct the Bayesian Network you need to investigate the cause-and-effect relationships between the following variables in your area of expertise: {variables}. Based on the meaning of variables, analyze the cause-and-effect relationships between them. Please give the results as a directed graph network. Make sure that each edge represent a direct causality between the two variables.

Return valid JSON-list of the following format: {{ "result": [[from node (A), to node(B)], # (meaning that there is a direct causal effect from node A to node B) [from node (F), to node(E))] # (meaning that there is a direct causal effect from node F to node E) [from node (D), to node(G))] # (meaning that there is a direct causal effect from node D to node G) ...] }} """

A.2 Subgraphs pairing prompt

You are an expert on {domain}. You are constructing a Bayesian Network aimed to fulfill the following task: {task}.

You are given two subgraphs of this Bayesian Network. Each subgraph is described by its nodes and the directed edges already established within it.

Subgraph 1: - Nodes: {nodes_1} - Edges: {edges_1}

Subgraph 2: - Nodes: {nodes_2} - Edges: {edges_2}

Your task is to identify direct causal relationships that exist **between** the two subgraphs — that is, edges from a node in one subgraph to a node in the other subgraph. Do NOT propose edges between nodes that are both within the same subgraph.

Return valid JSON of the following format: {{ "result": [["A", "B"], ["C", "D"]] }}

Where each pair ["A", "B"] means there is a direct causal effect from node A to node B, and A and B belong to different subgraphs.

If there are no causal relationships between the two subgraphs, return: {{ "result": [] }}

NLG Evaluation: Past, Present, Future

Ehud Reiter
Dept of Computing Science
University of Aberdeen
Aberdeen, UK
e.reiter@abdn.ac.uk

Abstract

Natural Language Generation (NLG) evaluation has changed dramatically since 1990, and will continue to evolve in the future. In 1990, when NLG had close ties to linguistics, there was very little formal experimental evaluation in the modern sense. In 2026, when NLG is closely linked to machine learning, experimental evaluation is expected and indeed fundamental to research. Many evaluation techniques were developed over this period, including most recently LLM-as-Judge. I expect NLG evaluation will continue to evolve in the future. In particular, impact, qualitative, and safety evaluation will become more important as large numbers of people routinely use NLG technology.

1 Introduction

The evaluation of Natural Language Generation (NLG) systems has changed dramatically over my career. In 1990, when I got my PhD in NLG, most NLG research papers did not include a quantitative experimental evaluation of a research question. By 2026, NLG research papers are expected to include structured experimental evaluations of hypotheses, although the quality and validity of these evaluations is variable. I expect that by 2036, impact, safety, and qualitative evaluations will be much more important, because NLG technology will be widely used by large numbers of people. Table 1 summarises my view of NLG evaluation at different points in time.

2 NLG Evaluation in the Past

2.1 1990: Little quantitative experimental evaluation

The International NLG (INLG) conference in 1990 had 25 papers. *None* of them included a structured quantitative hypothesis test. Instead, these papers mostly presented an algorithm, technique, resource, or system, and justified it on engineering or linguistic criteria. For example, McCoy et al. (1990) proposed combining tree-adjoining and systemic grammars, and justified this by arguing that their approach did a better job of handling long-distance dependencies (linguistics) and also makes it easier to build grammars (engineering). Their argument was qualitative, no numbers were given.

In the broader NLP world, the speech recognition community had adapted the idea of quantitative comparisons of the performance of systems (Waibel and Lee, 1990), but this was unusual in the rest of NLP. Perhaps the most important NLP paper in 1990 was Brown et al. (1990), which introduced statistical machine translation, but even it did not provide quantitative comparisons of the sort we expect in 2026.

2.2 2000: Wide range of evaluation techniques

INLG in 2000 had 38 papers, and these included many different kinds of evaluation, as well as one of the first paper that was *about* evaluation (Bangalore et al., 2000). Types of evaluation included

- Human evaluation (Cheng and Mellish, 2000)
- Metric-based evaluation (Minnen et al., 2000)
- Task-based evaluation (Carenini, 2000)

There were also papers which continued to assess their contribution using engineering or linguistic arguments, as in 1990.

In short, by 2000 experimental evaluations was recognised as being important. However there were no widely accepted standard evaluation techniques in NLG.

A similar mix was seen at larger NLP events such as ACL. Evaluation was clearly regarded as important, but many techniques were being tried. The broader NLP community focused more on metric-based evaluation, including Gildea and Jurafsky

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 8–15

year	NLG evaluation	example paper and its evaluation
1990	non-quantitative evaluation, often using linguistic or engineering arguments	McCoy et al. (1990): qualitative argument that their grammatical approach handles long-distance dependencies better
2000	wide mix of different techniques, including metrics, human ratings, and task-based	Cheng and Mellish (2000): use human ratings to evaluate different ways of expressing causal and temporal relationships
2010	standardised evaluation techniques and shared tasks based on these	Belz and Kow (2010): results of GREC shared task on generating referring expressions
2020	research on evaluation becomes an important research area	Howcroft et al. (2020): gives recommendations for reporting human evaluations, based on meta-analysis of published evaluations
2026	LLM-as-Judge, annotation by human experts, safety, interdisciplinary	Bean et al. (2026): use medical evaluation techniques to assess system that answers health queries
2036	impact, qualitative, safety evaluation	*not yet written*

Table 1: NLG evaluation over the years

(2000), which won a Test of Time award. However ACL in this period also included papers reporting complex task-based evaluations (Mani et al., 1999; Reiter et al., 2001).

2.3 2010: Shared tasks and standard evaluations

INLG in 2010 had 37 papers, many of which were shared task submissions. Shared tasks (such as the GREC challenge for generating referring expressions (Belz and Kow, 2010)) had become an accepted part of NLG as well as NLP research, and used metrics and/or human evaluations to evaluate the performance of submissions. Some papers also began to describe evaluations in considerable detail (Murray et al., 2010).

The wider NLP community had embraced ngram-based metrics for evaluation of text production, and the BLEU and ROUGE metrics had effectively become standards. Papers in machine translation were expected to use BLEU, and papers in summarisation were expected to use ROUGE.

Human evaluation had become unusual in ACL conferences, although the annual WMT shared task continued to use it. The NLG community, however, insisted on using human evaluations, and could point to papers which suggested that metrics were not reliable in NLG (Reiter and Belz, 2009). When doing human evaluations, most researchers either used Likert scales or asked subjects to rank a set of texts by a quality criteria; these became standard techniques for human evaluation of generated texts.

2.4 2020: Evaluation is important research area

INLG in 2020 had 46 papers (it has not seen the exponential growth that ACL has had in recent years). Perhaps the most notable change compared to 2010 was that evaluation has become a very important part of the community's research agenda. Indeed both of the INLG2020 best papers were about evaluation (Belz et al., 2020; Dušek and Kasner, 2020), and there were several other papers about evaluation methodology (Howcroft et al., 2020; Thomson and Reiter, 2020) in INLG2020.

The wider NLP community also placed increasing importance on evaluation as an important research theme. For example the ACL 2020 best paper was about testing (Ribeiro et al., 2020), and one of the two honourable mention papers was about evaluation (Mathur et al., 2020).

In short, evaluation was now not just something which researchers had to do, but also an important research topic in its own right.

3 NLG Evaluation in 2026

Reiter (2025a) summarised NLG evaluation in 2025, including links to papers that gave best practice suggestions. Large language model (LLM) technology had become widespread and this had changed NLG evaluation and introduced new challenges.

3.1 Evaluation challenges from LLMs

There are many challenges in evaluating LLMs, including the following.

Higher quality generated texts: Texts produced by LLMs are usually higher quality than texts produced by previous technologies (rule-based, LSTM), and can in some cases be human quality, or even better-than-human. This means that many traditional evaluation techniques, such as metrics that compare generated texts against human-written reference texts, no longer work well. If we expect a generated text to be better than human, then evaluating it by comparing it to a human-written reference text does not make sense.

Semantic and pragmatic evaluations: Texts produced by LLMs are almost always fluent and readable, so evaluating readability is less useful. Instead, there is more emphasis on evaluating semantic and pragmatic quality criteria (Reiter, 2025a), such as accuracy/hallucinations, omissions, and contextual appropriateness.

Data contamination: Since LLMs are trained on the Internet, an evaluation that uses Internet data may not mean much, since the LLM may have memorised the test data (Balloccu et al., 2024).

Worst-case and safety evaluation: The growing real-world usage of LLMs in safety-critical contexts such as medicine (where flawed texts could harm patients) means that we need techniques that evaluate 'worst-case' performance of LLMs (Reiter, 2025a). If a medical LLM gives good output in 99.9% of cases but harmful output in 0.1% of cases, this is not acceptable.

Interdisciplinary interest and usage: The growing real-world usage of LLMs means that other disciplines (such as medicine and law) want LLM-based NLG systems to be evaluated using their methodologies and expectations (Duggan et al., 2025).

3.2 Changes in NLG evaluation

The above challenges have changed the way NLG is evaluated.

LLM as Judge: The above problems have stimulated interest in reference-free metrics which work for semantic and pragmatic quality criteria, including in particular using LLMs to evaluate the quality of texts produced by other LLMs (Gao et al., 2025); this is called *LLM as Judge*. This seems to work well in some cases but not others; unfortunately many researchers use LLM evaluators without checking that they are effective in their use case.

Human evaluation using expert annotations: Human evaluations in NLG have traditionally used Likert-type rating scales. This seems to work less well when evaluating semantic and pragmatic problems in high-quality LLM texts, especially with crowdworkers (who may cheat by using LLMs to do the evaluation task (Asher et al., 2026)). We are seeing more human evaluations that instead ask knowledgable people to annotate specific problems in a generated text (Thomson et al., 2023).

Private test data: In 2020, test data sets were typically published (e.g., on GitHub repos), which made replication easier. But in 2026, data contamination concerns mean that test data is sometimes not published or shared.

Safety evaluations: Many techniques have been proposed for safety evaluation. This area is heavily influenced by cyber security, and includes techniques for risk analysis (such as red teaming), risk mitigation (e.g., monitoring), and risk governance (such as incident reporting) (Bengio et al., 2026).

Interdisciplinary evaluations: High-quality evaluations of NLG systems are appearing in other fields, notably medicine, that use medical evaluation techniques such as randomised controlled trials (which are very rare in the NLP literature (Reiter, 2025b)). Sometimes these give different results from classical NLP evaluation, which raises important questions about the best way to evaluate NLG .

3.3 Ongoing challenges for NLG evaluation

The new evaluation techniques described above are being adopted by many researchers and help in addressing some of the new evaluation challenges of LLMs. But there are many problems and concerns that still need to be addressed. These include

- *Experimental rigour:* Unfortunately, many experiments are poorly designed, poorly executed, or distorted by bugs (Thomson et al., 2024).
- *Replicability:* Many experiments cannot be replicated, in part because their authors do not support replication (Belz et al., 2023).
- *Construct validity:* Many evaluation techniques, especially benchmarks, do not measure what they claim to measure (Bean et al., 2025).

- *Cheating:* LLMs engage in behaviour such as reward hacking (Arx et al., 2025), which is essentially cheating. Asadi et al. (2026) show that LLMs can get very high benchmark scores even when input data is withheld, by picking up on subtle clues in the wording of questions in the benchmark.

- *Commercial bias and incentives:* A lot of evaluation research and development is funded by AI companies such as OpenAI, who have an interest in ensuring that their systems do well on these evaluations (Cheng et al., 2025).

- *Evolving benchmarks:* New evaluation benchmarks are constantly being proposed, and existing benchmarks often become saturated (Akhtar et al., 2026) and hence useless. It is difficult for many researchers to stay up-to-date on the best benchmark to use.

A generic challenge is that the research culture in NLP is often not very supportive of high quality evaluation. Many people feel pressure to publish large numbers of papers, and reviewers often show limited interest in quality of data sets, validity of evaluation metrics, experimental rigour, etc. This encourages researchers to conduct 'quick and dirty' evaluations.

4 NLG Evaluations in the Future

What will NLG evaluation be like in ten years time (2036)? The above challenges will hopefully be addressed, but more generally we also need to go beyond measuring performance on a test set, which dominates NLG and NLP evaluation in 2026. If we care about how our technology affects the real world, we need to do more of the following:

- Directly measure the real-world **impact** of NLG systems.

- Use **qualitative** techniques to get insights about the effectiveness of our techniques in messy and complex real-world contexts.

- Analyse what happens in worst-case or adversarial contexts, especially for **safety** criteria.

These techniques will help ensure that evaluation is relevant and meaningful in a future world where NLG is a widely-used technology.

Note that impact, qualitative, and safety evaluations are not new, they are already being done in 2026 to a limited degree in NLG; they are much more common in Medicine, perhaps because medical research has had real-world consequences for decades or indeed centuries. So the challenge for the NLG community is to embrace these types of evaluation and learn how to do them well in an NLG context.

The spread of more types of NLG evaluation may lead to the evolution of evaluation frameworks, which show how different types of evaluation can be combined to obtain a holistic understanding of what a system can do (Reddy et al., 2021).

4.1 Impact evaluation

As discussed by Reiter (2025a), there is very little evaluation of real-world impact in the NLP and NLG research literature, by which we mean how real-world usage of an NLG system changes key performance indicators (KPIs). As NLG technology improves and becomes more widely used, we need more impact studies, especially if we want to measure utility in messy real-world contexts.

A good example is Bean et al. (2026), which measured how well LLMs can respond to health queries based on scenarios. LLMs do well at this task if given the scenario directly, or if they interact with an LLM-simulated user. However, if they interact with human users (who often communicate in a confused way), their performance is much worse. Hence if we want to genuinely evaluate how well an LLM can respond to health queries, we need to measure what happens when real people interact with the LLM. Ideally, this should be based on real patients asking about their health problems (Brodeur et al., 2026).

There are many ways to evaluate impact, including randomised controlled trials (RCT), A/B tests, before-and-after (pre-post) studies, and observational studies (Reiter, 2025b). By 2036, we hope that such evaluations will much more common. Most NLG evaluations will probably still use simpler and cheaper techniques, but a significant number will evaluate real-world impact.

4.2 Qualitative evaluation

Evaluation in NLG and NLP is almost always quantitative, and typically uses statistical hypothesis testing. Such evaluation is very important, but should be supplemented by qualitative evaluation, which can provide additional insights which are very useful in complex real-world contexts (Greenhaigh and Taylor, 1997; Tisdell et al., 2025).

Some qualitative evaluation techniques are already used in NLG, including error analysis (van Miltenburg et al., 2023) and analysis of free-text comments from participants (van der Lee et al., 2021). But many other techniques are rarer, including data collection techniques such as focus groups (Sun et al., 2026) and (semi-)structured interviews (Zhou et al., 2022), and analysis techniques such as thematic analysis (Guest et al., 2011) and content analysis (Sambaraju et al., 2011).

As NLG systems become more capable and are used in a wider variety of complex contexts, we expect that qualitative evaluation and insights will become more important, especially since many quantitative results will quickly become dated as newer models are released.

4.3 Safety evaluation

Safety evaluation is not new, it is a rapidly growing area of evaluation, which looks at whether AI systems can harm individuals (for example by encouraging suicide[1] or giving dangerous medical advice (Bickmore et al., 2018)) or society (e.g., by empowering hackers or terrorists) (Bengio et al., 2026).

We expect that safety will become one of the main foci of evaluation research. Ultimately, performance evaluation is of interest primarily to companies and academics who develop NLG technology, whereas safety evaluation is of interest to everyone who *uses* NLG technology, which is a much larger number of people. Safety evaluation is probably more important to society than performance evaluation. Indeed, governments have begun to impose safety standards on AI systems[2], and this may lead to formal government involvement in AI evaluation methodology.

Safety evaluation is also more challenging than performance evaluation, because it is about worst-case behaviour, and behaviour under adversarial attack (e.g., hackers trying to break into a system). Performance evaluations usually look at average case performance, so they can be computed based on a representative sample. Safety evaluation requires looking for misbehaviour everywhere, including edge cases, which are hard to predict for complex stochastic black box neural models. It will almost certainly require monitoring of the actual behaviour of deployed systems, as well as experiments on test data or test subjects.

[1] See https://www.thehumanlineproject.org/ stories, such as Badshah (2026)

[2] https://www.gov.uk/government/publications/generative-ai-product-safety-standards

5 Conclusion

NLG evaluation has changed dramatically between 1990 (mostly lingiustic evaluation) and 2026 (LLM-as-Judge and human annotation protocols). It continues to evolve, and the next ten years should be exciting, with more focus on impact, qualitative, and safety evaluation.

Acknowledgements

Many thanks to the anonymous reviewers, the members of the Aberdeen CLAN research group, and Saad Mahamood for their very helpful comments.

References

Mubashara Akhtar, Anka Reuel, Prajna Soni, Sanchit Ahuja, Pawan Sasanka Ammanamanchi, Ruchit Rawal, Vilém Zouhar, Srishti Yadav, Chenxi Whitehouse, Dayeon Ki, Jennifer Mickel, Leshem Choshen, Marek Šuppa, Jan Batzner, Jenny Chim, Jeba Sania, Yanan Long, Hossein A. Rahmani, Christina Knight, and 18 others. 2026. When ai benchmarks plateau: A systematic study of benchmark saturation. *Preprint*, arXiv:2602.16763.

Sydney Von Arx, Lawrence Chan, and Elizabeth Barnes. 2025. Recent frontier models are reward hacking. https://metr.org/blog/2025-06-05-recent-reward-hacking/.

Mohammad Asadi, Jack W. O'Sullivan, Fang Cao, Tahoura Nedaee, Kamyar Rajabalifardi, Fei-Fei Li, Ehsan Adeli, and Euan Ashley. 2026. Mirage: The illusion of visual understanding. *Preprint*, arXiv:2603.21687.

Michael W. Asher, Gillian Gold, Eason Chen, and Paulo F. Carvalho. 2026. Chatbots are undermining crowdsourced research in the behavioral sciences: Detecting artificial intelligence–assisted cheating with a keystroke-based tool. *Advances in Methods and Practices in Psychological Science*, 9(1):25152459261424723.

Nadeem Badshah. 2026. Teenager died after asking chatgpt for 'most successful' way to take his life, inquest told. *The Guardian*.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93.

Srinivas Bangalore, Owen Rambow, and Steve Whittaker. 2000. Evaluation metrics for generation. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 1–8, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Andrew M Bean, Ryan Othniel Kearns, Angelika Romanou, Franziska Sofia Hafner, Harry Mayne, Jan Batzner, Negar Foroutan, Chris Schmitz, Karolina Korgul, Hunar Batra, and 1 others. 2025. Measuring what matters: Construct validity in large language model benchmarks. *arXiv preprint arXiv:2511.04703; presented at Neurips 2025*.

Andrew M Bean, Rebecca Elizabeth Payne, Guy Parsons, Hannah Rose Kirk, Juan Ciro, Rafael Mosquera-Gómez, Sara Hincapié M, Aruna S Ekanayaka, Lionel Tarassenko, Luc Rocher, and 1 others. 2026. Reliability of llms as medical assistants for the general public: a randomized preregistered study. *Nature Medicine*, pages 1–7.

Anja Belz and Eric Kow. 2010. The GREC challenges 2010: Overview and evaluation results. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023. Non-repeatable experiments and non-reproducible results: The reproducibility crisis in human evaluation in NLP. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3676–3687, Toronto, Canada. Association for Computational Linguistics.

Yoshua Bengio, Stephen Clare, Carina Prunkl, Maksym Andriushchenko, Ben Bucknall, Malcolm Murray, Rishi Bommasani, Stephen Casper, Tom Davidson, Raymond Douglas, David Duvenaud, Philip Fox, Usman Gohar, Rose Hadshar, Anson Ho, Tiancheng Hu, Cameron Jones, Sayash Kapoor, Atoosa Kasirzadeh, and 73 others. 2026. International ai safety report 2026. *Preprint*, arXiv:2602.21012.

Timothy W Bickmore, Ha Trinh, Stefan Olafsson, Teresa K O'Leary, Reza Asadi, Nathaniel M Rickles, and Ricardo Cruz. 2018. Patient and consumer safety risks when using conversational assistants for medical information: an observational study of siri, alexa, and google assistant. *Journal of medical Internet research*, 20(9):e11510.

Peter Brodeur, Jacob M. Koshy, Anil Palepu, Khaled Saab, Ava Homiar, Roma Ruparel, Charles Wu, Ryutaro Tanno, Joseph Xu, Amy Wang, David Stutz, Wei-Hung Weng, Hannah M. Ferrera, David Barrett, Lindsey Crowley, Jihyeon Lee, Spencer E. Rittner, Ellery Wulczyn, Selena K. Zhang, and 29 others. 2026. A prospective clinical feasibility study of a conversational diagnostic ai in an ambulatory primary care clinic. *Preprint*, arXiv:2603.08448.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Giuseppe Carenini. 2000. A task-based framework to evaluate evaluative arguments. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 9–16, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Hua Cheng and Chris Mellish. 2000. An empirical analysis of constructing non-restrictive NP modifiers to express semantic relations. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 108–115, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Zerui Cheng, Stella Wohnig, Ruchika Gupta, Samiul Alam, Tassallah Abdullahi, João Alves Ribeiro, Christian Nielsen-Garcia, Saif Mir, Siran Li, Jason Orender, and 1 others. 2025. Benchmarking is broken–don't let ai be its own judge. *arXiv preprint arXiv:2510.07575, presented at Neurips2025*.

Matthew J. Duggan, Julietta Gervase, Anna Schoenbaum, William Hanson, III Howell, John T., Michael Sheinberg, and Kevin B. Johnson. 2025. Clinician experiences with ambient scribe technology to assist with documentation burden and efficiency. *JAMA Network Open*, 8(2):e2460637–e2460637.

Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating semantic accuracy of data-to-text generation with natural language inference. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 131–137, Dublin, Ireland. Association for Computational Linguistics.

Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2025. LLM-based NLG evaluation: Current status and challenges. *Computational Linguistics*, 51:661–687.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

T Greenhaigh and Rod Taylor. 1997. Papers that go beyond numbers (qualitative research)'. *British Medical Journal*, 315(7110):740–743.

Greg Guest, Kathleen M MacQueen, and Emily E Namey. 2011. *Applied thematic analysis*. sage publications.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 1999. The TIPSTER SUMMAC text summarization evaluation. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 77–85, Bergen, Norway. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kathleen F. McCoy, K. Vijay-Shanker, and Gijoo Yang. 1990. Using Tree Adjoining Grammars systemic framework in the. In *Proceedings of the Fifth International Workshop on Natural Language Generation*, Linden Hall Conference Center, Dawson, Pennsylvania. Association for Computational Linguistics.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 201–208, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Sandeep Reddy, Wendy Rogers, Ville-Petteri Makinen, Enrico Coiera, Pieta Brown, Markus Wenzel, Eva Weicken, Saba Ansari, Piyush Mathur, Aaron Casey, and 1 others. 2021. Evaluation framework to guide implementation of ai systems into healthcare settings. *BMJ health & care informatics*, 28(1):e100444.

Ehud Reiter. 2025a. *Natural Language Generation*. Springer.

Ehud Reiter. 2025b. We should evaluate real-world impact. *Computational Linguistics*, 51(4):1419–1431.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Ehud Reiter, Roma Robertson, A. Scott Lennox, and Liesl Osman. 2001. Using a randomised controlled clinical trial to evaluate an NLG system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 442–449, Toulouse, France. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Rahul Sambaraju, Ehud Reiter, Robert Logie, Andy Mckinlay, Chris McVittie, Albert Gatt, and Cindy Sykes. 2011. What is in a text and what does it do: Qualitative evaluations of an NLG system – the BT-nurse – using content analysis and discourse analysis. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 22–31, Nancy, France. Association for Computational Linguistics.

Mengxuan Sun, Ehud Reiter, Peter Murchie, Anne E Kiltie, George Ramsay, Lisa Duncan, and Rosalind Adam. 2026. Can chatgpt give holistic and accurate patient-centred information to oncology patients? a mixed-methods evaluation with stakeholders. *medRxiv*.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Anya Belz. 2024. Common flaws in running human evaluation experiments in NLP. *Computational Linguistics*, 50(2):795–805.

Craig Thomson, Ehud Reiter, and Barkavi Sundararajan. 2023. Evaluating factual accuracy in complex data-to-text. *Computer Speech and Language*, 80:101482.

E.J. Tisdell, S.B. Merriam, and H.L. Stuckey-Peyrot. 2025. *Qualitative Research: A Guide to Design and Implementation*. Wiley.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech and Language*, 67:101151.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Stephanie Schoch, Craig Thomson, and Luou Wen. 2023. Barriers and enabling factors for error analysis in NLG research. *Northern European Journal of Language Technology*, 9.

Alexander Waibel and Kai-Fu Lee. 1990. *Readings in speech recognition*. Morgan Kaufmann.

Kaitlyn Zhou, Su Lin Blodgett, Adam Trischler, Hal Daumé III, Kaheer Suleman, and Alexandra Olteanu. 2022. Deconstructing NLG evaluation: Evaluation practices, assumptions, and their implications. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 314–324, Seattle, United States. Association for Computational Linguistics.

Evaluation and Assessment as Complementary Frameworks

Elie Antoine
DIRO, RALI, Université de Montréal
Montréal, Québec, Canada
elie.antoine@umontreal.ca

Abstract

Language model capabilities have advanced faster than the methods used to evaluate them, particularly since the move from task-specific systems to general-purpose models which are deployed across an ever-widening range of tasks. When models were built for a single task, evaluation sat in a tight relationship between the task, the data, and the model. General-purpose models have weakened this relationship, and the evaluation practices that were built around it have not adjusted. This paper argues that addressing this gap requires treating *evaluation*, understood as quantitative performance measurement, and *assessment*, understood as the analysis of mechanisms and real-world behavior, as complementary rather than interchangeable. This distinction matters because *evaluation* is now often asked to stand alone in settings where a benchmark score cannot tell us what a model is doing, or how its behavior will hold up outside the benchmark.

1 Introduction

Natural Language Processing (NLP) has in recent years experienced an unprecedented expansion and societal impact. Until recently, it was less visible as a field, reaching the general public mainly through specific applications such as machine translation, autocorrect, or autocomplete. Today, a large share of the population[1] has heard of NLP through commercial models such as ChatGPT or Claude and a growing part use them in their professional and personal life or in even more personal contexts such as a substitute for consulting a physical or mental health professional.

[1]Numbers vary depending on region and sources : 68% to 90% of European and American workers have heard of generative AI according to (Bick et al., 2026), while roughly 90% of Americans have (Kennedy et al., 2025). These figures are centered around North America and Europe, thereby creating a picture that is likely inflated and not representative of the global population, a pattern reflected in the data on investment and impact (Microsoft AI Economy Institute, 2026).

This broad adoption has outpaced the field's ability to characterize what these models can and cannot do, as well as how and why they succeed or fail in the tasks for which they are used. Benchmarks remain the main framework through which the community tracks progress and compares models, and they have grown considerably in size and scope, now covering capabilities ranging from general knowledge and reasoning to instruction following, coding, tool use, and more abstract dimensions such as alignment and safety. Their logic, however, has mostly stayed the same: model outputs are compared against gold references, and the result is compressed into a single figure per task or benchmark. This is inherited from an era of task-specific systems, where the task was fixed and the reference was the right or at least reasonably attainable answer, and it was well suited to both the tasks and capabilities of the models at the time. Applied to general-purpose models that now saturate those benchmarks (Akhtar et al., 2026), it continues to produce rankings, but they tell us neither how a model handles a given input, nor how its evaluated behavior relates to its behavior in the open-ended tasks they are used for in practice. For example, ROUGE (Lin, 2004) was built for n-gram overlap on short extractive news summaries, where a reference output was close to the system output by design; it was stretched within summarization to longer and more abstractive cases, and is now reported as evidence of factual correctness in hallucination detection (Janiak et al., 2025) and of retrieval quality in RAG pipelines (Yu et al., 2025). These are settings where many outputs can be correct and similarity to one reference cannot tell them apart. The score keeps being produced, but is asked to carry the weight of a much broader claim about the model.

The position this paper takes is that filling this gap does not require replacing benchmarks, nor

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 16–23

scaling up the same logic. The question benchmarks answer, *how do models compare against a reference*, is not the only one worth asking, and the question of *how and why a model behaves as it does* calls for a different methodology. This is done here by clearly separating the concept of "examining" models into two frameworks, *evaluation* and *assessment*, which serve different purposes and answer different questions, and by arguing that we need to treat them as complementary rather than interchangeable, in the sense that one is often asked to do the work of the other.

The rest of this paper proceeds as follows. Section 2 sketches how the relationship between tasks, their associated data, inference methods, and evaluation has evolved with the rise of general-purpose models, and identifies the structural shift that motivates the rest of the argument. Section 3 defines and presents the difference between *evaluation* and *assessment*, and argues that the two are complementary rather than interchangeable. Section 4 discusses this distinction in relation to existing work, from linguistic probing and fine-grained evaluation to user-centered studies and more recent attempts to formalize evaluation by "vibes", and situates it alongside adjacent methodological proposals.

2 From task-specific to generic models

The development of NLP methods can be described through three main components, strongly interacting with each other: **the task**, together with the annotated or curated data that supports it, **the inference method** used to produce outputs, which today is largely a Large Language Model, and **the evaluation methods and metrics** through which outputs are judged. What is interesting is how the interactions between these components have evolved. Earlier in the development of NLP as a field, research was primarily focused on the task. When a new method was developed, the task was fixed and relatively narrow, and the data was annotated specifically for that task, both for training the inference method and for evaluating it. The architectures themselves were often designed around the task, with specific inductive biases for sequence labeling, parsing, or machine translation, partly because model capacity at the time did not permit a unified approach. The evaluation methods and metrics were tied just as directly to the task: a tagger for named entity recognition, a translation system, and a summarizer would not be evaluated in the same way. This made sense both because each metric measured something specific to the task and because model performance was lower, so comparing systems against a few canonical references was already challenging enough to produce meaningful differences.

The first shift came with the arrival of early transformer (Vaswani et al., 2017) models on a moderate scale, such as BERT (Devlin et al., 2019), BART (Lewis et al., 2020) and RoBERTa (Liu et al., 2019). Rather than training a model directly on the final task, a pre-training stage was introduced on a much simpler objective, typically predicting words hidden from their context, which was not itself a task of interest but proved remarkably effective as a "foundation" (Bommasani et al., 2021) for the real tasks. On top of this foundation, what was done was essentially the same as before, task-specific training, now in the form of fine-tuning a model.

The major change came with the idea of general-purpose models. T5 (Raffel et al., 2020) proposed that every task, regardless of its nature, could be cast in a single text-to-text format, while GPT-3 (Brown et al., 2020) showed that tasks could be specified at inference time through natural language prompts and a few examples, without any fine-tuning. Pushed to its limit through training models on conversational data and human feedback (Ouyang et al., 2022), this idea moved from a specialist-facing prompting paradigm to a general interaction mode, where tasks are handled through natural language interactions with a model never explicitly trained on most of them.

This last shift, sketched in Figure 1, marks an important change in which component is now driving the others. Research was previously organized around the task: the data was annotated for it, the architecture was designed for it, and the evaluation was tied to it. It is now organized around the model: the same system handles a wide range of tasks through prompting, and the tasks, data, and evaluations are defined in relation to what the model can do. One consequence is that because the model is generative, evaluation and task must be generative-compatible, regardless of their underlying structure. A classification or extraction problem becomes, through those models, a text-generation task, and the metric operates on the generated text rather than directly on the format and context it was originally about.

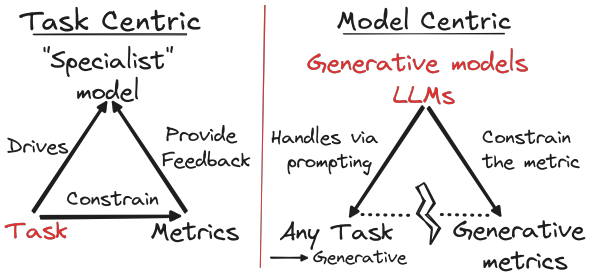

Figure 1: The shift from task-centric to model-centric NLP: the component that drives the others moves from the task to the model, and task and metric are recast in generative form to fit it.

A second consequence of this shift is the increased diversity and abstraction of the tasks that these models can now plausibly be asked to perform. Asking a model to read a long financial document, identify the companies mentioned in it, extract the relevant performance indicators, and produce a summary in the form of slides with supporting visualizations is not an unrealistic request today (Egg et al., 2025; Zheng et al., 2025). Evaluating such a task is another matter entirely. What should be evaluated? Each atomic component? The named entity recognition and table extraction steps can probably be formalized, but are not as simple to test in practice as this framing suggests, and more abstract or open-ended components, such as data cleaning, normalization, the choice of which indicators are most relevant (relevant for whom, and for what use?), the use of code tools to produce visualizations, and the judgment of those visualizations, raise a deeper question: what would a gold reference even look like? Even with expert human annotators, it is unclear what should be annotated, against what criteria, and whether two equally competent annotators would agree on the answer. *What makes a visualization good*, for instance, depends on whether it is grounded in real data, readable, communicative, and well-designed, criteria spread across factual, aesthetic, and communicative dimensions that sit beyond the reach of the grading formats we currently rely on. Cases like this are part of what motivates looking more carefully at what we mean when we talk about "evaluation".

3 Evaluation and Assessment

Two practices are often discussed together under the general heading of evaluation, but they serve different purposes. The first, *evaluation* in a stricter sense, is the established practice of quantitative performance measurement, and its canonical question is: *Does my model perform better than some other model?* Given a model and a reference, *evaluation* produces a score, or a set of scores, that allows models to be compared at a given point in time on one or several criteria. Its primary function is comparative. The second, *assessment*, is broader, and its canonical question is: *How and why does my model or method work, or fail to work?* In the definition adopted here, it covers any method aimed at answering that question, including linguistic probing, behavioral testing, human-centered studies, and mechanistic interpretability, among others.

Beyond the definitions just given, "assessment" as a term is also increasingly common in recent NLP writing. A keyword search across the ACL Anthology, covering paper titles and abstracts of the main *ACL venues, shows that the share of papers using *both* terms has grown sharply in recent years, from roughly 1.7% in 2020 to 10.6% in 2025, a nearly sixfold increase in five years.[2] Over the same period, the share of papers using only "assessment" has stayed flat at 3–4%.

How to read this trend is not obvious. One reading is that "assessment" is simply a more modern term, adopted under the pull of topics like capability assessment, risk assessment, or safety evaluation (Shevlane et al., 2023), without any underlying change in practice. Another is that the community is not replacing one term with the other but adding "assessment" alongside "evaluation" in contexts where the latter alone no longer seems to cover what authors want to say: on this reading, something closer to a second practice is in fact taking shape. This data alone cannot decide between these readings. Following the argument of this paper, settling the question would itself require *assessment* rather than pure quantitative description. The aim in the rest of this section is therefore different: to argue that the two questions just set out are different in kind, and that clearly separating *assessment* and *evaluation* in the way proposed here is useful for thinking about model analysis, whatever the vocabulary ends up doing in community practice.

Evaluation and *assessment*, on this distinction, are not competing practices, and the argument of this paper is not that one should replace the other. *Evaluation* produces comparable numbers, which

[2] More detail and figures on this can be found in Appendix A

is what lets a field track progress and decide between methods. *Assessment*, by contrast, takes a model as an object of study rather than a point on a scale, and asks what it is doing, where it breaks, and how its behavior looks in the settings where it is actually used. Many of the questions about a model are not comparative at the level of aggregate performance, even when the methods used to answer them are quantitative, as in probing accuracies or behavioral test pass rates. A rounded account of what a model is and does typically draws on both.

In current practice, however, the two are far from balanced. *Evaluation* remains the dominant one, inherited from task-specific systems where comparing scores against a reference was both the natural thing to do and a reliable indicator of progress. It has continued largely unchanged, even as the conditions that supported it have weakened. This is visible in benchmark scores reported for commercial models that do not reliably return the same output, or in capability claims staked on a single aggregate number. The imbalance is not that *evaluation* is done too much, but that it is often asked to stand alone, in settings where a score on its own simply cannot tell us what the model is doing or how its behavior will hold up outside the benchmark. This imbalance is visible in the literature itself: Reiter (2025) reports that roughly 0.1% of ACL Anthology papers evaluate real-world impact, and that even these typically treat the impact finding as secondary to a metric-based one. When *assessment* is done, it is often positioned as a supplement to *evaluation* rather than a finding in its own right.

4 Existing Work

The distinction between *evaluation* and *assessment* as drawn here is not a new one in the sense that the practices it names already exist, and have for some time. Assessment of NLP systems, and of generative systems in particular, has a long history. The STOP system, which generated tailored smoking-cessation letters, was evaluated in the early 2000s through a randomised controlled clinical trial with over 2,500 participants (Reiter et al., 2001, 2003). The trial measured whether smokers receiving tailored letters were more likely to quit than smokers receiving a generic letter, rather than how the generated letters scored against a reference. It was not, however, the dominant practice even then, and mostly operated as a complement to metric-based evaluation rather than driving the analysis. This distinction is even more necessary now, given that modern models are expected to handle a wide range of open-ended tasks through a single interface. In fact, a substantial body of existing work is already doing *assessment*, even when it is not labeled as such, and the question is less how to build the practice from scratch than how to recognize it as a coherent framework.

The clearest cases are "evaluations" that sit closer to *assessment* than to *evaluation* in the narrow sense, such as BLiMP (Warstadt et al., 2020) and Holmes (Waldis et al., 2024), and more generally the fine-grained evaluation tradition (Gehrmann et al., 2023; Ribeiro et al., 2020), where what matters is the detail of what is being examined rather than the overall ranking of models: no one really seems to care, in practice, about a model's rank on BLiMP. Human evaluation of natural language generation outputs belongs here as well, along with user studies and deployment reports that track how systems behave once they leave the lab, a concern shared with the broader human-computer interaction community.

The same holds for probing and mechanistic interpretability, where the goal is again not to rank models but to understand how they function, this time by looking at the model's internals rather than its behavior. Probing is routinely described as evaluation, but what it actually does is closer to *assessment* in the sense defined here: the goal is not to rank models but to characterise what they have learned (Rogers et al., 2020). Mechanistic interpretability is a different case: the field already positions it as reverse-engineering rather than evaluation, which makes it a limit case for the framing in a different way, not because it has to be reclassified, but because it raises its own validity questions. In the line of work surveyed by Feldhus and Kopf (2025), which focuses on generating natural-language concept descriptions for neurons, attention heads, and SAE features, automation now operates at two distinct layers: the descriptions themselves are generated by other language models, and their quality is evaluated mostly through automatic measures. They note that "concept descriptions are for humans, making human judgment essential for validating the meaningfulness of automated metrics", and yet observe that human evaluation remains comparatively rare in this part of the field. The practice fits our definition of *assessment*, but the move to automate it should be

approached with care: the automated judges and metrics are themselves measurement instruments, and the validity questions Wallach et al. (2025) raise for evaluation apply to them too.

One further approach deserves its own mention: evaluation by "vibes". Unlike most of the practices just mentioned, it did not originate in research and move outward to users, but the opposite: it grew out of informal discussions among users on 𝕏/Twitter, where people shared their own practical tests alongside a more diffuse sense of a model's competence. This includes the more classical dimensions of code and writing, as well as less tangible qualities: the model's capacity to interact in ways that feel useful rather than flattering, avoiding what work on language models describes as *sycophancy*: the tendency of a model to adapt its answer to the user's stated beliefs or preferences, including when this leads it to endorse incorrect claims (Perez et al., 2023; Sharma et al., 2024). It also includes the ability to avoid something closer to a textual uncanny valley, where the output reads as almost-right but not quite. Recent work has begun attempting to formalize this (Dunlap et al., 2025; Itzhak et al., 2026). These attempts raise a question: what makes the practice interesting is that it is grounded in the user's own intuitive and contextual judgment, and proposals to automate it have to decide how much of that grounding to preserve as it is turned into something reproducible at scale.

Three recent proposals name a related gap, each reaching for different vocabulary. Wallach et al. (2025) argue that evaluating generative AI systems should be understood as a *measurement* problem using the tools of social-science measurement theory. Weidinger et al. (2025) call for a mature *evaluation science* for NLP, and in particular for a *behavioral approach* that overlaps with what is here called *assessment*. Where these two are broad methodological reframings, Reiter (2025) is narrower, drawing a sharper binary between *metric evaluation* and what fits here as one specific kind of *assessment*, his *impact evaluation*: the measurement of real-world performance indicators in deployed usage rather than performance on a test set. Together with the vocabulary shift documented in Section 3, these proposals suggest that the community is actively trying to articulate and develop a practice that current methods do not quite cover.

5 Conclusion

This paper argues that *evaluation* and *assessment* should be treated as complementary practices rather than as a single one. The argument is not about vocabulary: other words could be used in place of these two, and several recent proposals already rely on different vocabularies to describe related concerns. The point is that *evaluation* is often used as a wide term, stretched to cover practices ranging from scoring against references to probing and behavioral testing, grouped together without clear distinctions about what each is set up to do.

Naming the two apart matters because *evaluation* carries a lot of weight in NLP and in benchmark-driven research more broadly. Comparing methods and tracking progress is what it is set up to do, but it is also one of the forces that direct research attention. What can easily be scored against a reference becomes the natural target of new work, and what resists that format is harder to argue for in publications. The open-ended cases sketched in Section 2 tend, on those grounds, to be taken up through *evaluation* rather than *assessment*, with the task reshaped until it produces a comparable score even where that framing does not really fit. *Assessment*, by contrast, is slower to set up and produces findings that do not reduce to a single comparable number, what Gehrmann et al. (2023) call an "incentive mismatch between conducting high-quality evaluations and publishing new models or modeling techniques". The asymmetry between the two is not the problem in itself; what follows from it is that the field's overall direction, what gets researched and optimized, ends up shaped largely by what *evaluation* can take up.

Recent proposals, from measurement theory to evaluation science to impact evaluation, are visibly trying to reach beyond *evaluation* as it is currently used. Naming the two apart is a small move toward that, and a reminder that not every question worth asking about a model is one a benchmark score can answer.

Limitations

The one piece of empirical material in this paper is the keyword search reported in Section 3, and it is too coarse and only quantitative to settle the question, as already noted in Section 3. A search over titles and abstracts cannot tell whether the rise of *assessment* alongside *evaluation* reflects a

change in what authors are doing or a change in what they call it. Settling what the trend actually reflects would itself require the kind of close reading the paper places under *assessment*.

Acknowledgments

I am particularly grateful to Frédéric Béchet, whose conversations during my PhD shaped how I think about NLP and evaluation, and which are at the root of the ideas developed here. I also thank Guy Lapalme for pointing me toward this workshop and encouraging me to formalize my ideas on the topic into a paper, and for his feedback on early versions, and Eliot Maes for his comments on later draft. Thanks also to the reviewers, whose feedback pushed me to clarify several parts of this paper. I am also grateful to Jian-Yun Nie, my postdoc advisor, for generously supporting my participation in this workshop.

References

Mubashara Akhtar, Anka Reuel, Prajna Soni, Sanchit Ahuja, Pawan Sasanka Ammanamanchi, Ruchit Rawal, Vilém Zouhar, Srishti Yadav, Chenxi Whitehouse, Dayeon Ki, Jennifer Mickel, Leshem Choshen, Marek Šuppa, Jan Batzner, Jenny Chim, Jeba Sania, Yanan Long, Hossein A. Rahmani, Christina Knight, and 18 others. 2026. When ai benchmarks plateau: A systematic study of benchmark saturation. *Preprint*, arXiv:2602.16763.

Alexander Bick, Adam Blandin, David J Deming, Nicola Fuchs-Schündeln, and Jonas Jessen. 2026. Mind the gap: Ai adoption in europe and the u.s. Working Paper 34995, National Bureau of Economic Research.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2021. On the opportunities and risks of foundation models.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lisa Dunlap, Krishna Mandal, Jacob Steinhardt, Joseph E Gonzalez, and 1 others. 2025. Vibecheck: Discover and quantify qualitative differences in large language models. In *International Conference on Learning Representations*, volume 2025, pages 69177–69205.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. 2025. Dabstep: Data agent benchmark for multi-step reasoning. *ArXiv preprint*, abs/2506.23719.

Nils Feldhus and Laura Kopf. 2025. Interpreting language models through concept descriptions: A survey. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 149–162, Suzhou, China. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *J. Artif. Int. Res.*, 77.

Itay Itzhak, Eliya Habba, Gabriel Stanovsky, and Yonatan Belinkov. 2026. From feelings to metrics: Understanding and formalizing how users vibe-test llms.

Denis Janiak, Jakub Binkowski, Albert Sawczyn, Bogdan Gabrys, Ravid Shwartz-Ziv, and Tomasz Jan Kajdanowicz. 2025. The illusion of progress: Re-evaluating hallucination detection in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34728–34745, Suzhou, China. Association for Computational Linguistics.

Brian Kennedy, Eileen Yam, Emma Kikuchi, Isabelle Pula, and Javier Fuentes. 2025. How americans view AI and its impact on people and society. Chapter: "Americans' awareness of AI and views of use in daily life, control over it." Available at https://www.pewresearch.org/science/2025/09/17/ai-in-americans-lives-awareness-experiences-and-attitudes/.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692.

Microsoft AI Economy Institute. 2026. AI diffusion data.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, and 44 others. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ehud Reiter. 2025. We should evaluate real-world impact. *Computational Linguistics*, 51(4):1419–1431.

Ehud Reiter, Roma Robertson, A. Scott Lennox, and Liesl Osman. 2001. Using a randomised controlled clinical trial to evaluate an NLG system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 442–449, Toulouse, France. Association for Computational Linguistics.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1):41–58.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, and 1 others. 2023. Model evaluation for extreme risks. *ArXiv preprint*, abs/2305.15324.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647.

Hanna Wallach, Meera Desai, A. Feder Cooper, Angelina Wang, Chad Atalla, Solon Barocas, Su Lin Blodgett, Alexandra Chouldechova, Emily Corvi, P. Alex Dow, Jean Garcia-Gathright, Alexandra Olteanu, Nicholas J Pangakis, Stefanie Reed, Emily Sheng, Dan Vann, Jennifer Wortman Vaughan, Matthew Vogel, Hannah Washington, and Abigail Z. Jacobs. 2025. Position: Evaluating generative AI systems is a social science measurement challenge. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Laura Weidinger, Inioluwa Deborah Raji, Hanna Wallach, Margaret Mitchell, Angelina Wang, Olawale Salaudeen, Rishi Bommasani, Deep Ganguli, Sanmi Koyejo, and William Isaac. 2025. Toward an evaluation science for generative ai systems.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2025. Evaluation of retrieval-augmented generation: A survey. In *Big Data*, pages 102–120, Singapore. Springer Nature Singapore.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu,

Xianpei Han, and Le Sun. 2025. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14402–14418, Suzhou, China. Association for Computational Linguistics.

A Anthology Search: Method and Data

The numbers reported in the main text come from a keyword search run directly on the ACL Anthology XML dump[3], parsed at the per-volume level rather than queried through the website. The corpus covers 117,273 papers from the main *ACL venues (ACL, EMNLP, NAACL, COLING, TACL, CL, LREC, and associated workshops) between 1952 and 2025; 2026 was excluded as only partially indexed at the time of the search.

For each paper, the title and abstract were concatenated and matched against two case-insensitive regular expressions with word boundaries:

```
\b(?:re)?(assess|assesses|assessed|
  assessing|assessment|assessments)\b
\b(?:re)?(evaluate|evaluates|evaluated|
  evaluating|evaluation|evaluations)\b
```

Each paper was then assigned to one of three mutually exclusive buckets, *evaluation only*, *assessment only*, or *both terms*, and yearly shares were computed against the total number of papers indexed for that year. An optional *re-* prefix is allowed (matching *reassess*, *reevaluate*, and their inflections). Other surface variants such as *evaluator*, *evaluative*, or *assessor* were not included.

Abstract coverage in the Anthology XML is uneven before roughly 2016. Several early volumes carry titles only, with the abstract field empty or missing, so pre-2000 rates under-count all three buckets, since only titles contribute to the match. Years before 1990 sit close to zero across all three buckets and were trimmed from the figure to keep the post-1990 trend readable. The post-2016 trend, on which the main argument rests, is not affected.

Figure 2 shows the full series from 1990 onward as the share of papers per year falling into each bucket. Shares are the comparable view across years given that the absolute number of papers grows by a factor of roughly twelve over the period.

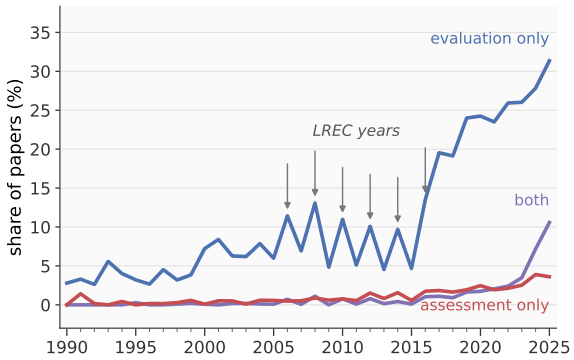

Figure 2: Share of papers in the ACL Anthology using the terms *evaluation*, *assessment*, or both, by year. Buckets are mutually exclusive per paper. Based on a keyword search over titles and abstracts of 117,273 papers across the main *ACL venues (1952–2025). The share of papers using both terms has grown from roughly 1.7% in 2020 to 10.6% in 2025.

[3] https://github.com/acl-org/acl-anthology

The Arabic Bible as an Evaluation Tool: The Case Study of the Khalīlī Arabic Dialect

Jakub Zbrzeżny[1] **Ehud Reiter**[2] **Wei Zhao**[2]

Department of Divinity[1] Department of Computing Science[2]

University of Aberdeen

jakub.zbrzezny@abdn.ac.uk e.reiter@abdn.ac.uk wei.zhao@abdn.ac.uk

Abstract

The paper presents a fully documented case study of how high-quality data combined with evaluators' expertise can be utilised for conducting basic NLP experiments in the realm of low-resource languages such as local varieties of Colloquial Arabic, and how the Arabic Bible, hitherto underutilised in NLP, can serve as an evaluation tool. Our experiments on one of the rural Palestinian Arabic dialects of al-Khalīl / Hebron illustrate two points. On the one hand, popular models are clearly limited in their ability to produce outputs of a high level of dialectal specificity (here: rural area surrounding a major urban centre). On the other hand, they are capable to generate accurate translations from such dialects into Modern Standard Arabic. Thus, the models appear better at understanding dialects than at producing dialects.

1 Introduction

Whether in its Jewish or one of several Christian forms, the Bible is a remarkable linguistic data set of extraordinary comparability, synchronic and diachronic alike. That Natural Language Processing (NLP) can benefit from the Bible both as a training data set and as an evaluation tool has been well noted (Alastruey et al., 2026). Still, even if Bible translations have often been the first written literary productions in many languages, there seems to be remarkable little literature on their usefulness for NLP in low-resource languages.

Such is the case of written expressions of Colloquial Arabic, the actual mother tongue of Arabic native speakers, which must be distinguished from formally learnt types of Arabic like Classical Arabic, the language of literature, or Modern Standard Arabic (MSA), the language of official media[1]. Unlike highly standardised forms of formally learnt Arabic, Colloquial Arabic represents a spectrum of varieties, which are commonly associated with geographical regions, countries, and smaller areas, from country districts to cities and their surroundings to individual towns or villages. Even such local varieties are often further subdivided into nomadic and sedentary types, of which the latter have distinct urban and rural forms[2].

The issue of the under-representation of Colloquial Arabic in NLP has been emphatically noted (e.g. Fakhraddin et al. (2025); Nacar et al. (2025)). Related Machine Translation challenges started to be addressed long before the advent of the Large Language Models (LLMs) and continue to be discussed (e.g. Zbib et al. (2012); Baniata et al. (2018); Harrat et al. (2019); Zakraoui et al. (2021); Alabdullah et al. (2025)). There is also no lack of experiments in testing LLMs capability to deal with it (e.g. Kadaoui et al. (2023); Khondaker et al. (2023))[3]. Nonetheless, in this context it is very rare to see the use of Colloquial Arabic Bible translations, and when such use is mentioned, the quality of data is not discussed (Sajjad et al., 2020). Admittedly, there is no comprehensive research on their largely undocumented nature in terms of their underlying texts, the sociolinguistic profile of translators, or translation techniques[4].

This paper provides an exploratory case study showing how a specific sample of Colloquial Arabic representing a sub-city variety can be employed in NLP in the context of collaboration between lo-

[1]For a more nuanced overview of the use of Classical Arabic and Modern Standard Arabic see reference works such as *Semitic Languages: an international Handbook* (Weninger et al., 2012), or the *Encyclopedia of Arabic Language and Linguistics* (Versteegh, 2005–2007).

[2]The aforesaid reference works will provide further introductory guidance into Arabic dialects.

[3]Note, however, that the research concerns high level (region / country) varieties of Colloquial Arabic (e.g. Elmadany et al. (2023); Al-Haff et al. (2022)). It is still rare to see work concerning varieties at a city-level (see Bouamor et al. (2018) on the MADAR corpus, or Mekki et al. (2026) on the Alexandria corpus.

[4]Such texts appear on confessional websites like `https://www.bible.com/` (currently including partial translations labelled as representing several country-level dialects.

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 24–32

cal experts, biblical scholars, and computer scientists. We discuss experiments in challenging selected LLMs to create dialect outputs meant to represent translations from English and MSA into this specific dialect, and to translate from samples of this dialect into MSA. Our case study is meant to be replicable and adaptable beyond the scholarship concerning the Arabic Bible, and even beyond Colloquial Arabic, extending towards other low-resource Semitic languages.

2 Methodology

2.1 Input texts

The source of our dialect data is a new paraphrastic rendition of selected biblical books from the pre-modern Arabic Bible into a particular dialect of the Levantine Arabic (al-Shāmī)[5]. It is one of the statistically most common Palestinian dialects, which is spoken in the southern West Bank (al-Khalīl / Hebron Governorate), and hence is called Khalīlī (hereafter Khalili[6]). Whether in its rural or urban form, it is locally easily identifiable and it bears significant cultural associations. Nonetheless, the dialect remains largely unexplored[7]. The rendition used as the source of data for our experiments is being prepared by a team of socio-linguistically aligned collaborators, who speak the rural variety of the Khalili dialect. The whole process is academically documented within research projects conducted at the University of Aberdeen[8]. At this moment, it encompasses drafts of the complete Book of Genesis (hereafter Genesis), and the complete Gospel according to Matthew (hereafter Matthew), constituting a corpus of approximately 40,000 words.

Our experiments investigate the following linguistic pairs or triplets:

- English to Khalili Dialect
- English to MSA to Khalili Dialect
- MSA to Khalili Dialect
- Khalili Dialect to MSA

The English text of Genesis was taken from *The Holy Scriptures According to the Masoretic Text* (1917)[9]. The English text of Matthew was taken from the OpenEnglishBible (2020)[10]. The MSA translation of both books comes from the STEP-Bible edition of the popular Van Dyck's translation (1865)[11] available under the licence CC BY-SA 4.0. The selection of English and MSA translations required that they are modern (but not necessarily the most recent), documented (but not necessarily meeting the current disciplinary standards), and available digitally in public domain.

Some of our experiments included among the input texts an example of how the dialect is translated into English or MSA. The dialect text was taken from selected fragments from the team's rendition of Genesis and Matthew, and these were accompanied by English and MSA translations created by our team for the purpose of the experiments.

The input texts were organised into twenty units, ten from Genesis, and ten from Matthew. On average, English units had 500 words, MSA units had 250 words, and Dialect units had 300 words (numbers rounded). The average word count was determined through initial checks, which tested the capacity of selected models to return meaningful and complete results. The examples consisted of 2,000 words in the dialect, 4,000 words in English, and 1,400 words in MSA (numbers rounded).

2.2 Models

The experiments involved three models. The selection condition was the presence of a privacy policy protecting input data from being incorporated into training data sets of a given model. Further, two models were meant to represent Generative AI tools that were publicly available at the time of experiments. The following were selected: Gemini 2 Flash (hereafter Gem2F) and ChatGPT 4o mini (2024.07.18) (hereafter GPT4om). One model was meant to represent a Generative AI tool used by more advanced non-expert users at the time of experiments. The following was selected: ChatGPT 4o (2024-08-06) (hereafter GPT4o). All three models were accessed through API calls facilitated by one of the commercial platforms (AI/ML API). This was to ensure that the texts from the hitherto unpublished Khalili dialect rendition of Genesis and Matthew remain outwith LLM training data sets.

[5] Eastern Mediterranean. Note that commonly used English categorizations based on wider geographical regions or modern states occasionally reflect the legacy of European colonialism rather than the actual dialect distribution.

[6] Simplified Romanization.

[7] For the most comprehensive bibliography of literature on Palestinian Arabic dialects, see Ulrich Seeger's list at https://arab.useeger.de/lit/Seeger-Biblio-Pal-Arabic.pdf (updated regularly; note that the list includes few positions in Modern Hebrew).

[8] Under the overarching title Hexapla Arabica.

[9] See https://en.wikisource.org/wiki/Bible_(Jewish_Publication_Society_1917.

[10] See https://openenglishbible.org/oeb/2022.1/OEB-2022.1-Cth.txt.

[11] See https://www.stepbible.org/version.jsp?version=AraSVD.

2.3 Prompts

All prompts for dialect outputs included the task of translating the input text from the stated language ("English" or "Modern Standard Arabic") into "the rural Arabic dialect of al-Khalīl in the West Bank.", or vice versa, from the dialect to MSA. The prompt default structure was: "Translate Text 1 from X into Z. Text 1 is as follows." There were three subsets of prompts:

- plain prompts, which were equal to the default (applied to all three models)
- more advanced versions of plain prompts to translate from English but with a mid-translation into MSA[12] (applied to all three models)
- prompts with an example among the input texts[13] (applied to GPT4o only).

Note that prompts for Gem2F and GPT4om had the parameters "temp" and "top_p" unstated. For GPT4o, these were always given explicit values: 0.0 and 0.1, respectively.

2.4 Output texts

The output texts were meant to be produced in the dialect and in MSA. The models were prompted to produce 200 units in the dialect. These included:

- 60 units from English directly to the dialect (all three models),
- 60 units from English through MSA to the dialect (all three models),
- 60 units from MSA to the dialect (all three models),
- 20 units from English to the dialect with an example (GPT4o).

On average, the dialect units had 250 words each, resulting in a corpus of approximately 50,000 words. There were also 80 units meant to represent MSA translations from the dialect. These included:

- 60 units created with prompts without an example (all three models),
- 20 units created with an example (GPT4o).

On average, the MSA units had 275 words, providing a corpus of approximately 22,000 words.

[12]These were formulated as follows: "Translate Text 1 from English into Modern Standard Arabic and then translate the Modern Standard Arabic translation into the rural Arabic dialect etc."

[13]These were formulated as follows: "Use the example of how Text 2 (the dialect) has been translated into Text 3 (English / MSA). […] Text 2 is as follows: […]. Text 3 is as follows: […]."

2.5 Evaluation

The evaluation was conducted independently by three team members acting as evaluators, who are native speakers of rural Khalili Arabic with solid knowledge of Classical Arabic (religious education) and MSA (secular education)[14]. The evaluators were closely aligned in terms of socio-linguistic features. They were highly familiar with the content of Genesis and Matthew through their earlier work on transcribing relevant texts from manuscripts and creating their Khalili dialect rendition. Before starting the evaluation, the evaluators discussed the scoring matrix among themselves and agreed on principles of scoring within the given parameters.

The related dialect outputs were randomly sorted in two batches: English to the dialect (140 units, grouped into 20 files, each with 7 units) and MSA to the dialect (60 units, grouped into 20 files, each with 3 units). The related MSA outputs were similarly arranged in one batch (80 units, grouped into 20 files, each with 4 units). The numerical identifiers of all units were anonymised through randomly generated numbers.

The evaluators were instructed to score each dialect output unit in terms of its dialect specificity according to the following metric with a 0-100 range:

100: rural Khalili dialect

75: Khalili dialect

50: Palestinian dialect

25: Levantine dialect

0: Colloquial Arabic

The dialect scoring matrix was formulated in writing in Colloquial Arabic in a descriptive way as follows:

100: العاميه الخليليه الفلاحيه

["rural Khalili Colloquial Arabic"]

75: العاميه الخليليه

["Khalili Colloquial Arabic"],

with the gloss: يعني الخليلي بس مش معروف مدني او فلاحي ["i.e. Khalili Arabic, but impossible to see whether urban or rural"]

50: العاميه الفلسطينيه

["Palestinian Colloquial"],

[14]The experiments were so designed that the evaluators' knowledge of English was irrelevant.

with the gloss: يعني الفلسطيني بس مش معروف من وين في البلد ["i.e. Palestinian Arabic, impossible to see from where in the country"]

25: العاميه الشاميه

["Levantine dialect"],

with the gloss: ممكن الفلسطيني ممكن الاردوني ممكن اللبناني ممكن السوري يعني بلاد الشام بس مش واضح من وين ["perhaps Palestinian Arabic, perhaps Jordanian Arabic, perhaps Lebanese Arabic, perhaps Syrian Arabic, that is, from the Levant, but unclear from where"[15]]

0: العاميه

["Colloquial Arabic"],

with the gloss: يعني ببين العاميه بس مش معروف من وين ["i.e., it seems to be Colloquial Arabic but it is impossible to see from where"]

The evaluation task formulated for MSA units was to score each MSA output unit on the scale from 0 to 100 in terms of its accuracy in translating the underlying dialect input. Again, the matrix was formulated in writing in Colloquial Arabic:

100: الترجمه الدقيقه

["accurate translation"]

50: مكس

["mixture"]

0: النص اللي مش ترجمه من اللي انتو كتبتو في لهجتكو بس مكس من ترجمات التوراه او الانجيل للرسميه اللي موجوده اونلاين

["The text does not constitute a translation from what you wrote in your dialect, but it is a mixture of MSA translations of the Hebrew Bible and the New Testament that are available online."].

The nature of the gloss to the score 0 stemmed from our initial experiments. The preliminary test outputs for translations from the dialect into English seemed to represent not the underlying paraphrastic dialect text in English, but rather a text appearing to be a mixture of modern English translations of corresponding passages in the Bible. This apparent tendency to align the supposed English translation of the Arabic paraphrase with the standard English text requires further investigation, which would to measure it and establish whether it occurs in relation to other textual corpora beyond the Bible in English.

Note that the entire work communication between the evaluators and the team members based in the UK was conducted in their dialect and without the use of English.

3 Results and Analysis

3.1 English / MSA to Dialect

The Figures 1 to 4 present the results of the evaluation. The horizontal axis represents the frequency of occurrences of a particular score given on the vertical axis:

75-99: Khalili dialect

50-74: Palestinian dialect

25-49: Levantine dialect

0-24: Colloquial dialect

Note that no unit was evaluated at **100** (rural Khalili dialect). The column 'error' indicates the number of instances where the model did not create an output.

The scoring of outputs created with prompts to translate from English directly into the dialect are shown in Figure 1. It will be seen that the Gem2F outputs received much higher scores than those created by the two other models. There is also a very high number of cases in which GPTo refused to perform the task where the prompt included an example). Figure 2 shows the results of prompts to translate from English into the dialect with an MSA mid-translation. Again, the Gem2F outputs were scored higher. When the scoring is compared across the outputs created with or without an MSA mid-translation, it will be noticed that Gem2F and GPTo performed better with MSA, but the opposite is true for GPT4om. This is shown in Figure 3. Finally, in direct translation from MSA to the dialect, the outputs created by Gem2F again scored much better than the two other models. This is shown in Figure 4.

[15] Due to the influence of Modern Hebrew and its distinctive features, the Israeli Arabic has not been enumerated here.

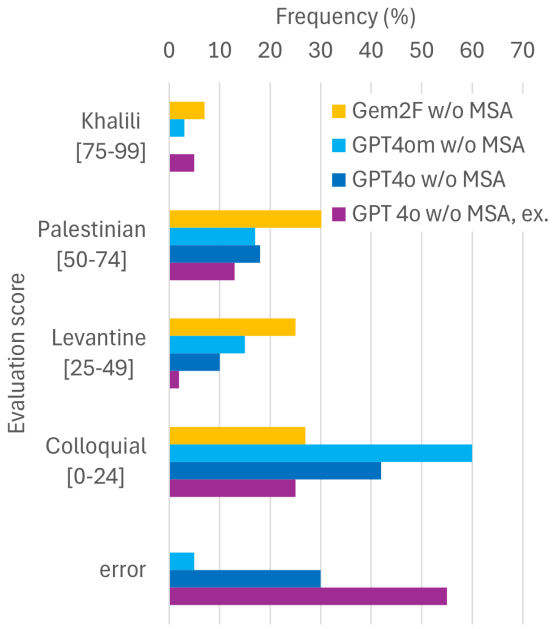

Figure 1: English to Dialect

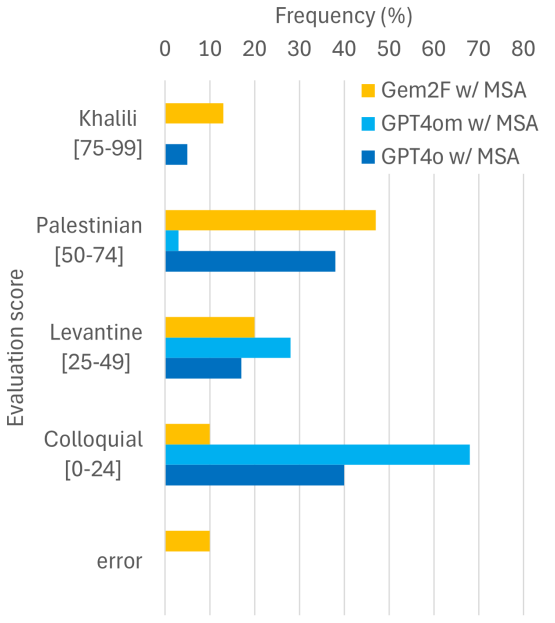

Figure 2: English to MSA to Dialect

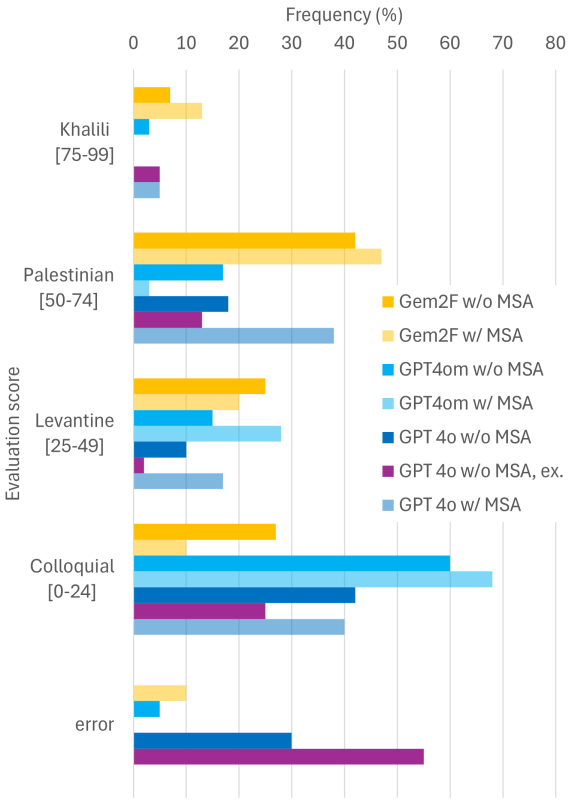

Figure 3: English to Dialect with/without MSA

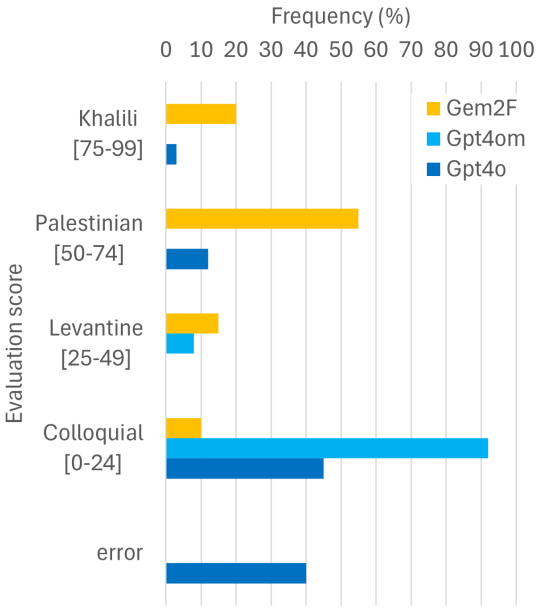

Figure 4: MSA to Dialect

Whereas the evaluation of particular models that were in use at the time of conducting our experiments provides only a snapshot into their capability at a point of time, the results give an insight into the evaluation process that remains independent of technical developments. This pertains to consistency in scoring among socio-linguistically aligned expert evaluators. It is shown in Figure 5. The figure gives the frequency of disagreements among the evaluators (the horizontal axis) of a particular value (the vertical axis), calculated as the difference between the highest and lowest scoring for each unit.

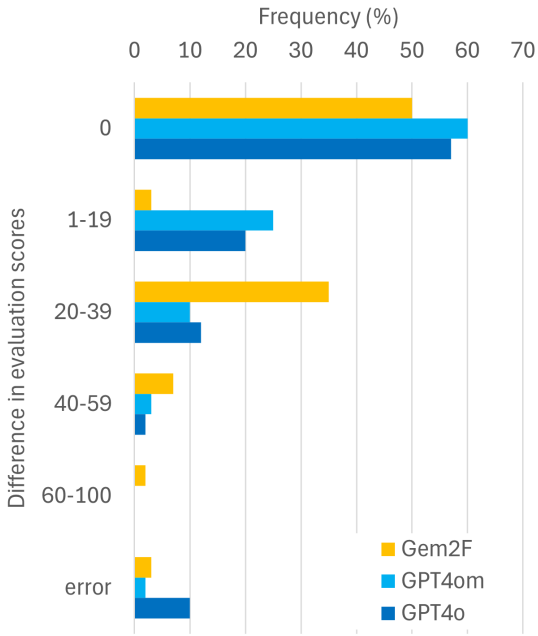

Figure 5: Disagreement scale in evaluation

This consistency is further illustrated by the average scores across the three evaluators given in Table 1[16].

Model	Ev.1	Ev.2	Ev.3	Diff. range
E>D / E>MSA>D				
Gem2F	52	48	47	5
GPT4o	39	36	37	3
GPT4om	27	28	25	3
MSA>D				
Gem2F	62	58	51	11
GPT4o	31	30	28	3
GPT4om	16	15	15	1

Table 1: Average scores across evaluators

[16]Cases where the models refused to produce an output were excluded from calculations.

3.2 Dialect to MSA

The evaluation of MSA units in terms of their accuracy towards the corresponding dialect input texts brought an unexpected lack of variation. With very few exceptions, the scoring was almost uniformly 100. This means that outputs were seen as representing an accurate translation of the underlying dialect texts. The experiments provide measurable evidence for this important finding.

This result underwent a deeper albeit preliminary investigation by a team member with expertise in Biblical Studies. Firstly, a unique set of four units that scored consistently not 100 but 50 across all three evaluators was investigated along the lines of Qualitative Error Analysis. The unit is made largely of the passage with the so-called genealogy of Jesus (Matthew 1:1-17), which is a list of personal names. A sample of Matthew 1:2-7 contains 30 names in the dialect text. In many cases, their orthography differs from that found in published MSA translations of Matthew[17]. Indeed, out of 30 names, only 8 were uniform across the dialect input text and the generated MSA output texts[18], and only 3 were in partial agreement. The other 17 cases had the names 'corrected' in the MSA outputs into forms identical with or closer to those found in published MSA translations (e.g. the 'incorrect' "Būdh" amended into "Boaz", or the 'incorrect' "Dūth" amended into "Ruth"). These 'corrections' were detected by our evaluators and interpreted as signs of inaccuracy in translating the dialect version into MSA.

Secondly, it was investigated in detail how a clearly paraphrastic dialect passage was translated into MSA while scoring 100 in terms of accuracy. The selected passage was Matthew 1:18-19, which reads in one of the standard English translations as follows:

> Now the birth of Jesus the Messiah took place in this way. When his mother Mary had been engaged to Joseph, but before they lived together, she was found to be pregnant from the Holy Spirit. Her husband Joseph, being a righteous man and unwilling to expose her to public disgrace, planned to divorce her quietly. (NRSV2021)

[17]This is a result of pre-modern and modern scribal mistakes in copying the list of largely unfamiliar and non-Arabic names.

[18]These were mostly well-known names such as, in their English form, "Abraham", "Isaac", "Jacob", "David", or "Judah".

The dialect rendition gives a significantly different text, with several omissions, additions, and changes. It can be translated into English as follows, with omissions indicated by the underscore, additions by the underline, and changes by italics:

> _____ Mary, the mother of Christ, was engaged to Joseph. Before they got married (that is, before the nuptial night occurred), *Joseph noticed* that Mary, his fiancée, is pregnant _____. Just note: it was before the nuptial night. Now, Joseph was a good person. When he learnt about the matter, he did not want to put her to shame. To the contrary, he wanted to protect her. Thus, he said to himself that he should divorce her in secret, leave her, and stay away from her[19].

All the outputs closely followed the dialect input text. Significantly, they did not add the 'missing' fragments such as the introductory sentence or the mention of the Holy Spirit. Further, they replicated explanatory additions either in full alignment with the dialect input (e.g. on the 'nuptial night'), or in partial alignment (e.g. the phrase "to protect her" occurs in 2 out of 4 outputs, and the phrase "leave her, and be away from her" occurs in 3 out of 4 outputs). They also followed the changed constructions (the active "Joseph noticed" instead of the passive "she was found"). Thus, the evaluation was correct in terms of assessing the surprising accuracy of the translation of a passage that could have been aligned with standard MSA translations.

Finally, one of the key New Testament passages was assessed, that is, the so-called "Lord's Prayer", also known as "Our Father", in Matthew 6:9-13. This central text of Christian tradition would have been expected to be particularly prone to being aligned with its standard translations, especially given the fact that the Khalili dialect rendition departs from the well-known wording and provides a highly poetic rendition of the text. Among its most striking features is shift in the possessive pronoun attached to the word 'Father' from 'our' to 'your' (plural), and a paraphrastic translation of 'your will be done' into 'may what God has decreed come to being (o Lord, according to what you wish!)'. These highly unusual features are retained in all four MSA outputs, with just one exception with a case of the standard 'Our Father'.

[19]Probably meant as to refrain from violence against her.

This preliminary investigation into a handful of notable cases exemplifies what the Evaluators detected in their assessment of MSA outputs: accuracy and capacity to morph dialect phrases into MSA. Its potential advantages notwithstanding, this raises a concern related to speeding up the process of dialect levelling, especially if predictive text or autocorrections are powered by LLMs.

4 Conclusions

The case study presented in this paper shows NLP experiments with Colloquial Arabic related to the Arabic Bible bringing meaningful results. This is shown in Table 2 with average scores[20] across the three models. One model (Gem2F) performed clearly better than others in creating dialect outputs, especially when translating from MSA or with an MSA mid-translation. However, even in these two cases, the model did not reach the level of Khalili specificity, and, on average, produced outputs categorised as representing only more broadly Palestinian dialects. The majority of outputs from other models were categorised as representing Levantine dialects. This inability of the models to produce outputs meant to have a high level of dialect specificity contrasts with the fact that all the models were assessed as highly accurate in translating dialect inputs into MSA.

The fact that the results were meaningful lies in the expertise of the evaluators and in the collaboration of local experts, biblical scholars, and computer scientists. This ensured the quality of data as well as culturally and socially appropriate evaluation process. Using Colloquial Arabic as a medium of work communication should also be noted.

It remains to be explored how applying NLP to Colloquial Arabic can contribute to the investigation of some complex linguistic phenomena such as Arabic diglossia. This potential extends beyond Arabic and is applicable to other low-resource Semitic languages such as ancient Hebrew, ancient Aramaic, or endangered Neo-Aramaic dialects. Such exploration can be successfully conducted not only within the digital humanities, but also — perhaps even more effectively — by means of multidisciplinary collaboration between the humanities and computer science.

[20]Cases where the models refused to produce an output were excluded from calculations.

Model	Direction	Avg. score	Category
Dialect Specificity			
Gem2F	MSA>D	57	Palestinian dialect
Gem2F	E>MSA>D	54	Palestinian dialect
Gem2F	E>D	45	Levantine dialect
GPT4o	E>MSA>D	41	Levantine dialect
GPT4o	E>D+ex	38	Levantine dialect
GPT4o	E>D	32	Levantine dialect
GPT4om	E>D	31	Levantine dialect
Gpt4o	MSA>D	30	Levantine dialect
GPT4om	E>MSA>D	24	Colloquial Arabic
Gpt4om	MSA>D	15	Colloquial Arabic
Accuracy			
GPT 4o	D>MSA+ex	97	Accurate translation
GPT 4o	D>MSA	96	Accurate translation
Gem2F	D>MSA	96	Accurate translation
GPT4om	D>MSA	95	Accurate translation

Table 2: Average scores across models

Limitations

It would be valuable to conduct follow-up experiments with evaluators who are not immediately familiar with Genesis and Matthew to see how they score human-created outputs in relation to AI-created counterparts, when both types of texts are randomly mixed with anonymised labels. High scores for the actual Khalili texts would provide further validation for the method presented in this paper.

Ethical Statement

All due considerations (i.a. religious, social, cultural, political, ethical, economic) were undertaken prior to the commencement of the research to ensure that the safety of the evaluators was not put at risk, and that their work was appropriately remunerated.

Acknowledgments

The research presented in this paper was conducted as part of the project 'Al-ᶜĀmmīyah (Colloquial Arabic) and Generative AI — A Snapshot of its Emerging Text-to-Text Abilities' funded by the Royal Society of Edinburgh (Collaboration Grant 4423). The underlying dialect texts were created within projects co-funded by the British Academy, the University of Aberdeen, the Department of Near Eastern Studies at Cornell University, and by private donors. This complex work involves nearly twenty colleagues from Professional Services at the University of Aberdeen, to whom we are grateful for their essential support as part of the wider team.

Finally, this research would not have been possible without the significant intellectual contributions of our five Khalili project partners in the West Bank, led by Ahmad Hroub. Their continued commitment is deeply appreciated — يعطيكو العافيه.

References

Karim Al-Haff, Mustafa Jarrar, Tymaa Hammouda, and Fadi Zaraket. 2022. Curras + baladi: Towards a levantine corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 769–778, Marseille, France. European Language Resources Association.

Abdullah Alabdullah, Lifeng Han, and Chenghua Lin. 2025. Advancing dialectal arabic to modern standard arabic machine translation. *Preprint*, arXiv:2507.20301.

Belen Alastruey, Niyati Bafna, Andrea Caciolai, Kevin Heffernan, Artyom Kozhevnikov, Christophe Ropers, Eduardo Sánchez, and 1 others. 2026. Omnilingual MT: Machine translation for 1,600 languages. *arXiv preprint arXiv:2603.16309*.

Laith H. Baniata, Seyoung Park, and Seong-Bae Park. 2018. A neural machine translation model for arabic dialects that utilises multitask learning (MTL). *Computational Intelligence and Neuroscience*, 2018:7534712.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani,

Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association.

AbdelRahim Elmadany, ElMoatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. ORCA: A challenging benchmark for arabic language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9559–9586, Toronto, Canada. Association for Computational Linguistics.

Alwajih Fakhraddin, Abdellah El Mekki, Samar Mohamed Magdy, AbdelRahim A. Elmadany, Omer Nacar, El Moatez Billah Nagoudi, Reem Abdel-Salam, Hanin Atwany, Youssef Nafea, Abdulfattah Mohammed Yahya, Rahaf Alhamouri, Hamzah A. Alsayadi, Hiba Zayed, Sara Shatnawi, Serry Sibaee, Yasir Ech-chammakhy, Walid Al-Dhabyani, Marwa Mohamed Ali, Imen Jarraya, and 25 others. 2025. Palm: A culturally inclusive and linguistically diverse dataset for arabic LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32871–32894, Vienna, Austria. Association for Computational Linguistics.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2019. Machine translation for arabic dialects (survey). *Information Processing Management*, 56(2):262–273. Advance Arabic Natural Language Processing (ANLP) and its Applications.

Karima Kadaoui, Samar M. Magdy, Abdul Waheed, Md Tawkat Islam Khondaker, Ahmed Oumar El-Shangiti, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. TARJAMAT: Evaluation of bard and ChatGPT on machine translation of ten arabic varieties. In *Proceedings of ArabicNLP 2023*, pages 52–75, Singapore (Hybrid). Association for Computational Linguistics.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on arabic NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247, Singapore. Association for Computational Linguistics.

Abdellah El Mekki, Samar M. Magdy, Houdaifa Atou, Ruwa AbuHweidi, Baraah Qawasmeh, Omer Nacar, Thikra Al-hibiri, Razan Saadie, Hamzah Alsayadi, Nadia Ghezaiel Hammouda, Alshima Alkhazimi, Aya Hamod, Al-Yas Al-Ghafri, Wesam El-Sayed, Asila Al sharji, Mohamad Ballout, Anas Belfathi, Karim Ghaddar, Serry Sibaee, and 28 others. 2026. Alexandria: A multi-domain dialectal arabic machine translation dataset for culturally inclusive and linguistically diverse llms. *Preprint*, arXiv:2601.13099.

Omer Nacar, Serry Taiseer Sibaee, Samar Ahmed, Safa Ben Atitallah, Adel Ammar, Yasser Alhabashi, Abdulrahman S. Al-Batati, Arwa Alsehibani, Nour Qandos, Omar Elshehy, Mohamed Abdelkader, and Anis Koubaa. 2025. Towards inclusive arabic LLMs: A culturally aligned benchmark in arabic large language model evaluation. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 387–401, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hassan Sajjad, Ahmed Abdelali, Nadir Durrani, and Fahim Dalvi. 2020. AraBench: Benchmarking dialectal arabic-english machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5094–5107, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kees Versteegh, editor. 2005–2007. *Encyclopedia of Arabic Language and Linguistics*. Brill.

Stefan Weninger, Geoffrey Khan, Michael P. Streck, and Janet C. E. Watson, editors. 2012. *Semitic Languages: An International Handbook*. De Gruyter.

Jezia Zakraoui, Moutaz Saleh, Somaya Al-Maadeed, and Jihad M. Alja'am. 2021. Arabic machine translation: A survey with challenges and future directions. *IEEE Access*, 9:161445–161468.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT '12)*, pages 49–59, USA. Association for Computational Linguistics.

RAG as a collapsed NLG pipeline

Adarsa Sivaprasad, Barkavi Sundararajan, David M Howcroft
Department of Computer Science
University of Aberdeen, UK

Abstract

The NLG pipeline of Reiter and Dale has long served as the foundational framework for data-to-text system design and evaluation. However its relationship to modern generative architectures remains underexplored. In this conceptual analysis, we argue that Retrieval-Augmented Generation (RAG) constitutes a collapsed and partially reconstructed instantiation of the classical NLG pipeline, using it to identify failure modes of RAG around context faithfulness and retrieval non-determinism.

1 Background and motivation

The Natural Language Generation (NLG) pipeline proposed by Reiter and Dale (1997) has been the dominant conceptual framework for the design and development of data-to-text systems for over two decades. By decomposing data-to-text generation into the discrete modular stages of document planning, microplanning and surface realisation, the pipeline provided both a principled architecture for system builders and a shared vocabulary for researchers, structuring how data-to-text systems were designed and analysed (Reiter, 2025). Its influence is evident in systems across application domains, such as SumTime (Sripada, 2003), which employed explicit data interpretation and document planning for weather-forecast generation, and BT-45 (Portet et al., 2007) for clinical text summarisation. The pipeline shaped data-to-text generation by framing generation as a sequence of explicit decisions about content selection, document structuring, and linguistic expression.

The emergence of neural architectures for language generation shifted the field away from this modular paradigm. Sequence-to-sequence models (Sutskever et al., 2014) and later transformer-based architectures (Vaswani et al., 2017) enabled end-to-end generation without explicitly encoding content selection, planning, and surface realisation. Rather than eliminating these pipeline functions, neural architectures leave them implicit, distributing the corresponding decisions across learned representations. While some work has explored explicitly learning to make these decisions in neural networks (e.g., Puduppully et al., 2019), most recent work has built on transformer models without architectural changes designed to capture these aspects of the classical pipeline.

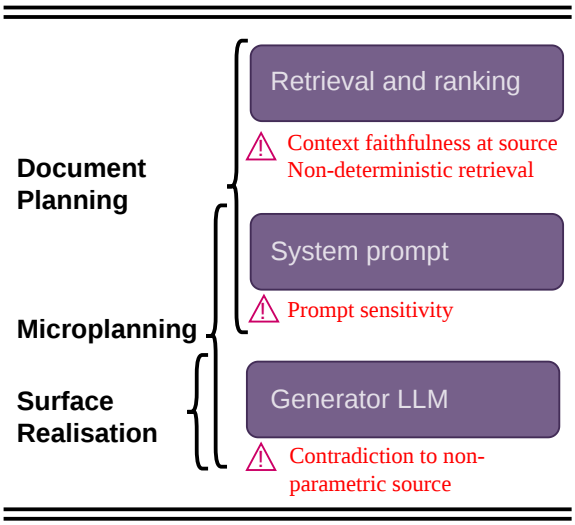

Figure 1: Classical NLG pipeline stages (left) and the corresponding RAG components (right), including overlap between document planning and microplanning. Each of the three RAG components is associated with particular challenges, highlighted below each component.

The advent of large-scale generative language models in 2020 (Brown et al., 2020) marked a further shift from both the symbolic pipeline and earlier neural architectures toward generation based on (large) language models ((L)LMs). While these models exhibit remarkable performance across a wide range of generation tasks, their reliance on parametric knowledge encoded during pre-training renders them susceptible to factual errors and confident confabulation, widely termed hallucination

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 33–38

(Ji et al., 2023). Retrieval-Augmented Generation (RAG), (Lewis et al., 2020), emerged as a widely adopted architectural response to this limitation. By grounding text generation on external, curated, non-parametric knowledge sources, RAG substantially reduces hallucination (Shuster et al., 2021) and has become a dominant framework for deploying LLMs in knowledge-intensive tasks.

In this work, we reflect on RAG through the lens of the classical NLG pipeline, analysing how its components align with — and diverge from — the classical NLG pipeline stages of data interpretation, document- and micro-planning, and surface realisation. We argue that RAG implicitly implements several features of the NLG pipeline in novel forms (summarised in Figure 1), with document planning taking place across the retrieval, ranking, and prompting stages of a RAG system, microplanning overlapping with prompting and generation from LLMs, and surface realisation fully limited to the LLM generation stage. The analogy also highlights the fundamentally new concerns in a RAG around retrieval quality, context faithfulness, and conflicts between retrieved content and the generator's parametric knowledge (Longpre et al., 2021), that fall outside the original pipeline's scope. By mapping this correspondence, we aim to illuminate both what has been recovered and what has been lost in the transition from symbolic to retrieval-augmented generation and to explore the implications for evaluation, interpretability, and system design in contemporary NLG.

2 The Classical NLG Pipeline

The classical NLG pipeline of Reiter and Dale (1997) decomposes generation into three main stages: *document planning*, *microplanning*, and *surface realisation*. Later data-to-text architectures extended this framework with earlier stages, such as signal analysis and data interpretation, to handle raw data inputs (Reiter, 2007, 2025). We consider the original three-stage pipeline to be closer to how RAG systems are deployed. Signal analysis and data interpretation as pre-cursor stages to document planning are designed to take masses of raw data and convert them into meaningful units, while the retrieval index of a RAG system is more like the database or collection of facts which results from such analyses: they provide the materials from which the document can be planned based on a given user query.

We illustrate these stages using SumTime (Sripada, 2003), a rule-based NLG system that generates marine weather forecasts from numerical weather prediction (NWP) data (e.g., wind speed and direction, temperature, pressure) to support offshore operations. It uses rules derived from knowledge acquisition tasks such as corpus analysis, expert consultations, and think-aloud sessions with forecasters, to drive decisions at each stage (Sripada et al., 2004). An extract of SumTime output is reproduced from Reiter (2025):

Wind(10M): S 16–21 backing SSE 21–26 by mid afternoon, then veering S by early evening and SSW 18–23 by midnight.

In this example, *S*, *SSE*, and *SSW* denote wind directions on the standard 16-point compass, and *"backing"* and *"veering"* are change verbs to describe shifts in wind direction.

Document planning makes two key decisions: *content selection* (which events to communicate) and *document structuring* (how to organise them into a coherent narrative) (Reiter, 2025). In SumTime, document structuring follows the forecast structure recommended by Weathernews UK, while content selection uses a bottom-up segmentation algorithm to group adjacent readings and identify meteorologically significant wind states and direction changes to mention (Sripada et al., 2003, 2004). These selected events are then ordered chronologically, as in the example above.

Microplanning determines how the document plan is expressed, including *lexical choice*, *referring expression generation*, and *aggregation*. In SumTime, corpus-derived rules guide the selection of domain-specific forecast verbs, such as *"backing"* and *"veering"* for wind-direction changes and *"increasing"* for wind-speed changes (Reiter, 2025). The rules also map time steps to time expressions such as *"by mid afternoon"* and *"by midnight"* (Sripada et al., 2002).

Surface Realisation renders the microplan as grammatically correct text, handling syntax, morphology, and punctuation (Gatt and Reiter, 2009).

3 RAG as a Collapsed Pipeline

A RAG system combines the parametric knowledge encoded in a pre-trained neural language model with a document index containing supple-

mental information which is called *non-parametric* as it is not encoded in the LMs trained parameters (Lewis et al., 2020). The non-parametric knowledge source is encoded in dense vector representations to enable easy retrieval based on distributional semantics. These representations capture meaning in a vector space, similar to the 'semantic' representations learned by neural LMs.

The LM combines retrieved non-parametric knowledge with parametric knowledge guided by a *prompt* which is a natural language specification of what the resulting text should look like, with or without examples (in few-shot and zero-shot prompting, respectively). In this analysis, we examine each of these components of a RAG system to characterise the data-to-text functions they perform.

3.1 Retrieval and reranking

Domain knowledge, typically in the form of documents, is represented as a collection of embeddings which implicitly encode the semantic content of the underlying document. Retrieval selects content from this indexed knowledge source based on a *user query*, potentially reranking this content by relevance (Glass et al., 2022), is therefore functionally analogous to content selection in the document planning module (Reiter and Dale, 1997).

However, classical document planning can be deterministic and is fully controllable when it is rule-based. The content selection is purposive and input is interpreted with respect to a specific generation task. In RAG, encoding is performed offline and independently of any particular query – the same representation must serve all possible future retrieval contexts. Hence, while a non-parametric source created for a specific task can be faithful to its context, general-purpose knowledge sources are susceptible to semantic ambiguity, since embeddings constructed without a specific generation intent cannot anticipate the full space of queries they will be expected to serve. Following Es et al. (2024), we term this *context faithfulness at the source* – a faithfulness risk structurally absent from the classical pipeline.

Further, document structuring in the classic NLG pipeline is governed by explicit communicative goals, and can be evaluated by comparing system outputs against expert-authored texts (Sripada, 2003). Retrieval, by contrast, is generally probabilistic and less deterministic. In the absence of an exact semantic match for a query, the retriever may select a misaligned yet closest match, introducing uncertainty into the content selection process. Uncertainty may also arise from the contradiction of the retrieved contents with the parametric knowledge of the generator LLM (Longpre et al., 2021). Since relevance scores do not guarantee factual alignment, the document planning stages of RAG require dedicated retrieval quality assessment.

3.2 System prompt

Prompt engineering has become an integral component of generative AI workflows, and, within RAG specifically, the prompt provides instructions influencing content aggregation and specificity, and specifying how to align to the communicative goal of the response (Schulhoff et al., 2024), ultimately overlapping with both document and micro-planning decisions.

This correspondence, however, remains underexamined. Unlike the retrieval stage, where the functional analogies to document planning is relatively direct, the prompt and the generator LLM relationship is tightly coupled. The prompt does not operate independently: its effect on aggregation and lexical choice is contingent on the instruction-following behaviour of the specific generator model it addresses, making it difficult to isolate prompt-level microplanning from the broader generative behaviour of the LLM. Further, different prompting strategies — zero-shot, few-shot, or chain-of-thought, may exert different influences on how the retrieved content is aggregated and expressed.

3.3 Generator LLM

Surface realisation in the classical NLG pipeline is the final stage, responsible for converting abstract linguistic representations into grammatically well-formed, fluent text. Within a RAG architecture, the LLM generator occupies this role.

Neural language models collapse the distinct stages of the NLG pipeline into a single learned process. Puduppully et al. (2019) demonstrate that neural models can encapsulate functions such as content selection, aggregation, and surface realisation within unified parameter spaces, for example. RAG can therefore be understood as a deliberate architectural counter-move. By externalising data interpretation, content planning and expression planning (prompting) into discrete upstream stages, it partially reconstructs the modularity that end-to-end neural generation had collapsed, constraining the LLM to focus on what it demonstrably does

NLG Pipeline Stage	Classical System *(Weather forecasts)*	RAG System *(Suggesting activities based on weather)*	Key Divergence
Document planning *(Content selection & structuring)*	Bottom-up segmentation selects significant wind states and direction changes. Selected events are ordered chronologically in the forecast text.	Probabilistic retrieval of activity descriptions based on ideal weather conditions provides high level structure in the LLM prompt.	Retrieval is non-deterministic with relevance scores which do not guarantee factual alignment.
Microplanning *(Lexicalisation & aggregation)*	Corpus-derived rules choose forecast verbs such as *"backing"*, *"veering"*, and *"increasing"*, and map time steps to expressions (*"by midnight"*).	Prompt guidance for what kind of register to use, influencing word and aggregation choices.	Prompt and LLM coupling means microplanning cannot be isolated from surface realisation.
Surface realisation	Realiser renders the forecast in domain-standard form, including syntax, punctuation, and formatting, e.g., *"S 16–21 backing SSE 21–26 by mid afternoon"*.	Generator LLM producing fluent activity suggestions based on on retrieved passages and given prompt.	LLM generation combines surface realisation with content and expression decisions.

Table 1: Highlighting similarities and differences between classical data-to-text pipeline systems like SumTime for weather forecasts (Sripada, 2003) and a potential RAG application in an related domain (suggesting activities appropriate for given weather conditions) highlighting differences between a classical pipeline application and a RAG application.

best, namely producing fluent, coherent, grammatically correct text conditioned on a structured input context. However, as a probabilistic model, the RAG pipeline accumulates different uncertainties discussed earlier at each stage. Further, the contradiction of retrieved information with the learned parametrised knowledge of the generator LLM is an added risk (Longpre et al., 2021).

3.4 Implications for Evaluation

The pipeline decomposition reveals where RAG can fail, and also informs how to detect those failures. We summarise evaluation criteria in RAG, motivated by classic NLG stage-specific failures :

- Contextual faithfulness check at retrieval to ensure appropriate content is available in knowledge source .
- Conduct explicit retrieval performance evaluations.
- Test the sensitivity of how the document plan is expressed and handled on the prompting strategy. This must include stability of output, output length and meaning preservation (Schulhoff et al., 2024).
- Quantifying uncertainty in surface realisation due to the contradiction of retrieved context and underlying LLM training knowledge, such as proposed in (Longpre et al., 2021).
- Granular error annotation addressing microplanning and surface errors, such as done (Thomson and Reiter, 2020; Sundararajan et al., 2025), that account for both lexical choices and aggregation decisions.

4 Discussions and Conclusion

By showing the analogy of RAG to the classic data-to-text pipeline, we illustrate that the architecture constrains generative LLMs, in which the model is

implicitly responsible for all pipeline stages simultaneously and where hallucination and factual drift are consequently most acute. As a surface realiser, unlike classical surface realisers, which operate on verified, structured linguistic representations, we note that the LLM generator lacks an intrinsic mechanism to detect or reject factually inconsistent, uncertain retrieved content. Faithfulness to the retrieved context is therefore not guaranteed by the architecture itself, but must be enforced through additional evaluation. The classical NLG pipeline has been applied across a wide range of data-to-text use cases and input modalities, including time series (Sripada et al., 2003), relational tables (Puduppully et al., 2019) and semantic triples (Gardent et al., 2017), which may call for additional steps of signal analysis or data interpretation. In this work, we abstract away from the input modality and focus on the three core generative stages that directly implicate the limitations of the RAG architecture.

We acknowledge that this analysis is primarily conceptual in nature and that the analogies are grounded in the literature rather than empirically validated. Future quantitative work will need to explore task-specific strategies for explicitly encoding classical NLG pipeline stages into RAG, such as prompting-based approaches of few-shot or chain-of-thought, and pipeline-informed retrieval strategies. Our analysis suggests that the NLG pipeline retains significance as a framework for understanding neural architecture, especially RAG, and is a diagnostic tool to identify where it can fail.

Acknowledgments

We thank Ehud Reiter for insightful discussions on the NLG pipeline and LLMs, which helped sharpen the analysis presented here. We thank the anonymous reviewers for their helpful comments, which have improved our paper, and for their encouragement to expand this work in the future. DMH was supported by CRUK grant EDDPJT-May23/100001.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proc. of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations*, pages 150–158.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proc. of the 12th European Workshop on Natural Language Generation (ENLG)*, pages 90–93, Athens, Greece. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2701–2715.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9459–9474.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7052–7063.

François Portet, Ehud Reiter, Jim Hunter, and Somayajulu Sripada. 2007. Automatic generation of textual summaries from neonatal intensive care data. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 227–236. Springer.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proc. of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915.

Ehud Reiter. 2007. An architecture for data-to-text systems. In *Proc. of the 11th European Workshop on Natural Language Generation (ENLG)*, pages 97–104.

Ehud Reiter. 2025. *Natural Language Generation*. Springer.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: A systematic survey of prompt engineering techniques. Preprint, arXiv:2406.06608.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Somayajulu Sripada. 2003. SumTime-Mousam: Configurable marine weather forecast generator. *Expert Update*, 6(3):4–10.

Somayajulu Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2002. Sumtime-meteo: Parallel corpus of naturally occurring forecast texts and weather data. Technical report, Computing Science Department, University of Aberdeen, Aberdeen, Scotland, Tech. Rep. AUCS/TR0201.

Somayajulu G. Sripada, Ehud Reiter, Ian Davy, and Kristian Nilssen. 2004. Lessons from deploying nlg technology for marine weather forecast text generation. In *Proc. of the 16th European Conference on Artificial Intelligence*, ECAI'04, page 760–764, NLD. IOS Press.

Somayajulu G Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2003. Generating english summaries of time series data using the gricean maxims. In *Proc. of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196.

Barkavi Sundararajan, Somayajulu Sripada, and Ehud Reiter. 2025. Input matters: Evaluating input structure's impact on LLM summaries of sports play-by-play. In *Proc. of the 18th International Natural Language Generation Conference*, pages 795–809, Hanoi, Vietnam. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 27.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proc. of the 13th International Conference on Natural Language Generation (INLG)*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.

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

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 39–52

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

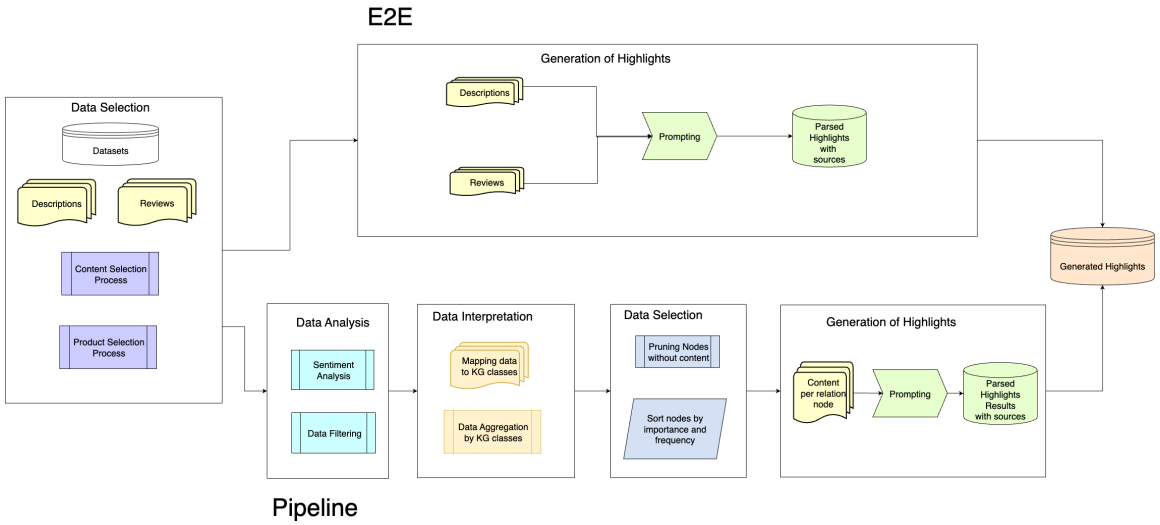

Figure 1: Highlights system architectures for both E2E and PIPE

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× 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 × 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*.

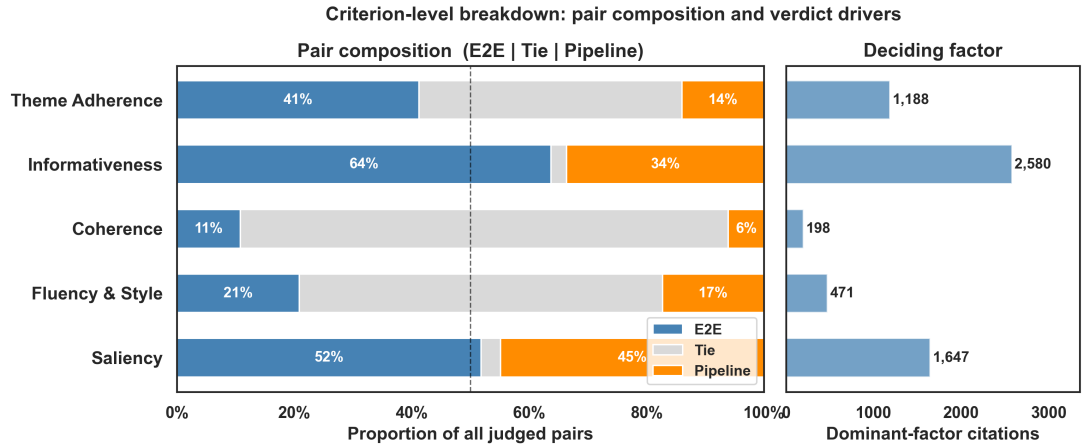

Figure 2: Criterion-level breakdown of judge preferences (E2E vs. PIPE) across all judged pairs. **Left:** proportion of pairs rated as E2E win (blue), tie (grey), or PIPE win (orange) on each evaluation criterion. **Right:** number of times each criterion was cited by the judge as the dominant factor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

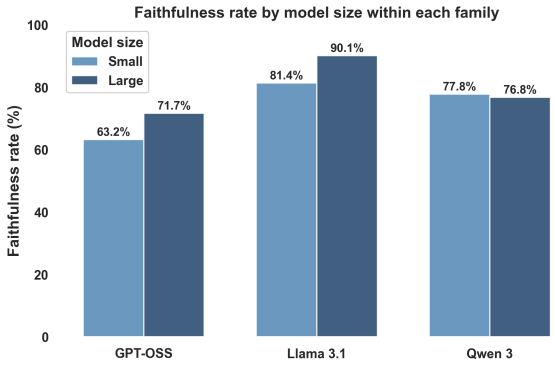

Figure 3: Faithful rate by model size within each family. GPT-OSS and Llama show a clear size benefit; Qwen does not.

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.

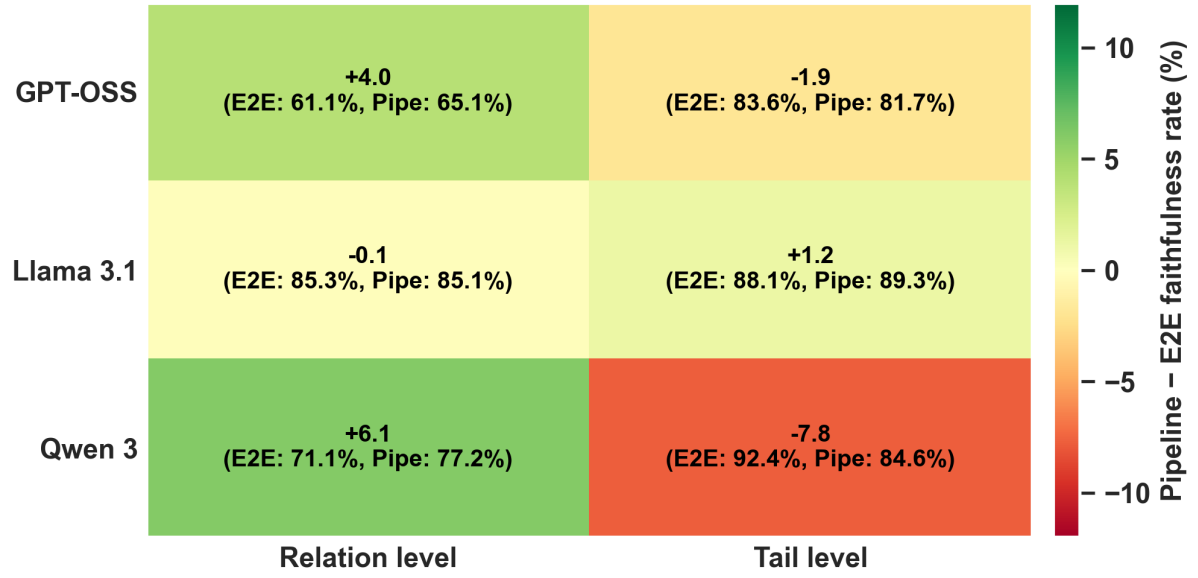

Figure 5: Pipeline minus E2E faithful rate by model family and content level. Green indicates pipeline is more faithful; red indicates E2E is more faithful.

Solving the Task but Not the Problem: A Customer Support Case Study on Why Extrinsic Evaluation Matters

Daniel Braun
Marburg University
Department of Mathematics and Computer Science
daniel.braun@uni-marburg.de

Abstract

Natural Language Processing has long been used in customer support to automate and augment human agents. Despite its long-standing use and clear practical relevance, most scientific evaluations rely on intrinsic evaluations and metrics such as accuracy or F1-score. In this paper, we argue that such evaluations often fail to reflect real-world system impact. We present a case study of an NLP system for email-based customer support evaluated both intrinsically and extrinsically via a before-and-after study in deployment. While the system achieves strong intrinsic performance, we observe no measurable improvement in key operational metrics such as average handle time per email. These results highlight a mismatch between benchmark performance and real-world effectiveness, supporting calls for more systematic extrinsic evaluation of NLP systems.

1 Introduction

Natural Language Processing (NLP) has been successfully applied in customer support for decades, to automate and augment the work of customer support representatives. Early systems focused on rule-based dialogue management and information retrieval, while more recent approaches leverage machine learning and large language models to enable tasks such as intent detection, automated response generation, ticket routing, and conversational assistance (see Section 2). Across these developments, the overarching goal has remained consistent: to improve efficiency, reduce operational costs, and enhance customer experience.

Despite this long-standing application and clear practical relevance, the scientific evaluation of NLP in customer support has been predominantly focused on intrinsic evaluation. Systems are regularly assessed based on measures such as accuracy and F1-score on narrow tasks. (Jones and Galliers, 1995) While such metrics provide insights into model performance on these isolated tasks, they often fail to capture the broader, real-world impact of these systems once deployed.

This limitation is not specific to customer support but reflects a broader issue in NLP research. In *We Should Evaluate Real-World Impact*, Reiter (2025) highlights the lack of extrinsic, real-world evaluation across the field and calls for a shift in evaluation practices. He argues that, if NLP systems are intended to be deployed and provide tangible benefits, it is essential to assess their impact on real-world key performance indicators (KPIs) under production conditions, because intrinsic metrics alone are insufficient proxies for practical success.

In this paper, we present a case study of a real-world NLP system for email-based customer support evaluated both intrinsically on annotated test data and extrinsically in deployment using a before-and-after study. While the system achieves strong intrinsic performance (accuracy 0.85 - 0.90), these results do not translate into improvements in KPIs such as average handle time per email. This discrepancy illustrates that intrinsic evaluation not only provides an incomplete picture of system performance, but can in some cases be a poor predictor of real-world impact altogether. The findings reinforce the need to complement traditional benchmarks with evaluations grounded in practical outcomes, supporting the broader call by Reiter (2025) to also assess the real-world effectiveness of NLP systems.

2 Related Work

NLP has long been applied to customer support for email handling, helpdesk systems, and conversational agents. Early work focused on text classification and information retrieval for support requests, such as automated email categorization (Cohen et al., 2004; Carvalho and Cohen, 2006) and helpdesk call routing (Garfield and Wermter,

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 53–62

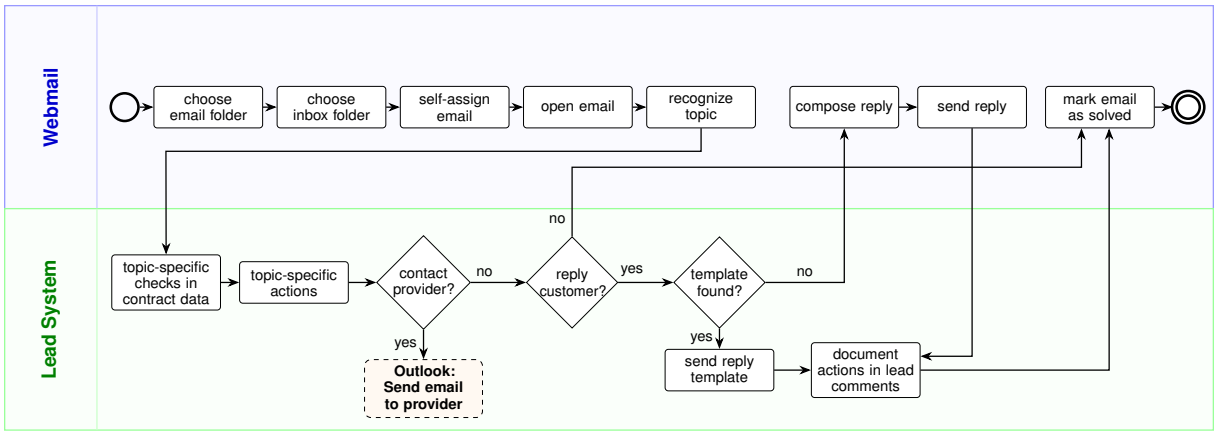

Figure 1: Workflow Customer Support

2002). These systems are typically evaluated using intrinsic metrics such as accuracy, precision, recall, and F1-score on annotated datasets.

More recent approaches leverage neural models and large-scale datasets. For example, Kannan et al. (2016) introduce Smart Reply, a system for automated email response suggestion, mainly evaluated using prediction accuracy and ranking metrics, despite the fact that the system was actually deployed.

Similarly, modern conversational systems (Xu et al., 2017; Hardalov et al., 2018; Farea and Emmert-Streib, 2025; Farnaz and Huyck, 2026) are commonly assessed using benchmark datasets and automatic metrics such as BLEU, dialogue state tracking accuracy, or response selection accuracy. While these evaluations enable comparison across models, they remain largely detached from real-world usage conditions.

A smaller body of work considers extrinsic evaluation in deployed environments. Jain et al. (2018), for example, analyzed aspects like the total interaction time in actual conversations of a chatbot or the count of messages, as well as asking users directly about their satisfaction. Kagan et al. (2025) measure the KPI of chatbot uptake in a context where users can choose between talking to a human customer support representative or a chatbot in a series of A/B tests. Our work adds to this line of research by focusing on the comparison of intrinsic and extrinsic evaluation outcomes, demonstrating that strong intrinsic performance does not necessarily translate into measurable real-world improvements.

3 Business Context

The case study was conducted at a company that brokers and manages energy contracts for consumers. The revenue of the company is generated through commissions for brokered contracts. Therefore, customer support is one of the most important aspects of the business. Due to rapid growth of the company, the service department was struggling to keep up with demand. At the beginning of the collaboration, the customer support of the company received up to 1200 emails per day.

Because the existing software used for handling the emails was based on legacy technology, the company wanted to introduce a new software and in this process also introduce new automation and support features in order to reduce the time that employees spend answering an individual email and reduce the waiting time for customers, particularly for urgent request. We scientifically accompanied the deployment and evaluation of the system.

Through a series of interviews with the head and the deputy-head of the customer service department existing workflow was discovered and formalized. As shown in Figure 1, the customer support mainly relies on two systems in their workflow: a webmail client and a lead system. The webmail client is a simple mail client that is connected to the support email address of the company. Employees log into the system, pick an email, assign themselves to the email, and then take the necessary steps based on the content. The lead system is a CRM system that contains all customer data and information about existing contracts. One noteworthy specificity of the described workflow is that answer templates exist, however they are stored in the lead system and have to be copied manually from the lead system to the webmail client.

In the interviews, five main customer service workflows were identified based on the topic of incoming emails: *data changes* (e.g. updates to ad-

Feature name	NLP	Product
Integrate lead data		X
Integrate lead actions		
Integrate response templates		X
Topic classification	X	X
Lead data augmentation	X	X
Response template suggestion	X	X
Prioritization	X	X
Automatic assignment		
Outlook link for provider contact		X
Sentiment detection		
Lead comments		X
Resubmission		X
Advanced filtering		

Table 1: Implemented (NLP) features in the final software

dress information or electricity meter readings), *revocations* of newly signed contracts, *bonus*-related inquiries (e.g. contractual bonus payments), *status requests* regarding ongoing orders, and contract *cancellations*. An analysis of 1,300 consecutively received emails showed that these topics covered 56% of the incoming mails. Among the other emails, no other frequently (i.e. more than 10 emails) reoccurring topics could be identified.

4 System Design

Through the interview process, 13 new features were identified that could be added to the new system in order to improve operations in the customer service department. Table 1 shows an overview of the features and the nine features that ended up in the final product. Of those features, four are based on NLP models, on which we fill focus. These features are: *topic classification* of incoming emails according to the categories described in Section 3, *linking emails* to the lead database (e.g. via contract or customer ID), automatic *selection of response templates*, and *prioritization* of requests based on their urgency.

After several rounds of pre-experimentation within the company, a decision was made to use a rather simple combination of Tf-idf encoding of incoming emails and their subjects together with a Stochastic Gradient Descent classifier for both the topic and priority classification. For the lead data augmentation, a rule-based system was developed that extracts information like customer IDs, order IDs, invoice numbers, and addresses from incoming emails and matches them against the existing lead data, which will then be displayed in the mail system. Finally, the suggestion of the response templates is based on the identified topics.

Class	Acc.	Prec.	Rec.	F1	Supp.
Bonus	0.975	0.875	0.636	0.737	11
Cancellation	0.938	0.500	0.462	0.480	13
Data Change	0.867	0.826	0.422	0.559	45
Other	0.768	0.751	0.899	0.818	148
Revocation	0.951	0.864	0.731	0.792	26
Status	0.947	0.647	0.688	0.667	16
Total	0.901	0.753	0.753	0.753	259

Table 2: Topic Classification Performance

Class	Acc.	Prec.	Rec.	F1	Supp.
Low	0.855	0.882	0.681	0.769	373
Normal	0.833	0.751	0.798	0.774	386
High	0.858	0.764	0.885	0.820	383
Total	0.849	0.789	0.789	0.789	1142

Table 3: Priority Classification Performance

5 Intrinsic Evaluation

For the intrinsic evaluation of the developed NLP features, a standard evaluation approach was adopted in which real customer emails were annotated by employees, and the models were subsequently evaluated against this annotated data using standard metrics such as accuracy, precision, recall, and F1-score.

5.1 Topic Classification

For the topic classification, a total of 1,295 emails were manually annotated with their respective topic. In addition to the five classes described in Section 3, a sixth class "other" was introduced for all emails that do not fit in any of the classes. 80% of the set was used for training and 20% for testing. The result of the intrinsic evaluation are shown in Table 2, a Table with a confusion matrix can be found in the appendix. Overall, the simple approach performed well in the intrinsic evaluation with an overall accuracy of 0.901.

5.2 Priority Classification

Similarly, for the priority classification, a data set of 5,707 was annotated with three priority classes: low, normal, and high. The set was again split into 80% training and 20% test. The results are shown in Table 2. With an overall accuracy of 0.849, the intrinsic evaluation again revealed good results.

5.3 Information Extraction

Finally, since the information extraction is performed in a rule-based fashion, no training data was needed. Therefore, only a test set, consisting of 107 emails with 254 items of relevant information to be extracted was annotated. The result of

Label	Prec.	Rec.	F1	Support
Invoice Nr	1.00	1.00	1.00	1
City	1.00	0.90	0.95	21
Contract Nr	0.33	0.50	0.40	2
Date	0.91	0.77	0.83	39
Meter Nr	1.00	0.56	0.71	9
Money	1.00	0.65	0.79	23
Order Nr	1.00	1.00	1.00	4
Person	0.94	0.87	0.90	102
ZIP	1.00	0.92	0.96	12
Time	1.00	0.50	0.67	4
Vendor	0.97	0.85	0.91	37
Total	0.95	0.82	0.88	254

Table 4: Information Extraction Performance

the intrinsic evaluation is shown in Table 4.

6 Extrinsic Evaluation

The goal of the extrinsic evaluation was to assess whether the introduced NLP features had an impact on day-to-day operations in the customer support department. We considered two KPIs: average handle time per email, i.e. the time required to respond to an email, and first contact time for high-priority emails, i.e. the time until a customer receives an initial response.

Since the NLP features were introduced together with a new email client, the evaluation was conducted as a before-and-after study in four phases, with an initial eight-week adaptation phase in which employees familiarized themselves with the tool. The introduction of both a new tool and the NLP features could limit the conclusions that can be drawn from a before-and-after study, therefore the generous familirization phase was added.

1. Phase 1: Introduction of the new webmail client (8 weeks)
2. Phase 2: Baseline period without NLP features (10 days)
3. Phase 3: Evaluation period with NLP features (10 days)
4. Phase 4: Post-evaluation without NLP features (10 days)

The fourth phase was introduced because during phase 3 we saw a spike in customer requests compared to phase 2 and we wanted to make sure that effects that we measure between the two phases are not caused by the increased number of requests.

To compute the KPIs, the webmail client was instrumented with logging functionality during phases 2 to 4. The system recorded: when an email was opened, when an email was marked as solved, when a reply was sent, when and which response template was used. In addition, it was logged whether employees modified the automatically assigned topic and priority labels.

6.1 Observations

An average of 486 per Day were opened during Phase 2, 546 during Phase 3 and 677 during Phase 4. In the same period 8,011 emails were sent through the system. Of these replies, 53.84% did not use any of the templates. The increase in email volume between phases could potentially influence processing times. Therefore, as mentioned above, the fourth phase was introduced, so that the mail client without NLP features was used during both a low- and a high-load period.

6.2 Correction of System Predictions

During phase 3, in which the NLP features were activated, more than 10,000 emails were received. For 8,998 emails, the automatically assigned priority was confirmed and for 1,591 it was changed, implying an accuracy of 0.85, which confirms the findings of the intrinsic evaluation. The topic classification was changed 1,545 times and was confirmed 9,321 times, leading to an accuracy of 0.86, which is slightly lower than the result in the intrinsic evaluation, however still on a level that would be considered acceptable.[1]

6.3 Average Handle Time

The average handling time was:

- 3m 19s in phase 2,
- 2m 39s in phase 3, and
- 2m 25s in phase 4.

On a per-user level, we observe substantial variation: some employees remain consistently fast or slow across all phases, while others vary strongly over time. As noted earlier, the email volume increased during the evaluation period, and we observed a negative correlation between workload and handling time, i.e. higher workload is associated with faster responses. Overall, there is no clear evidence that the NLP features had an effect on average handling time.

[1] For some emails one or both predictions were neither actively confirmed nor changed.

Question	Avg. Agreement
Feature-related	
I can orient myself more quickly within an email when a topic has been assigned.	0.26
I generally use the suggested response template when replying to an email.	0.32
I can understand why a particular response template was suggested.	1.16
I generally trust the suggested response template.	0
Comparative	
I see more disadvantages than advantages in the topic recognition feature.	-0.63
I see more disadvantages than advantages in the priority recognition feature.	-0.89
I am more productive with Webmail-B than with Webmail-A.	-0.05
I enjoy working with Webmail-A more than with Webmail-B.	0.32

Table 5: Average agreement to statements from strongly disagree (-2) to strongly agree (2)

6.4 First Contact Time

Finally, the first contact time for high priority emails was reduced in the phase with the NLP features by 11% (from 293 minutes to 260 minutes). Given that the average handle time was not reduced, that meant on the other hand that the first contact time for low priority emails increased, namely from 300 minutes to 491 minutes on average.

7 Survey

After the extrinsic evaluation was concluded, the 19 participating employees received a survey. The survey contained a System Usability Scale (SUS, Vlachogianni and Tselios (2022)) for the client with NLP features (internally named Webmail-A), some questions about specific features, as well as some specific question about the version without NLP features (internally named Webmail-B), and some comparative questions like "With Webmail-B I am more productive than with Webmail-A". The whole survey can be found in Appendix A.

With a SUS score of 70, the NLP-enabled client is slightly above the commonly reported average SUS score of around 68 (Vlachogianni and Tselios, 2022). Given the nature of the SUS items, this result is likely influenced more by the user interface of the newly introduced client than by the underlying NLP features.

While none of the feature-related questions, and thereby the underlying features, were rated negatively, the results indicate only a slightly positive attitude overall (see Table 5). In the comparative question, a stronger signal emerges that the version with NLP features is perceived as overall more helpful and more enjoyable to work with.

Therefore, it is not surprising that, when asked directly, a majority of 59% would prefer to use the new webmail client with NLP features in the future. Only 11% would prefer the new client without NLP features. The remaining 30% would prefer the legacy system.

8 Conclusion

The results of both evaluations show that, despite the fact that NLP features were successfully fulfilling the tasks they were designed to do, the impact on the KPIs was limited, specifically with regard to the most important KPI, the time that is needed to work on customer requests.

There are plenty of potential reasons that can be identified. For example, more than half of all incoming emails fall in the topic category "other", severely limiting the potential impact the topic classification can have, even when working perfectly. This is a fact that was already clear during the design phase, but would have also been highlighted by a purely intrinsic evaluation. Similarly, less than half of the email replies are based on one of the existing template, limiting the potential advantage that the automatic selection of said templates can have.

Nevertheless, had the system been helpful for the remaining half of the emails, as suggested by the intrinsic evaluation, an improvement in the KPIs would still be expected. We believe that this case study illustrates the need for extrinsic evaluations of NLP systems in addition to intrinsic evaluations, as purely intrinsic evaluations are not necessarily good predictors of real-world impact. This is because they do not account for the practical relevance of the selected tasks (e.g., topic classification in this case) within the overall real-world process.

Limitations

This study has several limitations that should be considered when interpreting the results:

- The before-and-after design is susceptible to confounding factors. In particular, changes in workload. Although we reacted to the increase in workload, there was no comparison possible between two phases with the exact same amount of workload.
- The introduction of the NLP features was linked to the introduction of a new webmail client. This introduces additional confounds that may affect user behavior independently of the NLP functionality. We tried to minimize such effects by introducing an eight week period for the employees to familiarise themselves with the new system.
- The evaluation focuses on just two KPIs, namely average handle time and first contact time. While these are important indicators of efficiency, they do not capture other relevant dimensions such as customer satisfaction and response quality.
- Finally, the NLP techniques used in this system are relatively simple. However, despite their simplicity, the techniques proved sufficiently effective in the intrinsic evaluation. Optimizing model architectures or achieving state-of-the-art performance on individual tasks was not the goal of this study, but comparing the results of intrinsic and extrinsic evaluation.

References

Vitor Carvalho and William Cohen. 2006. Improving "email speech acts" analysis via n-gram selection. In *Proceedings of the Analyzing Conversations in Text and Speech*, pages 35–41, New York City, New York. Association for Computational Linguistics.

William W. Cohen, Vitor R. Carvalho, and Tom M. Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 309–316, Barcelona, Spain. Association for Computational Linguistics.

Amer Farea and Frank Emmert-Streib. 2025. Understanding question-answering systems: Evolution, applications, trends, and challenges. *Engineering Applications of Artificial Intelligence*, 156:110997.

Aneela Farnaz and Chris Huyck. 2026. Travquery: A customer support chatbot based on retrieval augmented generation (rag). In *Artificial Intelligence XLII*, pages 130–140, Cham. Springer Nature Switzerland.

Sheila Garfield and Stefan Wermter. 2002. Recurrent neural learning for helpdesk call routing. In *Artificial Neural Networks — ICANN 2002*, pages 296–301, Berlin, Heidelberg. Springer Berlin Heidelberg.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards automated customer support. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 48–59, Cham. Springer International Publishing.

Mohit Jain, Pratyush Kumar, Ramachandra Kota, and Shwetak N. Patel. 2018. Evaluating and informing the design of chatbots. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, page 895–906, New York, NY, USA. Association for Computing Machinery.

Karen Sparck Jones and Julia R Galliers. 1995. Evaluating natural language processing systems: An analysis and review.

Evgeny Kagan, Brett Hathaway, and Maqbool Dada. 2025. Deploying chatbots in customer service: Adoption hurdles and simple remedies. *arXiv preprint arXiv:2504.06145*.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 955–964, New York, NY, USA. Association for Computing Machinery.

Ehud Reiter. 2025. We should evaluate real-world impact. *Computational Linguistics*, 51(4):1419–1431.

Prokopia Vlachogianni and Nikolaos Tselios. 2022. Perceived usability evaluation of educational technology using the system usability scale (sus): A systematic review. *Journal of Research on Technology in Education*, 54(3):392–409.

Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3506–3510, New York, NY, USA. Association for Computing Machinery.

A Survey

A.1 Webmail-A

This section concerns only the current version, *Webmail-A*.

1. I think that I would like to use the system frequently.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

2. I found the system unnecessarily complex.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

3. I found the system easy to use.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

4. I think that I would need the support of a technically skilled person to use the system.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

5. I found the various functions in this system to be well integrated.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

6. I think there was too much inconsistency in the system.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

7. I imagine that most people would learn to use this system very quickly.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

8. I found the system very cumbersome to use.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

9. I felt very confident using the system.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

10. I needed to learn a lot before I could get going with the system.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

11. On a scale from 0 to 10, how accurate is the content recognition for *priority*?
(10 = very accurate, 0 = very inaccurate)

_____

12. On a scale from 0 to 10, how accurate is the content recognition for *topic*?
(10 = very accurate, 0 = very inaccurate)

_____

13. How easy is it for you to read an email in Webmail-A?

Very Difficult						Very Easy
☐	☐	☐	☐	☐	☐	☐

14. How easy is it for you to reply to an email in Webmail-A?

Very Difficult						Very Easy
☐	☐	☐	☐	☐	☐	☐

A.1.1 If the Topic of an Email Was Correctly Recognized

1. I can orient myself more quickly within an email when a topic has been assigned.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

2. I generally use the suggested response template when replying to an email.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

3. I can understand why a particular response template was suggested.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

4. I generally trust the suggested response template.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

A.1.2 If the Topic of an Email Was Incorrectly Recognized

1. I lose time because I expect a different context.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

2. I lose time because I manually correct the incorrect topic.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

3. The consequently incorrect response template does not affect the time required to compose a reply.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

A.2 Webmail-B

This section concerns only *Webmail-B*.

1. How easy is it for you to read an email in Webmail-B?

Very Difficult						Very Easy
☐	☐	☐	☐	☐	☐	☐

2. How easy is it for you to reply to an email in Webmail-B?

Very Difficult						Very Easy
☐	☐	☐	☐	☐	☐	☐

A.3 Final Questions

1. I am more productive with Webmail-B than with Webmail-A.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

2. I enjoy working with Webmail-A more than with Webmail-B.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

3. If you had to choose, which webmail variant would you prefer?

   - ○ Webmail-A
   - ○ Webmail-B
   - ○ The previous version

4. I see more disadvantages than advantages in the topic recognition feature.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

5. Same question, assuming the recognition accuracy were higher.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

6. I see more disadvantages than advantages in the priority recognition feature.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

7. Same question, assuming the recognition accuracy were higher.

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
☐	☐	☐	☐	☐

8. Which functions would you keep activated in Webmail-A?

   - ☐ Priority column in the folder view

- ☐ Topic column in the folder view
- ☐ Automatic selection of a response template
- ☐ Color highlighting of relevant lead data
- ☐ Direct provider contact through the Outlook button

9. How often do you use the webmail system?

   - ○ Continuously
   - ○ A few times per day
   - ○ A few times per week

Never Truly Out of Fashion: A Retrospective Look at Evaluation in NLG

Patrícia Schmidtová[1] ✉, **Saad Mahamood**[2], and **Ondřej Dušek**[1]

[1]Charles University, Faculty of Mathematics and Physics, Prague, Czechia
[2]Shopware, Düsseldorf, Germany
✉ Corresponding author: schmidtova@ufal.mff.cuni.cz

Abstract

Human evaluation (HE) remains the gold standard for assessing natural language generation (NLG) systems, yet automatic metrics are cheaper and faster, creating mounting pressure to skip it. We ask how evaluation practices have changed as NLG research scales. We analyse 24,291 papers from the ACL Anthology (1952–2025) through regular-expression-powered keyword analysis. Before 1990, the majority of NLG papers reported no evaluation at all; today, evaluation is near-universal and HE has held broadly stable over the past decade, despite the rapid emergence of large language model (LLM) judges (referred to as LLM-as-a-judge) since 2023. However, while LLM judges currently serve predominantly as a complement rather than a full substitute for human evaluation, a substantial share of papers already use them without any human validation. Faithfulness has become the fastest-rising evaluation criterion since 2020, coming back into fashion after almost 15 years of decline, tracking the prominence of hallucination research, while criteria such as grammaticality and fluency are receding, suggesting these qualities may increasingly be taken for granted as model outputs improve. Our findings provide a longitudinal baseline for tracking where the field stands.

1 Introduction

The ACL Anthology[1] now comprises over 120,000 papers and keeps growing at an unprecedented rate. Amid this pressure for volume, evaluation, which is the primary mechanism to certify the field's progress, risks becoming a formality rather than a guarantee. This has led to researchers using a plethora of automatic metrics inappropriately (Schmidtova et al., 2024), while at the same time human evaluations remain a fraction of conducted evaluations and struggle with methodological shortcomings (Howcroft et al., 2020; van der Lee et al., 2021). With the ever increasing number of new large language models (LLMs) and the prevalence of using LLMs to evaluate generated text, the incentive to reach for automatic approaches to evaluation has never been stronger (Gehrmann et al., 2023). Crucially, model development and iterative optimization generally require cheap, scalable automatic metrics for rapid feedback in the development loop, since conducting human evaluations at each iteration is logistically and financially impractical. Automatic metrics are thus a structural necessity in modern NLG engineering, even if human evaluation through the use of task-based evaluations remains the ultimate arbiter of quality.

Yet the field has navigated similar tensions before. Debates over the adequacy of BLEU (Papineni et al., 2002) for machine translation (Freitag et al., 2022; Reiter, 2018), the broader question of metric validity in NLG (Stent et al., 2005; Belz and Reiter, 2006; Reiter and Belz, 2009; Novikova et al., 2017), and the gradual consensus around structured evaluation protocols such as Direct Assessment and MQM (Freitag et al., 2022; Belz et al., 2020) all point to a community that periodically pauses, reflects, and self-corrects. This work offers such reflection through a quantitative analysis of evaluation trends across seven decades of the ACL Anthology. Our perspective is informed by a long line of work arguing that evaluation in NLG must be taken more seriously – from early calls for human-centred and realistic assessment (Reiter and Belz, 2009; Reiter, 2011) to recent evidence that the community still devotes vanishingly little attention on real-world impact evaluations (Reiter, 2025).

We use a corpus of 24,291 NLG system papers from the ACL Anthology (1952–2025)[2] to exam-

[1] https://aclanthology.org/

[2] See Section 4 for details on data availability.

Proceedings of the 1st Symposium on Natural Language Generation Evaluations, pages 63–72
June 1-2, 2026

ine: (1) how human evaluation (HE) prevalence and annotation methodology have evolved across venues and years; (2) whether LLM-as-a-judge is displacing human evaluation; (3) which evaluation criteria have risen, faded, or returned after periods of absence; (4) how the balance between automatic, human, and LLM-judge evaluation has shifted; (5) whether evaluation practices differ systematically by NLG task; or (6) which evaluation methodologies were popular.

2 Methodology

We construct a large-scale analysis pipeline over a corpus of NLP research papers, using keyword-based signal detection over available paper text (title/abstract and, where available, full text). An LLM-based extraction experiment proved insufficiently reliable at scale and is reported only in Appendix A. Our regular expressions prioritise precision; pattern details are listed in Appendix A.

2.1 Corpus Assembly

For papers published up to and partially including 2022, we use the ACL Anthology Open Research Corpus (ACL-OCL; Rohatgi et al., 2022),[3] comprising approximately 70,587 full-text articles across 215 venues. For 2022 onwards, we supplement with our own crawl of the ACL Anthology, restricted to nine core venues (ACL, EMNLP, NAACL, EACL, AACL, IJCNLP, INLG, TACL, CL) due to the cost of large-scale PDF processing.[4] After deduplication, the combined corpus comprises **85,792 papers** (1952–2025).

2.2 Paper Filtering

We apply a cascaded filter to retain papers that (a) are not meta-papers (surveys, tutorials, proceedings prefaces), (b) perform a generative NLP task such as MT, summarisation, dialogue, data-to-text, or question generation, matched against the full text of the paper, and (c) contain at least one mention of evaluation in the full text. Evaluation signals fall into three categories: *human evaluation* (mentions of human/manual evaluation, annotation studies, crowdsourcing, named protocols such as Direct Assessment and MQM); *automatic evaluation* (e.g., BLEU, ROUGE, BERTScore, COMET); and *LLM-as-a-judge* (mentions of situations an LLM evaluates system outputs). This procedure yields a final analysis corpus of **24,291 papers** (see Table 1 in the Appendix).

2.3 Analysis

All analyses are conducted at the paper level, with year and venue as primary grouping variables and task type as a secondary variable. Evaluation criteria are detected via keyword matching over full paper text using patterns derived from the taxonomy proposed by Howcroft et al. (2020).

3 Results

Human evaluation prevalence and trajectory. Human evaluation has been remarkably stable over the past decade: the proportion of NLG papers reporting at least one human evaluation has fluctuated between 34% and 42% every year since 2018, with no sustained decline despite the growing availability of automatic metrics and LLM judges. In absolute terms, the number of human-evaluated papers has grown substantially – from around 550 per year in 2018–2019 to over 650 in 2025 – but this tracks the overall growth of the field rather than a shift in evaluation culture. A logistic regression over the recent decade (2015–2025) confirms the absence of any statistically significant trend in the proportion of papers using human evaluation ($\beta = 0.0023, p = 0.69$), statistically validating this stability. The proportion of papers at core, general NLP venues closely mirrors the corpus-wide trend, though specialised generation venues such as the International Conference on Natural Language Generation (INLG) show consistently higher human evaluation rates: 53.1% overall, compared to 36.0% in core NLP venues (see Appendix B).

LLM-as-a-judge adoption. The adoption of LLM-as-a-judge has grown exponentially since 2022 (Bavaresco et al., 2025): 109 papers in 2024 and 224 in 2025, compared to 19 in 2023 and zero before. This rapid growth is highly statistically significant under logistic regression over 2020–2025 ($\beta = 0.9523, p = 4.25 \times 10^{-76}$, odds ratio $e^{0.9523} \approx 2.59$ per year). Earlier papers generally used either custom-finetuned encoder-only models, such as RoBERTa (Liu et al., 2019), for classification, or GPT-2 (Radford et al., 2019) to measure perplexity. Among papers using an instruction-tuned LLM judge, 61.4% report human evaluation as well, and 86.2% of these also explicitly mention validation or manual checking of the LLM outputs.

[3] https://huggingface.co/datasets/ACL-OCL/ACL-OCL-Corpus

[4] This venue asymmetry means that post-2022 comparisons involving smaller venues should be interpreted with caution.

This strong baseline suggests that LLM judges are currently viewed predominantly as complements to human evaluation rather than full substitutes. However, it also means that 38.6% of papers employing an LLM judge do so without any reported human evaluation, a share that bears watching as the paradigm matures. Adoption rates vary across tasks, with newer or long-form generation tasks showing the highest prevalence: story generation leads with 9.0% of its papers adopting LLM judges (9 papers), followed by code generation at 8.3% (15 papers), data-to-text at 3.2% (14 papers), and question generation at 3.1% (11 papers). In contrast, established tasks with massive historical paper volumes show much lower adoption rates, such as machine translation at 0.5% (51 papers), paraphrase generation at 1.4% (14 papers), and summarization at 2.2% (46 papers).

Human evaluation criteria. We detect evaluation criteria via keyword matching over full paper text using category names from the Howcroft et al. (2020) taxonomy, counting each criterion at most once per paper. As Howcroft et al. (2020) emphasise, the same term can carry different meanings across papers; our counts thus reflect terminology adoption rather than consistent operationalisation. Among 8,894 human-evaluation papers, fluency (22.5%), relevance (21.6%), and coherence (16.6%) are the most common. The apparent frequency of "accuracy" (52.8% of papers) is likely an artifact: the term is used loosely across many NLG papers to describe model performance broadly, exemplifying the terminological confusion highlighted by Howcroft et al. (2020). We retain it in Figure 1 but caution against interpreting its trend as reflecting deliberate evaluation design.

Figure 1 reveals genuinely cyclical patterns. Faithfulness (Maynez et al., 2020) follows a U-shaped trajectory: used in early NLG and MT evaluation, largely absent through the 2010s as automatic metrics displaced human assessment, and now rising sharply from 1.7% of HE papers in 2015 to 29.8% in 2025 ($\beta = 0.3329, p = 2.39 \times 10^{-94}$ under logistic regression over 2015–2025), driven by hallucination research (Schmidtova et al., 2025). Adequacy and grammaticality are on a declining arc, while relevance, consistency, and coherence show cyclical recoveries.

Evaluation modality mix. Automatic-only evaluation is by far the most common modality (57.8%) and has grown as a share of the corpus over time. Before 2000 – and largely before BLEU (Papineni et al., 2002) – automatic-only evaluation accounted for roughly 30–55% of papers per period, with human-only evaluation reaching 10–22%; by 2020–2025, automatic-only has stabilised around 55% while human-only has fallen to 2–3%, and combined human-and-automatic evaluation has risen from under 10% to over 30%. The near-absence of human-only evaluation today indicates that automatic metrics have become a de facto prerequisite: even papers that invest in human judgement almost always report automatic scores alongside them. LLM-judge-only papers remain a small fraction but are growing rapidly, rising from near-zero in the 2020 period to 5.5% of papers using an LLM judge without any human evaluation in 2025. Of those, roughly a half use API-only models from OpenAI, which so far seem to outperform open-weight models in correlation with human judgement (Huang et al., 2025); however, they raise questions about the reproducibility of such evaluations (Schroeder and Wood-Doughty, 2025). Evaluation modality also varies by venue group, with SIGGEN venues showing notably higher human evaluation rates than core NLP venues; we report the full breakdown in Appendix B.

Task-based differences. Human evaluation rates vary across NLG tasks, though less dramatically than one might expect: question generation has the highest rate at roughly 53%, while most other tasks cluster between 37% and 47%. Machine translation is the clear outlier at around 28% overall, reflecting its uniquely mature automatic metric ecosystem. However, MT's recent human evaluation rate is closer to 30%, with the lower overall average pulled down by early decades when human evaluation of MT was rarer. Indeed, the MT community has developed a notably rigorous meta-evaluation tradition – shared tasks such as WMT provide large-scale human annotations that calibrate automatic metrics against human judgements (Freitag et al., 2022; Zouhar et al., 2024). No comparable infrastructure exists for most other NLG tasks; establishing it would be a concrete step toward the same level of metric accountability.

Annotation methodologies. Among all 8,894 human-evaluated papers, Likert-scale rating is the most frequently detected annotation approach (19.0%), followed by post-editing (15.8%), binary and multi-class categorisation (13.9%), and ranking/pairwise comparison (13.2%). Error span an-

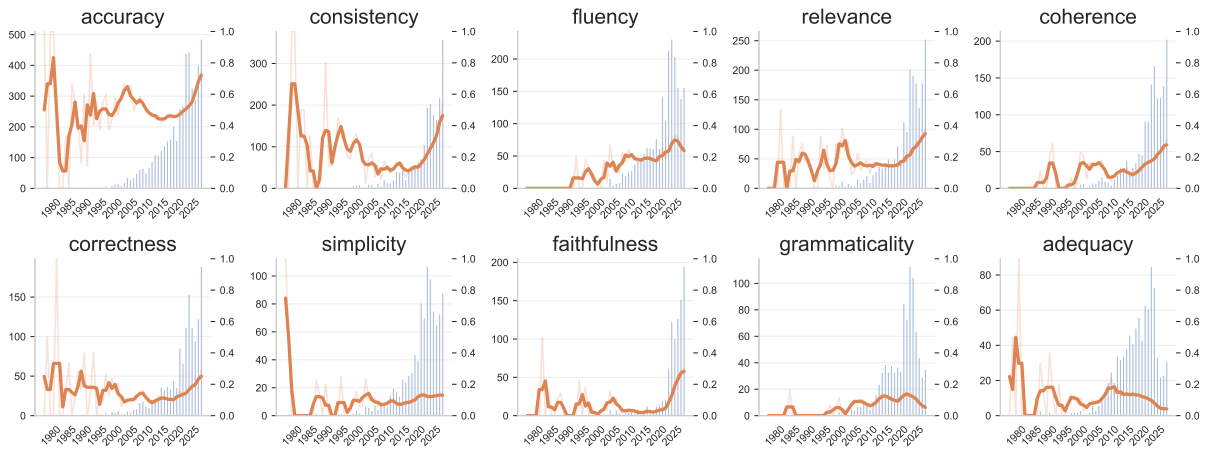

Figure 1: Prevalence of top 10 human-evaluation criteria over time. Left Y-axis: annual paper counts (bars); right Y-axis: proportion of HE papers mentioning the criterion (thick line: 3-year rolling mean; faint line: raw annual proportion).

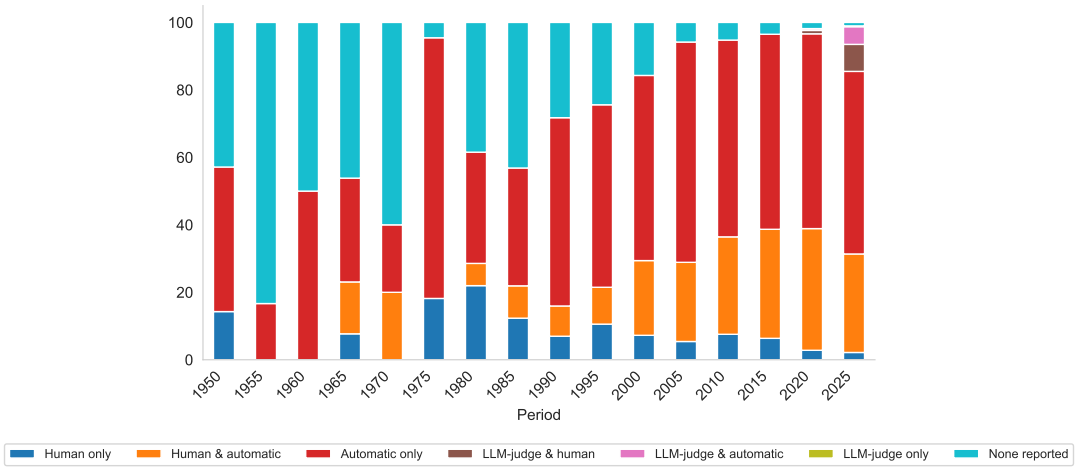

Figure 2: Evaluation modality mix by 5-year period (1950–2025) as a percentage of papers (Y-axis), showing the proportion relying on human-only, automatic-only, combined human-and-automatic evaluation, or LLM-as-a-judge approaches.

notation (4.0%), Direct Assessment (Graham et al., 2013) (2.3%), and Best-Worst Scaling (Kiritchenko and Mohammad, 2017) (0.8%) round out the detected methods. While we attempted to extract fine-grained metadata regarding reporting quality, specifically inter-annotator agreement (IAA), annotator counts, and rated item counts, these variables could not be reliably validated due to low recall and reporting inconsistencies, and are therefore omitted from our quantitative analysis. Best-Worst Scaling and error span annotation remain specialised, concentrated in MT and reference-free evaluation.

Method choices also shift over time (Figure 3): across all papers, post-editing was once the most common methodology, peaking in popularity around 2014–2015 when it was used in 28.6% of all human-evaluated papers (and 43.3% in machine translation). Over the last decade, however, its use has declined sharply, dropping to just 6.1% in 2025 (and 17.7% in machine translation specifically). In contrast, Likert-scale rating has experienced substantial growth, rising from 13.8% in 2015 to become the most prevalent approach at 28.7% in 2025. Additionally, error span annotation has steadily gained traction in recent years, growing from 3.7% in 2015 to 8.0% in 2025, reflecting a growing interest in fine-grained evaluation. We show the longitudinal breakdown of these methodology trends in Figure 3.

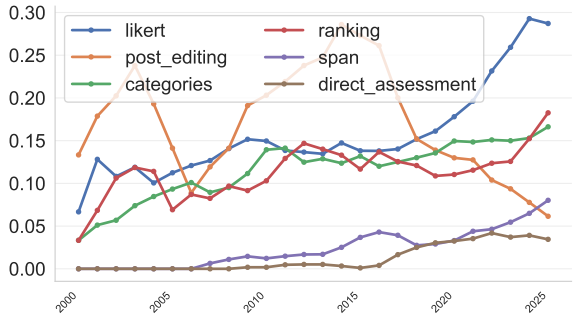

Figure 3: Prevalence of top human annotation methodologies over time, expressed as the fraction of human-evaluated papers on the Y-axis (3-year rolling mean).

4 Conclusions

We have presented a large-scale longitudinal analysis of NLG evaluation practices across 24,291 papers from the ACL Anthology spanning seven decades. The field has come a long way: before 1990, between 38% and 83% of NLG papers in each five-year period reported no evaluation at all, and the share only dropped below 20% after 2005. Today, evaluation is near-universal and human evaluation has held roughly steady at 35–45% of NLG papers over the past decade, defying predictions of decline (van der Lee et al., 2021; Reiter, 2025) even as it shows no upward trend despite repeated calls for more rigorous assessment.

Three findings stand out as actionable:

LLM judges need calibration infrastructure. LLM-as-a-judge has become standard practice in under two years: 109 papers in 2024 and 224 in 2025, up from near-zero before 2023. While 61.4% of these papers also report human evaluation, the remaining 38.6% rely on LLM judgements without any reported human validation. This mirrors earlier episodes the field later had to revisit, such as the widespread adoption of BLEU without adequate validation or the reliance on single-reference translation evaluation, and runs the risk of treating LLM judgements as a trusted signal before their failure modes are understood. We recommend that venues encourage, and reviewers expect, explicit calibration of LLM judges against human judgements, particularly for tasks and domains where no prior calibration exists.

Faithfulness evaluation needs standardisation. Faithfulness has become the fastest-growing evaluation criterion in our data and reflecting the centrality of hallucination as a research problem (Maynez et al., 2020; Schmidtova et al., 2025). Yet the broader picture is less encouraging: "accuracy" appears in over 50% of human-evaluation papers, used so loosely that it functions less as a defined criterion than as a catch-all for correctness (Howcroft et al., 2020). The risk is that faithfulness, as it matures, drifts toward the same terminological vagueness. Establishing shared annotation tasks for faithfulness – analogous to what WMT provides for translation quality via Direct Assessment and MQM (Freitag et al., 2022) – would help prevent this and provide the calibration infrastructure that other NLG tasks currently lack.

Some things never go out of fashion. The cyclical patterns in our data – faithfulness returning after a decade of absence, coherence and relevance waxing and waning with shifts in task popularity – suggest that the field's evaluation vocabulary is smaller and more stable than its rapid growth might imply. These questions recur because they reflect enduring properties of language use, not passing methodological fashions. In a similar spirit, Reiter (2024) argues that rule-based NLG systems retain lasting value even in an era dominated by neural methods – a reminder that what matters is fitness for purpose, not novelty. This principle extends to evaluation: human judgement, automatic metrics, and now LLM judges each have their place, and the field's task is not to choose among them but to understand when each is trustworthy.

One purpose of this paper is to document the current state of affairs and provide a baseline against which future shifts can be measured – whether toward a deeper LLM-judge reliance, new evaluation criteria, or a renewed emphasis on real-world impact (Reiter, 2025).

Our code and dataset (containing paper IDs, metadata, and extracted evaluation signals) are publicly available at `https://github.com/patuchen/trends_in_nlg_eval`. To respect publisher copyrights, the shared dataset does not include the raw full texts of the papers; instead, we provide the metadata, IDs, and extracted signals in a CSV file, along with utility scripts in the repository to automatically download and reconstruct the full-text corpus directly from the official ACL Anthology using the paper IDs.

Limitations

Methodological limitations Our measurements rely on keyword-based detection, which can yield

false positives when terms occur in related work or background sections and false negatives when papers use unusual terminology. We apply the same patterns to all available text (title/abstract and, when available, full text).

We deliberately use keyword-matching over LLM-based extraction due to initial experiments showing that LLMs had problematically low recall when extracting fine-grained methodological metadata at scale (see Appendix A). Consequently, we developed an extensive quality assurance procedure for our regex patterns, manually auditing samples to prevent false positives while balancing recall (100 test sentences for every regular expression). By aggressively stripping meta-references and generic terminology, our reported methodology counts necessarily represent a conservative lower bound of actual evaluation practices. We validated this regex-based approach on a manually curated sheet of 60 human-annotated papers, where 51 were successfully mapped to our corpus, yielding a recall of 82.4% after applying targeted pattern overrides.

Finally, our LLM-as-judge classification relies on model name regex matching. Pre-2023 uses of large models as evaluators (e.g. GPT-2 perplexity scoring, adversarial BERT discriminators) are conceptually distinct from the modern instruction-tuned paradigm; we distinguish these as *ml_evaluator* vs. *instruction_llm*, but it is possible that borderline cases remain.

Venue coverage asymmetry. Our post-2022 corpus covers only nine major venues, excluding workshops and smaller conferences that are present in the pre-2022 ACL-OCL data. Since workshop papers tend to report human evaluation less frequently, the post-2022 human evaluation rates may be slightly overestimated relative to earlier years where the full breadth of venues is represented. The most important venues for NLG research are covered throughout, but longitudinal trends should be interpreted with this asymmetry in mind.

Task coverage. Our generative task filter covers a curated set of named NLG tasks (see Appendix A) rather than all possible forms of natural language generation. While this set includes the largest task categories – machine translation alone accounts for over half the analysis corpus – emerging or niche generation tasks may have been omitted from the analysis.

Acknowledgments

This research was co-funded by the European Union (ERC, NG-NLG, 101039303) and by Charles University projects GAUK 252986 and SVV project 260 821.

References

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Anja Belz and Ehud Reiter. 2006. Comparing Automatic and Human Evaluation of NLG Systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 313–320, Trento, Italy. Association for Computational Linguistics.

Anya Belz, Simon Mille, and David M Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. Association for Computational Linguistics (ACL).

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research*, 77:103–166.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg,

Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria. Association for Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ehud Reiter. 2011. Task-based evaluation of NLG systems: Control vs real-world context. In *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*, pages 28–32, Edinburgh, Scotland. Association for Computational Linguistics.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter. 2024. *Natural Language Generation*. Springer Nature.

Ehud Reiter. 2025. We should evaluate real-world impact. *Computational Linguistics*, pages 1–13.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Shaurya Rohatgi, Yanxia Qin, Benjamin Aw, Niranjana Unnithan, and Min-Yen Kan. 2022. The acl ocl corpus: advancing open science in computational linguistics. arXiv.

Patricia Schmidtova, Eduardo Calò, Simone Balloccu, Dimitra Gkatzia, Rudali Huidrom, Mateusz Lango, Fahime Same, Vilém Zouhar, Saad Mahamood, and Ondrej Dusek. 2025. Do my eyes deceive me? a survey of human evaluations of hallucinations in NLG. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 60–79, Hanoi, Vietnam. Association for Computational Linguistics.

Patricia Schmidtova, Saad Mahamood, Simone Balloccu, Ondrej Dusek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondrej Platek, and Adarsa Sivaprasad. 2024. Automatic metrics in natural language generation: A survey of current evaluation practices. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 557–583, Tokyo, Japan. Association for Computational Linguistics.

Kayla Schroeder and Zach Wood-Doughty. 2025. Can you trust llm judgments? reliability of llm-as-a-judge. *Preprint*, arXiv:2412.12509.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 341–351, Mexico City, Mexico. Springer.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Vilém Zouhar, Pinzhen Chen, Tsz Kin Lam, Nikita Moghe, and Barry Haddow. 2024. Pitfalls and outlooks in using COMET. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1272–1288, Miami, Florida, USA. Association for Computational Linguistics.

A Pipeline Details

Corpus statistics. Table 1 summarises the filtering stages.

Stage	Remaining	Dropped
Raw corpus (combined)	85,792	–
Meta-paper exclusion	81,953	3,839
Generative task filter	20,593	61,360
Evaluation signal filter	16,587	4,006
Full-text task pass	24,291	–

Table 1: Corpus size at each pipeline stage.

Task categories. The task filter recognises: *machine_translation*, *summarization*, *dialogue*, *data_to_text*, *question_generation*, *paraphrase*, *simplification*, *captioning*, *general_nlg*, *question_answering* (open-ended/generative QA), *instruction_following*, *counterspeech*, and *highlight_generation*. All of the code and the regular expressions are available in our public repository at https://github.com/patuchen/trends_in_nlg_eval.

Evaluation signal patterns. Human evaluation is detected by expressions such as *human eval**, *manual eval**, *annotators*, *inter-annotator*, *crowdsourc**, *Mechanical Turk*, *Direct Assessment*, *MQM*, and *Likert*. Automatic evaluation is detected by metric names including *BLEU*, *ROUGE*, *METEOR*, *chrF*, *BERTScore*, *BLEURT*, and *COMET*. LLM-as-a-judge is detected by patterns matching an LLM name followed by evaluation-adjacent verbs (e.g. *GPT-4 evaluates*, *Claude rates*).

LLM extraction experiment. To test whether fine-grained metadata could be extracted automatically, we prompted Qwen2.5-14B-Instruct (Team, 2024) (served via vLLM; Kwon et al., 2023) to extract structured evaluation records from isolated evaluation sections. Comparison against expert annotations revealed low recall for fields such as annotator counts and rated item counts, motivating our reliance on keyword-based signals for all reported results.

B Venue-Group differences.

Core NLP venues are the most automatic-only (60.6% auto-only; 33.1% human+auto), while SIGGEN venues show a more mixed profile (35.8% auto-only; 42.4% human+auto; 10.8% human-only). Journals resemble core venues in relying primarily on automatic metrics (50.8% auto-only; 40.3% human+auto), but show zero cases of LLM-as-a-judge evaluation without human evaluation, consistent with slower uptake of the judge-only pattern. See Figure 4 for the complete visual breakdown across all venue groups.

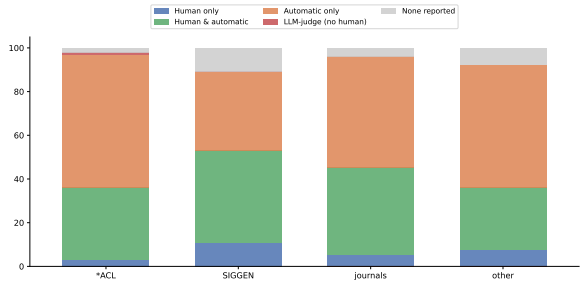

Figure 4: Evaluation modality profile by venue group, shown as a percentage of papers within each group on the Y-axis.

C Phrasing Examples of Historical NLG Papers Without Human Evaluation

For papers in earlier decades that reported no formal human evaluation or relied on informal, qualitative assessments of system output, the following examples illustrate typical phrasings:

- **1976:** *"We choose the interpretation showing the preferable matching of nouns and case by using an evaluation function below which has been established empirically"*
- **1984:** *"This translation is called by its authors as word-by-word, turn-by-turn one; several years have already passed in a complete satisfaction of the customers."*
- **1986:** *"The implementations of the whole system has already been completed and the translation results (10,000 sentences) are now being evaluated by professional translators and native speakers of English. The evaluation results obtained by now are quite satisfactory."*

D Human Evaluation Criteria by Task

Figure 5 shows a detailed heatmap breakdown of the top 10 human evaluation criteria across the ten most common NLG tasks in single-task papers.

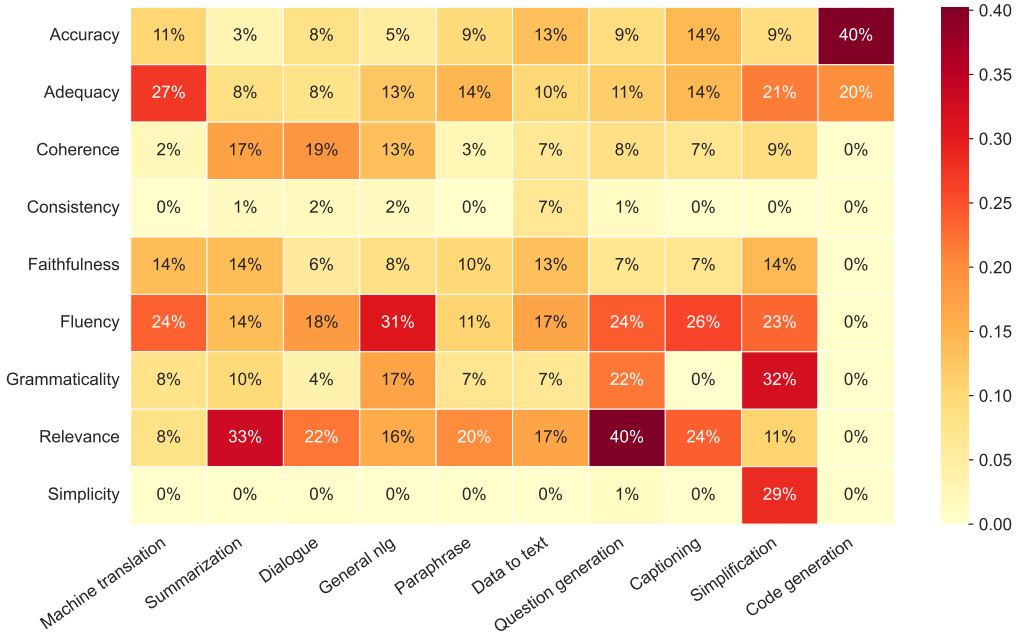

Figure 5: Prevalence of human evaluation criteria (listed on the Y-axis) across the top 10 single-task NLG tasks (listed on the X-axis), expressed as a percentage of papers per task that perform human evaluation.

Author Index