Towards Better Open-Ended Text Generation: A Multicriteria Evaluation Framework

Esteban Garces Arias[1,2], **Hannah Blocher**[1], **Julian Rodemann**[1], **Meimingwei Li**[1], **Christian Heumann**[1], **Matthias Aßenmacher**[1,2]

[1]Department of Statistics, LMU Munich,
[2]Munich Center for Machine Learning (MCML)

Correspondence: Esteban.GarcesArias@stat.uni-muenchen.de

Abstract

Open-ended text generation has become a prominent task in natural language processing due to the rise of powerful (large) language models. However, evaluating the quality of these models and the employed decoding strategies remains challenging due to trade-offs among widely used metrics such as coherence, diversity, and perplexity. This paper addresses the specific problem of multicriteria evaluation for open-ended text generation, proposing novel methods for both relative and absolute rankings of decoding methods. Specifically, we employ benchmarking approaches based on partial orderings and present a new summary metric to balance existing automatic indicators, providing a more holistic evaluation of text generation quality. Our experiments demonstrate that the proposed approaches offer a robust way to compare decoding strategies and serve as valuable tools to guide model selection for open-ended text generation tasks. We suggest future directions for improving evaluation methodologies in text generation and make our code, datasets, and models publicly available.[1]

1 Introduction

Large language models (LLMs, e.g., Dubey et al., 2024; Yang et al., 2024) have demonstrated remarkable capabilities in generating coherent and contextually appropriate text across diverse domains. However, the quality of LLM outputs is fundamentally determined not only by the underlying model architecture but also by the decoding strategies employed during inference—the algorithms that transform the model's output probability distributions into actual text sequences. As the landscape of both LLMs and decoding strategies continues to expand rapidly, the need for robust evaluation frameworks has become increasingly critical (Wiher et al., 2022; Garces-Arias et al., 2025).

[1]https://github.com/YecanLee/2BeOETG

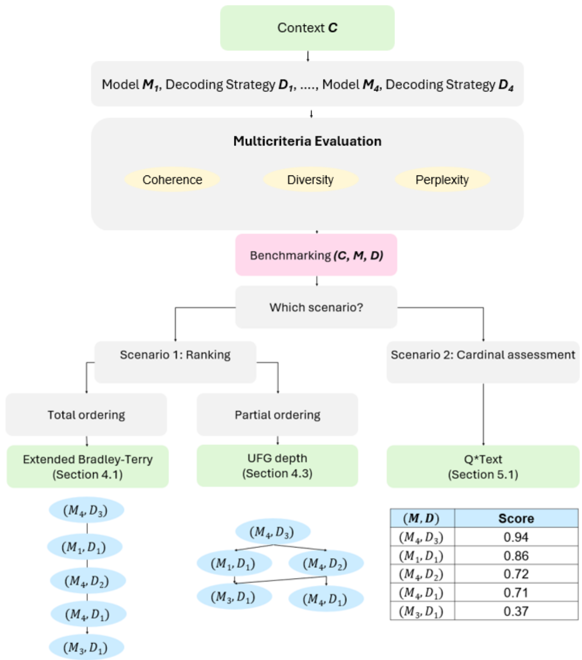

Figure 1: **Multicriteria evaluation framework** for benchmarking models and decoding strategies, i.e., *decoding methods*. We distinguish two scenarios for benchmarking (§1) and two ranking objectives (§4), giving rise to three use-case tailored, distinct methods (§4.1, 4.3 and 5).

Scope and Problem Definition. This paper specifically addresses the challenge of multicriteria evaluation in open-ended text generation, where we must simultaneously consider multiple, often conflicting quality dimensions (Holtzman et al., 2019; Su and Xu, 2022). We focus on developing principled methods for both relative and absolute rankings of decoding methods. Our approach centers on a subset of automatic evaluation metrics—coherence, diversity, and generation perplexity—that capture fundamental trade-offs in text generation quality. While numerous other metrics exist (e.g., relevance, informativeness, style consistency), we deliberately limit our scope to these three core dimensions to establish a foundational

Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM² 2025), pages 631–654
July 31 - August 1, 2025

framework that can be systematically extended.

Current evaluation approaches face remarkable limitations when assessing the quality of text generations within this multicriteria context. Traditional methods typically rely on either human judgments—considered the gold standard, but resource-intensive, and dependent on carefully designed protocols (Howcroft et al., 2020; van der Lee et al., 2021; Karpinska et al., 2021; Ruan et al., 2024)—or individual automatic metrics. While automatic metrics such as MAUVE (Pillutla et al., 2021), coherence (Su et al., 2022), diversity, and generation perplexity (Jelinek et al., 2005) provide valuable insights into specific aspects of generation quality, an isolated consideration of these measures offers only an incomplete perspective on overall performance and fails to address the fundamental multicriteria nature of the evaluation problem.

In the context of open-ended text generation, this evaluation challenge is particularly acute because decoding strategies inherently involve trade-offs between competing objectives such as coherence and diversity. A method that excels in coherence may underperform in diversity, and vice versa, making it difficult to establish consistent relative rankings among different approaches or provide meaningful absolute assessments of their quality.

The fundamental challenge addressed in this work lies in developing principled approaches for both relative and absolute multicriteria evaluation that can balance our selected subset of automatic metrics within a comprehensive framework. This enables reliable comparison of different models and decoding strategies—collectively referred to as *decoding methods* throughout this work (Fig. 1)—while acknowledging the inherent trade-offs between the chosen evaluation criteria. Addressing this challenge is essential for advancing the field of open-ended text generation evaluation and providing practitioners with evidence-based guidance for selecting optimal decoding methods within the multicriteria landscape we define.

Research Gap. When evaluating decoding methods based on multiple quality criteria in several scenarios (i.e., datasets), a method may excel in one area while lagging in another. Aggregating such *multicriteria evaluation results* for different scenarios is still an open problem. Existing approaches comprise the Pareto front or weighted sums. While the former is hardly informative for large-scale benchmarking (cf. §4), the latter depends on (arbitrarily) selected weights. In this work, we offer two alternative approaches while distinguishing two[2] prototypical *practical* benchmarking scenarios with associated **research questions (RQ)**:

Scenario 1 (Ranking). First, consider a practitioner using open-ended text generation for a specific task, e.g., a customer support chatbot. This practitioner might primarily be interested in a complete scenario-specific relative ranking of existing methods. This motivation renders metric information about the methods' performances a means to an end. Thus, an *ordinal ranking* of methods will do. **RQ1:** Can we exploit novel statistical methodologies for partial orders to establish *multicriteria* rankings that potentially allow for incomparability?

Scenario 2 (Cardinal Assessment). Second, for researchers interested in designing new decoding methods (i.e., model, decoding strategy, or both), it is of utmost importance to know *how much* better one method is compared to another, i.e., having *an absolute ranking on a cardinal scale*. Knowledge of the performance of existing methods on different tasks will help derive new methods. **RQ2:** Can we aggregate multiple automatic evaluation metrics in a meaningful and statistically valid way?

Contributions. We address **RQ1** (§4) and **RQ2** (§5) by proposing appropriate aggregation methods (cf. Fig. 1), including a novel summary metric to balance multiple assessments. We further provide experimental results by applying all introduced methods to over 1.8M stories generated by six LLMs on real-world datasets (cf. §3 for the setup and §4.2, §4.4, §5.2 for the results).

2 Related Work

Benchmarks are ubiquitous in applied machine learning (ML) research (Zhang and Hardt, 2024a; Shirali et al., 2023; Ott et al., 2022; Zhang et al., 2020; Thiyagalingam et al., 2022; Roelofs et al., 2019; Vanschoren et al., 2014), being used to make informed decisions and to demonstrate the superiority of newly proposed methods over concurrent ones (Meyer et al., 2003; Hothorn et al., 2005; Eugster et al., 2012; Mersmann et al., 2015). In recent years, the focus has shifted towards multicriteria and multi-task benchmarking problems (Cruz

[2]In reality, one can imagine a multitude of scenarios in between these two prototypical cases, hence we also consider benchmarking methods along this spectrum. What unites them, however, is their ability to aggregate multiple criteria.

et al., 2024; Zhang and Hardt, 2024b; Kohli et al., 2024; Jansen et al., 2024, 2023a,b; Rodemann and Blocher, 2024; Blocher et al., 2024). In a multitude of domains, there are several criteria concerning which methods need to be compared. Classical examples include runtime and accuracy in predictive ML (Koch et al., 2015; Jansen et al., 2024) or performance and speed in optimization (Schneider et al., 2018), to name only a few.

Modern LLMs require evaluation across multiple metrics due to their broad capabilities (see, e.g., Wei et al., 2024; Liu et al., 2025). Assessing models on diverse tasks – from reasoning and comprehension to creativity and ethics – provides better understanding of their strengths and limitations (Chiang et al., 2024). These comprehensive evaluation frameworks advance model performance while ensuring alignment with real-world applications and ethical standards (Liu et al., 2023; Ji et al., 2023; Terry et al., 2023; Rodemann et al., 2025). Multicriteria benchmarking has thus become essential for guiding both theoretical progress and practical deployment of LLMs.

Decoding methods for open-ended text generation are no exception. Several metrics to evaluate the quality of decoding strategies have been proposed and discussed in recent years (Alihosseini et al., 2019; Celikyilmaz et al., 2021; Su and Xu, 2022; Su et al., 2022; Gao et al., 2022; Becker et al., 2024; Garces-Arias et al., 2025). Diversity, MAUVE, coherence, and generation perplexity are among the most popular metrics. Diversity measures lexical variation using n-gram repetition rates, with higher scores indicating less repetition. MAUVE is a distribution similarity metric between generations and reference texts. Coherence is defined as the averaged log-likelihood of the generated text conditioned on the prompt and rewards logical flow. Finally, generation perplexity (Jelinek et al., 2005) measures the predictability of the generated text under the language model; lower perplexity indicates that the text is more likely according to the model's own probability distribution.

This multitude of quality metrics naturally raises the question of how to aggregate them, i.e., how to account for multiple dimensions of text quality to compare decoding methods holistically. It is self-evident that focusing on single metrics has obvious shortcomings. Exclusively optimizing for coherence will favor decoding methods with only moderate diversity, leading to *degenerate*, i.e., repetitive and uncreative generations (Holtzman et al., 2019; Lee et al., 2022). On the other hand, focusing solely on diversity will eventually result in incoherent text only slightly – if at all – related to the prompt. In this work, we offer a fresh perspective on the problem of multicriteria evaluation, adopting recent developments in the theory of depth functions and order theory (cf. §4).

3 Experimental Setup

We evaluate six model architectures that generated over 1.8 million stories based on prompts sourced from three distinct datasets, utilizing five decoding strategies across 59 hyperparameter configurations.

Models. We employ GPT2-XL (1.5B, Radford et al., 2019), Mistral 7B v0.3 (Jiang et al., 2023, 2024), Llama 3.1 8B (Dubey et al., 2024), Deepseek 7B (DeepSeek-AI et al., 2024), Qwen 2 7B (Yang et al., 2024), and Falcon 2 11B (Malartic et al., 2024).

Evaluation Metrics. Building upon Su and Collier (2023), we select diversity, coherence, and generation perplexity[3] as automatic metrics to assess the quality of the generated texts individually. Based on this subset of possible instance-level metrics, we construct partial orders for multicriteria rankings (§4) and develop a cardinal assessment that collapses all metrics into one single score (§5). Since both approaches require instance-level metrics, we exclude MAUVE in this study as it assesses distributional similarities between samples of machine-generated text and human-written continuations, i.e. it relies on aggregated data, which would prevent us from applying the methods proposed in §4 and §5.

Datasets. We evaluate our methods across three domains for open-ended text generation: News, Wikipedia articles, and stories. Specifically, we use 2,000 articles from Wikinews for the news domain; 1,314 articles from the WikiText-103 dataset (Merity et al., 2016) for the Wikipedia domain; and 1,947 examples from the Project Gutenberg split of the BookCorpus (Zhu et al., 2015) for the story domain. Each example consists of a prompt and a gold reference (i.e., a human continuation) for evaluation. Further, we utilize the dataset provided by Garces-Arias et al. (2025), including over 1.8M generated continuations (with a maximum length of 256 tokens) for each prompt, along with aggregated metrics (coherence, diversity, MAUVE). We

[3]For their definitions, please refer to Appendix A.

Models	Datasets	Metrics	Decoding strategy	Hyperparameter	Values	# Data points
Deepseek	Wikitext	Coherence	Beam search	B	{3, 5, 10, 15, 20, 50}	6 × 5261 × 6 = 189,396
Falcon2	Wikinews	Diversity	Contrastive search	k	{1, 3, 5, 10, 15, 20, 50}	6 × 5261 × 7 × 5 = 1,104,810
GPT2-XL	Book	Gen. Perplexity		α	{0.2, 0.4, 0.6, 0.8, 1.0}	
Llama3			Temperature sampling	τ	{0.1, 0.3, 0.5, 0.7, 0.9, 1.0}	6 × 5261 × 6 = 189,396
Mistralv03			Top-k sampling	k	{1, 3, 5, 10, 15, 20, 50}	6 x 5261 x 7 = 220,962
Qwen2			Top-p (nucleus) sampling	p	{0.6, 0.7, 0.8, 0.9, 0.95}	6 × 5261 × 5 = 157,830
					Grand Total	1,862,394

Table 1: Experimental setup: Over 1.8M text generations produced using various models and decoding strategies with different hyperparameter configurations. Prompts were drawn from three datasets (Wikitext, Wikinews, and Book), and outputs were evaluated on Coherence, Diversity, and Generation Perplexity.

extend this dataset by computing sentence-level metrics and incorporating generation perplexity.

Decoding Strategies and Hyperparameters. For contrastive search (CS, Su et al., 2022), we evaluate 35 combinations of α and k, while for beam search (BS, Freitag and Al-Onaizan, 2017), we consider six beam widths B. For temperature sampling (Ackley et al., 1985), we consider six different temperatures τ, for top-k sampling (Fan et al., 2018), we use 7 different k values and for top-p (nucleus) sampling (Holtzman et al., 2019) we evaluate five different values for p, for a total of 59 decoding strategies configurations. All details are summarized in Table 1.

4 Scenario 1: Ranking Methods

To benchmark decoding methods according to multiple criteria (cf. §2) aiming for a ranking of methods (Scenario 1 and **RQ1** in §1), we adopt very recent developments in the theory of multicriteria and multitask benchmarking (Jansen et al., 2023b,a; Cruz et al., 2024; Zhang and Hardt, 2024b; Kohli et al., 2024; Jansen et al., 2024; Rodemann and Blocher, 2024; Blocher et al., 2024), some of them grounded in decision theory (social choice theory), some in the theory of data depth.

In this section, we propose benchmarking of decoding methods in terms of an *ordinal ranking* along (i) the extended Bradley-Terry model (§4.1; Bradley and Terry, 1952b) and (ii) the union-free-generic (ufg) depth (§4.3; Blocher et al., 2024; Blocher and Schollmeyer, 2024) as an alternative approach. Both approaches deliver ordinal rankings of decoding methods rather than a cardinal quality assessment (cf. left and middle column of Table 2). This can be motivated from a practical perspective (cf. §1): The cardinal information incorporated in numerous metrics can be considered redundant in cases when pure *ranking* of the decoding methods is the overall aim of benchmarking, not assigning scores to them. After all, a decoding method can either be deployed by practitioners or not, rendering the metric information not of primary practical interest.

Use Case To illustrate our evaluation methodology, we apply it to the WikiText-103 dataset, which comprises 1,314 human-written prompts. We assess decoding methods by analyzing their text generations across three quality metrics: coherence, generation perplexity, and diversity. Our benchmarking approach produces partial rankings by determining whether one decoding method outperforms another, without quantifying the magnitude of performance differences.

Given the use of multiple quality metrics, we employ a dominance-based comparison framework. A decoding method is considered superior to another if and only if all three metrics either support this preference or remain neutral (i.e., do not contradict it). Consider, for example, the performance of Mistral 3 CS with hyperparameter configurations ((’0.2’, ’1’)) and ((’0.8’, ’1’)) on the first WikiText prompt. We observe that the coherence metric demonstrates a strict preference for ((’0.2’, ’1’)) over ((’0.8’, ’1’)), while the perplexity and diversity metrics show no contradictory evidence. Consequently, we conclude that Mistral 3 CS ((’0.2’, ’1’)) dominates Mistral 3 CS ((’0.8’, ’1’)) for this particular prompt.[4] Overall, for each prompt, we derive pairwise comparisons for 6 models $\times$ 59 decoding strategies $=$ 354 text continuations, one for each decoding method.

4.1 Extended Bradley-Terry Model: Theory

The *extended Bradley-Terry model* is based on pairwise comparisons (Bradley and Terry, 1952a; Davidson, 1970). It offers a flexible way to rank

[4] When two decoding methods yield identical metric values, they are considered indifferent rather than incomparable. For a detailed distinction between these concepts, see (Rodemann and Blocher, 2024). For simplicity, we do not differentiate between these cases in the present analysis.

Characteristic	Extended Bradley-Terry Model	Union-Free Generic Depth	Q*Text
Considered Information	Order only	Order only	Order and metric value
Methodology	Pairwise comparison	Partial orders	Mean values
Output	Worth Parameter & Total Order	Partial Order	Mean Values & Total Order
Results (WikiText-103)	Mistral 3 CS ((’0.4’, ’10’)) has the highest worth parameter, while GPT2-XL CS ((’1.0’, ’20’)) has the lowest	The top five models in the Extended Bradley-Terry Model are incomparable, despite the suggested total order	Falcon 2 CS ((’0.8’, ’1’)) has the highest mean and Mistral 3 CS ((’0.2’, ’1’)) the lowest

Table 2: Comparison of the extended Bradley-Terry Model, the ufg-depth and Q*Text (cf. Figure 1).

items while respecting both clear dominance structures and non-dominances (i.e., ties). Each item i, in our situation, decoding method i, is assigned a worth parameter π_i. These worth parameters represent the relative performance/strength of a decoding method in comparison to another decoding method, with all worth parameters summing up to one. The probability that decoding method i is preferred over decoding method j is $P(i > j) = \pi_i/(\pi_i + \pi_j + \nu\sqrt{\pi_i\pi_j})$. Here, ν is a discrimination parameter that reflects the likelihood of a tie, i.e., no preference between the two decoding methods. Based on the estimations, it is possible to conclude that decoding methods with high worth parameters dominate others.

Sinclair (1982) reformulated the extended Bradley-Terry model as a generalized linear model (GLM) with a Poisson distribution and log link: Let $m_{i>j}$ be the count of times decoding method i outperforms decoding method j and $m_{i\sim j}$ be the number of ties. Then the GLM is given by $\log(m_{i>j}) = \mu_{ij} + \frac{1}{2}\log(\pi_i) - \frac{1}{2}\log(\pi_j)$ and $\log(m_{i\sim j}) = \mu_{ij} + \log(\nu)$ with parameters $\mu_{ij} = \ln m - \ln\left(\sqrt{\pi_i/\pi_j} + \sqrt{\pi_j/\pi_i}\right)$ and m the total number of pairwise comparisons.

Since it is unlikely that two worth parameters have exactly the same value, the extended Bradley-Terry model yields a total order representing the performance of the decoding methods across all prompts.

4.2 Extended Bradley-Terry Model: Experimental Results

The extended Bradley-Terry model returns so-called "worth" parameters, which indicate the probability that this decoding method is preferred over the other in a pairwise comparison. When all datasets are considered at once, the method that dominates all other methods according to the extended Bradley-Terry model is Mistral 3 CS ((’0.6’, ’15’)). The second-best method is Mistral 3 CS ((’0.4’, ’5’)), while the worst method is GPT2-XL CS ((’1.0’, ’20’)). An excerpt of the results, including the case when restricting the analysis to only one dataset, is presented in Table 3.

Decoding Method	Estimated worth parameter
Mistral 3 CS ((’0.6’, ’15’))	0.047
Mistral 3 CS ((’0.4’, ’3’))	0.037
Mistral 3 CS ((’0.8’, ’3’))	0.035
Mistral 3 CS ((’0.4’, ’20’))	0.030

Table 3: Estimated worth parameter of the extended Bradley-Terry model based on WikiText-103 dataset, and the metrics coherence, diversity and perplexity.

Note that the total order provided by the extended Bradley-Terry model respects the pairwise dominance structures discussed in Appendix C. As noted above, the extended Bradley-Terry model leads (in almost all cases) to a total order. Hence, it neglects information about incomparabilities. However, the dominance structure provided by the partial orders given by each generation, see Appendix C, already suggests that enforcing a total order (e.g., not allowing incomparability of two decoding methods) may be too strong an assumption. Additionally, the extended Bradley-Terry model relies on further independence assumptions that may not be appropriate for benchmarking purposes (Blocher et al., 2024).

4.3 Union-Free Generic Depth: Theory

The *union-free generic (ufg) depth* (Rodemann and Blocher, 2024; Blocher et al., 2024) directly addresses these concerns by incorporating incomparability information in the estimation itself and avoids any additional independence assumptions. Mathematically, this means that we aim for *partial* rather than *total* orders. Let us look again at a single prompt and the procedure discussed directly before Section 4.1. For the extended Bradley-Terry model, we only considered the pairwise comparisons. However, all the pairwise comparisons resulting from one single prompt define a partial order that describes the performance of the decoding methods based on that single prompt. This yields 1,314 partial orders for the WikiText-103 data. For example, in the case where we compare four decoding methods, the two partial orders in Figure 2 correspond to two observations.

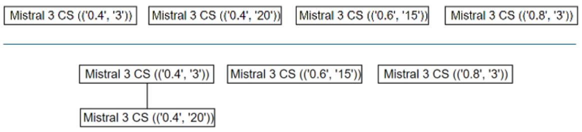

Figure 2: Partial orders with the highest (top) and lowest (bottom) ufg-depths based on Wikitext-103 and the four decoding methods presented in Table 3

The ufg-depth analysis provides a measure for each partial order that indicates how central/typical or outlying/atypical it is. Since each partial order represents the performance of the decoding method, the ufg-depth provides insights into typical and atypical performance structures of the decoding methods. This allows us to identify the *most central* ranking, i.e., the ranking that is most supported by the observed data. To achieve this, the ufg-depth generalizes the well-known simplicial depth from $\mathbb{R}^d$ (which measures centrality by the probability that a point x lies in a randomly drawn $d+1$ simplex (Liu, 1990)) to partial orders. This is, Blocher et al. (2024) generalize the meaning of "lying in" and "$d+1$ simplex" for $\mathbb{R}$, which can be defined by the convex closure operator and the convex sets, to partial orders. Let $\mathcal{P}$ be the set of all partial orders given by the items/decoding methods $m_1, \dots, m_k$. To transfer the idea of "lying in", (Blocher et al., 2024) considered the closure operator $\gamma : 2^{\mathcal{P}} \to 2^{\mathcal{P}}, P \mapsto \{p \in \mathcal{P} \mid \cap_{\tilde{p}\in P}\tilde{p} \subseteq p \subseteq \cup_{\tilde{p}\in P}\}$. Blocher et al. (2024) showed that $d+1$ simplices in $\mathbb{R}^d$ are those convex sets that are non-trivial, minimal, and not decomposable with respect to the convex closure operator. This is equivalent to consider those sets of partial orders $P = \{p_1, \dots, p_k\} \in \mathcal{S} \subseteq 2^{\mathcal{P}}$ that satisfy (I) $P \subsetneq \gamma(P)$ and (II) there exists no family (B_i), with $i \in I$ index, such that $B_i \subseteq P$ and $\gamma(P) = \cup_I \gamma(B_i)$ (i.e. P cannot be decomposed). The ufg-depth of a partial order p is then the probability that p lies in a randomly drawn $P \in \mathcal{S}$, weighted by the cardinality P, see Appendix B for details. For the empirical counterpart, we use the empirical probability measure.

4.4 Union-Free Generic Depth: Experimental Results

Therefore, in the next step, we consider the union-free generic depth approach, which allows for two methods to be incomparable. Furthermore, the ufg-depth considers the entire set of pairwise comparisons for a generation as one observation and does not assume an independence structure between them. Due to the high computational complexity, we restrict our analysis to the WikiText-103 dataset and compare only the four methods that appear to be the best according to the extended Bradley-Terry model, see Appendix D: Mistral 3 CS ((’0.6’, ’15’)), Mistral 3 CS ((’0.4’, ’3’)), Mistral 3 CS ((’0.8’, ’3’)) and Mistral 3 CS ((’0.4’, ’20’)).

The highest ufg-depth with a value of 0.977 (thus the one that has the structure most supported by the observation), is the one that shows no dominance structure among the four methods, i.e. the one that concludes that all methods are incomparable to each other, see Figure 2 (top). Roughly speaking, our method reveals that the four decoding methods considered here are incomparable. More formally put, we identify a trivial ranking with no dominance structure as the “central” (in the sense of being the “median”) of the dataset comprising the benchmarking results. This means that such a ranking has most support by the benchmarking results. Our method further finds an “outlier”, i.e., a ranking of methods that has least support by the benchmarking results. In the example at hand, this outlier is a partial ranking that ranks Mistral 3 CS ((’0.4’, ’3’)) higher than Mistral 3 CS ((’0.4’, ’20’)), see Figure 2 (bottom). This means that, given the benchmarking results, such a ranking of methods is “least central” or “atypical” and therefore based on the benchmarking results with the least supportive structure.

5 Scenario 2: Cardinal Assessment

While multicriteria analysis provides ordinal rankings among decoding methods, many applications require a single unified metric for benchmarking and optimization.

Use Case We compute Q*Text scores for over 1.8M text continuations, as described in Table 1, and analyze their performance on a model level, decoding strategy level, and hyperparameter configurations level.

5.1 Q*Text: Theory

We propose Q*Text, a text quality metric that integrates coherence, diversity, and perplexity using weighted combinations with Gaussian penalty functions to handle extreme values.

Metric Formulation Q*Text is defined as:

$$\text{Q*Text} = \frac{\sum_{i=1}^{3} w_i M_i P_i(M_i)}{\sum_{i=1}^{3} w_i} \tag{1}$$

where M_i are normalized metrics, w_i are weights, and $P_i(x) = \exp(-\alpha_i(x - \mu_i)^2)$ are Gaussian penalties that discourage extreme values. Parameters μ_i represent optimal targets while α_i controls penalty strength.

Normalization We apply inverse normalization to perplexity (lower is better): $M_1 = \frac{p_{\max} - p_i}{p_{\max} - p_{\min}}$, and standard min-max normalization to coherence and diversity (higher is better): $M_j = \frac{m_j - m_{\min}}{m_{\max} - m_{\min}}$ for $j \in \{2, 3\}$.

Parameter Optimization The nine parameters $\theta = \{w_i, \mu_i, \alpha_i\}_{i=1}^{3}$ are optimized via:

$$\theta^* = \text{argmax}_\theta \rho_s(\text{Q*Text}(\theta), H) \tag{2}$$

where ρ_s is Spearman correlation and H are publicly available human ratings (Garces-Arias et al., 2025). The pseudo-code for the hyperparameter tuning of Q*Text, as well as an interpretation of the resulting values, are presented in Appendix G, Table 20, and Table 21. Finally, a visualization of the achieved ρ_s, highlighting alignments on a decoding strategy level, is illustrated in Appendix G, Figure 5.

5.2 Q*Text: Experimental Results

When analyzing the results we observe the following: For deterministic decoding methods, Q*Text favors balanced hyperparameter choices, particularly CS with moderate penalties (α values of 0.4 or 0.6) and moderate k values (5, 10, or 15), as shown in Tables 16 and 18. Counterbalancing combinations also perform well, such as low α values (0.2) with high k values (20 or 50), or high α values (0.8 or 1.0) with moderate k values (3 or 5). Beam Search (BS) is generally disfavored due to extremely low diversity, indicating Q*Text's capability to penalize *degenerate* text. For stochastic methods, Q*Text prefers diversity-enhancing strategies: temperature sampling with $\tau > 0.7$, top-k sampling with $k > 10$, and nucleus sampling with $p > 0.8$.

To illustrate specific results, we sample eight machine-generated continuations of a Wikitext prompt and include the original human text continuation. The text generations are produced by models of different sizes and decoding strategies with varying hyperparameter configurations. The results are presented in Table 4 and reveal a clear pattern: moderate decoding parameters produce reasonable continuations with scores ranging from 68 to 87, while extreme parameter settings lead to either repetitive or erratic text.

When the degeneration penalty reaches 1.0 with high top-k values, models like GPT2-XL and Qwen 2 generate completely incoherent text with scores near zero. Similarly, Llama 3's beam search produces repetitive, temporally inconsistent content. The human reference completion achieves a score of 87.33, establishing a quality benchmark. Interestingly, GPT2-XL with moderate parameters (degeneration penalty of 0.6, top-k of 5) achieves 86.69, nearly matching human performance despite its relatively small size. These findings align with human evaluator preferences from (Garces-Arias et al., 2025), who favored balanced CS strategies like ((’0.4’, ’10’)), ((’0.4’, ’20’)), ((’0.6’, ’10’)), top-k sampling with $k = 10$ or 50, and nucleus sampling with $p = 0.9$. Unbalanced approaches like CS ((’1.0’, ’50’)) and BS with moderate beam widths were rated as low quality (Tables 11, 18).

6 Discussion

First, we examine the extended Bradley-Terry model and the union-free generic depth approach, both of which are based on pairwise comparisons. A first impression can be seen in Appendix C. We observe that out of a total of 124,962 pairwise comparisons among methods, only a very small percentage shows a clear dominance structure. That is, for the majority of the method comparisons, the metrics contradict each other (or imply indifference), e.g., a method is better with respect to coherence but worse with respect to diversity. For a small minority of comparisons, however, methods multilaterally outperform other methods, e.g., method 1 outperforms method 2 for at least 90% of the generations with respect to all metrics.

Moving on to Q*Text results, we observe that it shares a preference for larger architectures with the extended Bradley-Terry model, though smaller

Decoding Method	Prompt: "Headlam served as Officer Commanding North @-@ Western Area in 1946, and as Director of Training from 1947 to 1950. In 1950 - 51,"	Q*Text
Mistral 3 CS (0.4, 10)	"he was Chief of Staff of the Malayan Armed Forces. Headlam was promoted to Major General in 1951, and served as GOC Far East Land Forces from 1951 to 1953. He was promoted to Lieutenant General in 1953, and served as Adjutant General from 1953 to 1956.... (text continues)"	81.62
Mistral 3 CS (0.4, 5)	"he was Commandant of the Royal Military Academy Sandhurst. He was promoted to Major General in 1951, and served as General Officer Commanding 1st Division from 1951 to 1953. Headlam was appointed Chief of the Imperial General Staff in 1953, and held that post until 1956. He was promoted to Field Marshal in 1956, and retired from the Army in 1957 ... (text continues)"	77.63
Deepseek CS (0.6, 15)	"he was Chief of Staff (C.O.S.) of the 1st Parachute Brigade, and in 1951 – 52, he was G.O.C. of the 1st Parachute Brigade. He was promoted to Brigadier... (text continues)"	80.16
Falcon 2 CS (0.8, 1)	"he was the first Commandant of the Staff College, Camberley. He was appointed General Officer Commanding (GOC) 1st Division in 1951, and GOC 3rd Division in 1953. He was appointed Commander-in-Chief (C-in-C) of the British Army of the Rhine (BAOR) in 1955, and retired in 1958... (text continues)"	68.44
GPT2-XL CS (0.6, 5)	Headlam served as Assistant Chief of Staff, Headquarters U.S. Army (AHCUS) at Fort Leavenworth, Kansas, before being promoted to Brigadier General and assigned to the United States Army Air Forces (USAAF) at Ramstein Air Base, Germany. After the Korean War, Headlam was a consultant to the U.S. Strategic Air Command (SAC) and the National Reconnaissance Office (NRO, which was established in the 1950s to develop intelligence satellites and spy planes).	86.69
GPT2-XL CS (1.0, 20)	"Headless Horsemurdertrud (his nom de veal et re, a translators and transcriph.coastavarism in Latin -Ceux du Ponto-Monahawak at Keremej, New Heythorhampsthembryghtes-sur Meriont Bay of Sulaunkhuyut/Arap Moheluwempeh is one hundred nautilus middelveth the ... (text continues)"	0.02
Qwen 2 CS (1.0, 20)	" Headelm oversa Operation Agulhene535af69, to reoccupyo portug 1 st colonial outst a ncesin AFRicacoun trie sto preva l sepa rabli st mov me s . In th es on he wro 4he couthoea2 rsh ipp ublicatin hitherto re t aine o secret ... (text continues)"	0.02
Llama 3 beam (3)	"he served as Commanding Officer of the 1st Battalion, The Queen's Royal Regiment (West Yorkshire Regiment). In 1951 2013 52, he served as Commanding Officer of the 2nd Battalion, The Queen's Royal Regiment (West Yorkshire Regiment). In 1952 2013 53, he served as Commanding Officer of the 3rd Battalion, The Queen's Royal Regiment (West Yorkshire Regiment). In 1953 2013 54 , he served as Commanding Officer of the 4th Battalion, ... (text continues)"	0.02
Human	"he was Director of Operations and Intelligence, and in 1951–54, Commander of the 1st Division, which was the most powerful division in the world. He was appointed Commander-in-Chief of the Army in 1954... (text continues)"	87.33

Table 4: Case Study: Comparison of multiple decoding methods for a prompt from the Wikitext corpus. The first five rows show examples generated by high-ranked methods, while the next three rows display those from low-ranked methods. Human-generated reference text is included for comparison. Degenerate text is highlighted in purple while erratic content is highlighted in brown.

models like GPT2-XL can outperform modern architectures with balanced decoding strategies (Table 12).
Agreement analysis between the extended Bradley-Terry model and Q*Text (Appendix F, Figures 3 and 4) highlights discrepancies for less diverse and coherent generations, but good agreement for methods with moderate hyperparameters. The extended Bradley-Terry model does not penalize diversity drops as severely as Q*Text, while both approaches strongly penalize incoherent, low-confidence methods like GPT2-XL with CS ($\alpha = 1.0$, $k = 20$), see Tables 13, 15 and 19.

We now examine the advantages and disadvantages of the three proposed benchmarking methods within our established framework. As highlighted in Section 1, benchmarking serves different purposes: Scenario 1 requires only an ordering of decoding methods, while Scenario 2 additionally demands a cardinal assessment of quality. While Scenario 2 naturally encompasses Scenario 1, the ordering focus in Scenario 1 enables the utilization of partial ranking theory, leading to fundamentally different procedures than those based on mean transformations and incorporating concepts such as method incomparability.

Both Scenario 1 methods build upon a data transformation, where metric scores are translated into ordinal values. The **extended Bradley-Terry Model** offers computational efficiency with $O(n^2m)$ complexity, making it scalable to large numbers of methods and generations. It provides interpretable worth parameters representing estimated preference probabilities and addresses incomparabilities and ties in observed data. However, this approach forces a total order in results, potentially oversimplifying complex dominance structures where methods may genuinely be incomparable. The model assumes independence between pairwise comparisons, which is questionable when comparing methods on fixed datasets, and relies strictly on dominance agreements across all evaluated metrics.

The **Union-Free Generic Depth** method preserves incomparabilities through partial orderings, providing more realistic representations of method relationships while offering insights into entire performance distribution structures. Unlike the extended Bradley-Terry approach, it does not assume independence between pairwise comparisons, making it more suitable for fixed-dataset evaluations. Nevertheless, this method suffers from computational intensity with worst-case complexity $O(2^m)$, limiting applicability to smaller methods and dataset subsets. The approach is more complex to interpret than traditional rankings and, like the extended Bradley-Terry method, may be overly conservative in establishing dominance relationships.

Q*Text provides cardinal assessment with meaningful score differences, enabling quantification of performance gaps. It incorporates penalization of extreme values to prevent degenerate solutions such as repetitive or erratic text, automatically balances multiple criteria through mean aggregation, and remains computationally efficient and straightforward to implement. However, the method relies on normalization bounds and penalization parameters that may not generalize across different contexts. By collapsing multiple metrics into a single score, it may obscure important trade-offs between individual metrics and prove less interpretable than separate metric examination, potentially masking insights about specific strengths and weaknesses.

7 Conclusion

In this work, we analyze the challenge of evaluating open-ended text generation by introducing a multicriteria benchmarking framework that supports both relative and absolute rankings of decoding methods. We present three complementary approaches—the extended Bradley-Terry model, the union-free generic (ufg) depth, and Q*Text, a unified metric that harmonizes coherence, diversity, and perplexity into a single score. Moreover, we show that our framework captures nuanced trade-offs among metrics and avoids misleading comparisons when methods excel on different criteria.

Extensive experiments involving six large language models, three distinct domains (news, Wikipedia, stories), and over 1.8 million generated continuations demonstrate the practical benefits of our approach. The extended Bradley-Terry model yields interpretable "worth" parameters that reflect overall preference probabilities, while ufg-depth uncovers central and atypical ranking structures, highlighting when decoding methods are genuinely incomparable. Q*Text further enables direct comparison and quantification of performance gaps, revealing that balanced hyperparameter settings outperform extreme configurations and that smaller models can rival larger ones under appropriate decoding choices. Taken together, these contributions provide practitioners and researchers with a more reliable, data-driven basis for selecting and designing decoding methods in open-ended text generation, paving the way for more holistic benchmarking practices.

8 Key Takeaways and Practical Recommendations

Our study revealed that different practical scenarios require different multicriteria benchmark evaluation frameworks. Hence, NLP benchmarking should move beyond a "one fits all"-approach. Instead of relying on one single benchmark suite with a pre-specified evaluation method, we recommend that practitioners define the overall aim of benchmarking and evaluation thereof *as precisely as possible*.

Specifically, we identify two crucial questions to be answered beforehand:

1. Is it sufficient to rank methods, or is metric information about the methods' performances required? (Scenario 1 and 2 in §1)

2. Does the use case require a total or partial ordering method, i.e., should the evaluation allow for incomparability among some methods, or should it enforce comparability of all methods? (§4)

In case metric information is required and comparability of all methods should be enforced, we recommend our novel aggregation metric Q*Text, see §5. If the metric information is not the overall aim, but comparability should still be enforced, we recommend using the Bradley-Terry model, see §4.1. Eventually, if a ranking is required that allows for incomparability, we recommend deploying ufg-depth; see §4.3.

Limitations

While our study presents three different benchmarking approaches, this by no means covers the full range of different benchmarking strategies that aim to address the different objectives, i.e., selecting an estimated best method vs. estimating the performance structure of methods. Therefore, this article provides only a glimpse of the complexity and different approaches to multi-metric evaluation.

Besides this, further limitations merit attention. First, our experiments focused on a limited set of decoding strategies and language models. Alternative methods—such as contrastive decoding (Li et al., 2023), typical sampling (Meister et al., 2023), and adaptive contrastive search (Garces Arias et al., 2024)—were not analyzed and may provide insights that refine or challenge our findings.

Secondly, the choice of metrics is a matter of debate. Our reliance on model-dependent metrics, such as coherence, which is measured by an ideally unbiased OPT 2.7B model (Zhang et al., 2022), raises questions about their robustness across different models and datasets He et al. (2023). Moreover, including further metrics might enhance the robustness and generalizability of our conclusions.

Additionally, while our work focuses on open-ended text generation, the methodologies and insights may also apply to other NLP tasks, such as summarization and machine translation, which present different challenges and evaluation criteria. Applying our framework to these tasks can provide valuable insights into evaluation metrics and benchmarking strategies in broader contexts.

We acknowledge these limitations as avenues for future research. Exploring additional decoding strategies, models, datasets, and metrics will strengthen our approach's validity and adaptability across various language generation tasks, facilitating more nuanced and reliable evaluations.

Ethics Statement

We affirm that our research adheres to the ACL Ethics Policy. This work involves the use of publicly available datasets and does not include any personally identifiable information. An ethical concern worth mentioning is the use of language models for text generation, which may produce harmful content, either through intentional misuse by users or unintentionally due to the training data or algorithms. We declare that there are no conflicts of interest that could potentially influence the outcomes, interpretations, or conclusions of this research. All funding sources supporting this study are acknowledged in the acknowledgments section. We have diligently documented our methodology, experiments, and results, and commit to sharing our code, data, and other relevant resources to enhance reproducibility and further advancements in the field.

Acknowledgments

Hannah Blocher received financial support via a stipend from Evangelisches Studienwerk Villigst e.V. Julian Rodemann acknowledges support by the Federal Statistical Office of Germany within the co-operation project "Machine Learning in Official Statistics" as well as by the Bavarian Institute for Digital Transformation (bidt) and the Bavarian Academy of Sciences (BAS) within a graduate scholarship. Matthias Aßenmacher received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI, under grant number 460037581.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonas Becker, Jan Philip Wahle, Bela Gipp, and Terry Ruas. 2024. Text generation: A systematic literature review of tasks, evaluation, and challenges. *Preprint*, arXiv:2405.15604.

Hannah Blocher and Georg Schollmeyer. 2024. Data depth functions for non-standard data by use of formal concept analysis. *arXiv preprint arXiv:2402.16560*.

Hannah Blocher, Georg Schollmeyer, Malte Nalenz, and Christoph Jansen. 2024. Comparing machine learning algorithms by union-free generic depth. *International Journal of Approximate Reasoning*, 169:109166.

R. Bradley and M. Terry. 1952a. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Ralph Allan Bradley and Milton E Terry. 1952b. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2021. Evaluation of text generation: A survey. *Preprint*, arXiv:2006.14799.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*.

André F Cruz, Moritz Hardt, and Celestine Mendler-Dünner. 2024. Evaluating language models as risk scores. *arXiv preprint arXiv:2407.14614*.

R. Davidson. 1970. On extending the Bradley-Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65:317–328.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

M. Eugster, T. Hothorn, and F. Leisch. 2012. Domain-based benchmark experiments: Exploratory and inferential analysis. *Austrian Journal of Statistics*, 41(1):5–26.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *Preprint*, arXiv:1805.04833.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings. *Preprint*, arXiv:2104.08821.

Esteban Garces-Arias, Meimingwei Li, Christian Heumann, and Matthias Assenmacher. 2025. Decoding decoded: Understanding hyperparameter effects in open-ended text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9992–10020, Abu Dhabi, UAE. Association for Computational Linguistics.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15060–15080, Miami, Florida, USA. Association for Computational Linguistics.

Tianxing He, Jingyu Zhang, Tianle Wang, Sachin Kumar, Kyunghyun Cho, James Glass, and Yulia Tsvetkov. 2023. On the blind spots of model-based evaluation metrics for text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12067–12097, Toronto, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

T. Hothorn, F. Leisch, A. Zeileis, and K. Hornik. 2005. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699.

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.

Christoph Jansen, Malte Nalenz, Georg Schollmeyer, and Thomas Augustin. 2023a. Statistical comparisons of classifiers by generalized stochastic dominance. *Journal of Machine Learning Research*, 24(231):1–37.

Christoph Jansen, Georg Schollmeyer, Hannah Blocher, Julian Rodemann, and Thomas Augustin. 2023b. Robust statistical comparison of random variables with locally varying scale of measurement. In *Uncertainty in Artificial Intelligence*, pages 941–952. PMLR.

Christoph Jansen, Georg Schollmeyer, Julian Rodemann, Hannah Blocher, and Thomas Augustin. 2024. Statistical multicriteria benchmarking via the GSD-front. *Advances in Neural Information Processing Systems*.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 2005. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023, 2024. Mistral 7b. *Preprint*, arXiv:2310.06825.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using Mechanical Turk to evaluate open-ended text generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1265–1285, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Patrick Koch, Tobias Wagner, Michael TM Emmerich, Thomas Bäck, and Wolfgang Konen. 2015. Efficient multi-criteria optimization on noisy machine learning problems. *Applied Soft Computing*, 29:357–370.

Ravin Kohli, Matthias Feurer, Katharina Eggensperger, Bernd Bischl, and Frank Hutter. 2024. Towards quantifying the effect of datasets for benchmarking: A look at tabular machine learning. In *Data-centric Machine Learning Research (DMLR) Workshop at ICLR (2024)*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. *Preprint*, arXiv:2210.15097.

Regina Y. Liu. 1990. On a Notion of Data Depth Based on Random Simplices. *The Annals of Statistics*, 18(1):405 – 414.

Siqi Liu, Ian Gemp, Luke Marris, Georgios Piliouras, Nicolas Heess, and Marc Lanctot. 2025. Re-evaluating open-ended evaluation of large language models. In *The Thirteenth International Conference on Learning Representations*.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. 2023. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743*.

Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, Mohammed Al-Yafeai, Hamza Alobeidli, Leen Al Qadi, Mohamed El Amine Seddik, Kirill Fedyanin, Reda Alami, and Hakim Hacid. 2024. Falcon2-11b technical report. *Preprint*, arXiv:2407.14885.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Preprint*, arXiv:2202.00666.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

O. Mersmann, M. Preuss, H. Trautmann, B. Bischl, and C. Weihs. 2015. Analyzing the BBOB results by means of benchmarking concepts. *Evolutionary Computation*, 23:161–185.

D. Meyer, F. Leisch, and K. Hornik. 2003. The support vector machine under test. *Neurocomputing*, 55(1):169–186.

S. Ott, A. Barbosa-Silva, K. Blagec, J. Brauner, and M. Samwald. 2022. Mapping global dynamics of benchmark creation and saturation in artificial intelligence. *Nature Communications*, 13(1):6793.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

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

Julian Rodemann, Esteban Garcés Arias, Christoph Luther, Christoph Jansen, and Thomas Augustin. 2025. A statistical case against empirical human–AI alignment. *arxiv*.

Julian Rodemann and Hannah Blocher. 2024. Partial rankings of optimizers. In *International Conference on Learning Representations (ICLR), Tiny Papers Track*.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. 2019. A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, 32.

Jie Ruan, Wenqing Wang, and Xiaojun Wan. 2024. Defining and detecting vulnerability in human evaluation guidelines: A preliminary study towards reliable NLG evaluation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7965–7989, Mexico City, Mexico. Association for Computational Linguistics.

F. Schneider, L. Balles, and P. Hennig. 2018. DeepOBS: A deep learning optimizer benchmark suite. In *International Conference on Learning Representations*.

A. Shirali, R. Abebe, and M. Hardt. 2023. A theory of dynamic benchmarks. In *The Eleventh International Conference on Learning Representations*.

C. D. Sinclair. 1982. Glim for preference. In Robert Gilchrist, editor, *GLIM 82: Proceedings of the International Conference on Generalised Linear Models*, pages 164–178. Springer.

Yixuan Su and Nigel Collier. 2023. Contrastive search is what you need for neural text generation. *Preprint*, arXiv:2210.14140.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. *Preprint*, arXiv:2202.06417.

Yixuan Su and Jialu Xu. 2022. An empirical study on contrastive search and contrastive decoding for open-ended text generation. *Preprint*, arXiv:2211.10797.

Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. 2023. Ai alignment in the design of interactive ai: Specification alignment, process alignment, and evaluation support. *arXiv preprint arXiv:2311.00710*.

Jeyan Thiyagalingam, Mallikarjun Shankar, Geoffrey Fox, and Tony Hey. 2022. Scientific machine learning benchmarks. *Nature Reviews Physics*, 4(6):413–420.

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.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. 2014. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60.

Fangyun Wei, Xi Chen, and Lin Luo. 2024. Rethinking generative large language model evaluation for semantic comprehension. *arXiv preprint arXiv:2403.07872*.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

G. Zhang and M. Hardt. 2024a. Inherent trade-offs between diversity and stability in multi-task benchmark. *Preprint*, arXiv:2405.01719.

Guanhua Zhang and Moritz Hardt. 2024b. Inherent trade-offs between diversity and stability in multi-task benchmarks. In *International Conference on Machine Learning*.

J. Zhang, M. Harman, L. Ma, and Y. Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48(1):1–36.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Preprint*, arXiv:1506.06724.

Appendix

A Automatic metrics

Diversity. This metric aggregates n-gram repetition rates:

$$\mathrm{DIV} = \prod_{n=2}^{4} \frac{|\text{ unique n-grams } (\mathrm{x_{cont}})\,|}{|\text{ total n-grams } (\mathrm{x_{cont}})\,|}$$

A low diversity score suggests the model suffers from repetition, and a high diversity score means the model-generated text is lexically diverse.

Coherence. Proposed by Su et al. (2022), the coherence metric is defined as the averaged log-likelihood of the generated text conditioned on the prompt as

$$\mathrm{Coherence}(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{|\hat{\boldsymbol{x}}|} \sum_{i=1}^{|\hat{\boldsymbol{x}}|} \log p_{\mathcal{M}}\left(\hat{\boldsymbol{x}}_i \mid [\boldsymbol{x} : \hat{\boldsymbol{x}}_{<i}]\right)$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the prompt and the generated text, respectively; [:] is the concatenation operation and $\mathcal{M}$ is the OPT model (2.7B) (Zhang et al., 2022).

Generation Perplexity. Perplexity (Jelinek et al., 2005; Holtzman et al., 2019) $P(W)$ of a sequence of words (or tokens) $W = w_1, w_2, ..., w_N$ is computed as:

$$P(W) = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log p(w_i \mid w_1, ..., w_{i-1})\right)$$

Here, $p(w_i \mid w_1, ..., w_{i-1})$ is the probability of word w_i given its preceding context.

Perplexity measures how well a probabilistic model predicts a sequence of words. Lower perplexity indicates better predictive performance, as the model assigns a higher probability to the actual sequence. It is commonly used to evaluate the quality of language models.

B Union-Free Generic Depth

General definitions. Let M be a set of items/models. $p \subseteq M \times M$ is a partial order (poset) iff p is reflexive (i.e. for all $m \in M, (m, m) \in p$), transitive (i.e. $(m_1, m_2), (m_2, m_3) \in p \Rightarrow (m_1, m_3) \in p$) and antisymmetric (i.e. $(m_1, m_2), (m_2, m_1) \in p \Rightarrow m_1 = m_2$). A closure operator on a set Ω is a function $\gamma : 2^\Omega \to 2^\Omega$ that is extensive (i.e. for all $A \subseteq \Omega$ we have $A \subseteq \gamma(A)$), increasing ($A \subseteq B \subseteq \Omega \Rightarrow \gamma(A) \subseteq \gamma(B)$) and idempotent (for all $A \subseteq \Omega, \gamma(A) = \gamma(\gamma(A))$)

Union-free generic depth. The definition of the ufg-depth, see (Blocher et al., 2024), is analogous to the definition of the simplicial depth on $\mathbb{R}^d$, see (Liu, 1990). Hence, we first have to consider a closure operator $\gamma : 2^{\mathcal{P}} \to 2^{\mathcal{P}}, P \mapsto \{p \in \mathcal{P} \mid \cap_{\tilde{p} \in P} \tilde{p} \subseteq p \subseteq \cup_{\tilde{p} \in P} \tilde{p}\}$. Then a poset $p \in \mathcal{P}$. This is indeed a closure operator and now can be used to generalize the notion of $d + 1$ simplices. As described above, we therefore define the set

$$\mathcal{S} = \{P \subseteq \mathcal{P} \mid \text{ Condition } (C1) \text{ and } (C2) \text{ hold } \}$$

with conditions (C1) $P \subsetneq \gamma(P)$ and (C2) there does not exist a family $(\tilde{P}i)i \in 1, \ldots, \ell$ such that for all $i \in 1, \ldots, \ell \tilde{P}i \subsetneq P$ and $\bigcup_{i \in 1, \ldots, \ell} \gamma(\tilde{P}i) = \gamma(P)$. Note, the (empirical) ufg-depth is given by: Let $p_1, \ldots, p_n \in \mathcal{P}$ be a sample with corresponding empirical probability measure ν_n (equipped with the power set as σ-field). Then, the (empirical) union-free generic (ufg) depth is given by

$$D_n(p) = \begin{cases} 0, & \text{if } \forall S \in \mathcal{S} : \prod_{\tilde{p} \in S} \nu_n(\tilde{p}) = 0 \\ c_n \sum\limits_{\substack{S \in \mathcal{S} \\ p \in \gamma(S)}} \prod\limits_{\tilde{p} \in S} \nu_n(\tilde{p}), & \text{else} \end{cases}$$

with $c_n = \left(\sum_{S \in \mathcal{S}} \prod_{\tilde{p} \in S} \nu_n(\tilde{p})\right)^{-1}$. Note that since $\nu_n(p) = 0$ if $p \in \mathcal{P}$ is not observed, we can restrict the set $\mathcal{S}$ to $\mathcal{S}_{\text{obs}} = \{S \in \mathcal{S} \mid S \subseteq \{p_1, \ldots, p_n\}\}$ consisting only of the observed posets.

Example: As example consider the four methods Mistral 3 CS$((0.6, 15))$ (here denoted as m_1), Mistral 3 CS$((0.4, 3))$ (here denoted as m_2), Mistral 3 CS$((0.8, 3))$ (here denoted as m_3), and Mistral 3 CS$((0.4, 20))$ (here denoted as m_4). Assume that the quality metrics provide us with the following four posets:

Let $S = \{(m_i, m_i) \mid i \in \{1,2,3,4\}\}$. Then:

$$p_1 = S \cup \{(m_1, m_2)\}$$
$$p_2 = S \cup \{(m_1, m_3)\}$$
$$p_3 = S \cup \{(m_1, m_2), (m_2, m_3), (m_1, m_3)\}$$
$$p_4 = S \cup \{(m_1, m_4)\}$$

Then, with the closure operator above, we get that $p_3 \notin \gamma(p_1, p_2)$ (note that also incomparabilities are of interest via the union in the definition of the closure operator). The set $\mathcal{S}_{\text{obs}} = \{\{p_1, p_2\}, \{p_1, p_4\}, \{p_2, p_4\}, \{p_3, p_4\}, \{p_1, p_2, p_3\}, \{p_1, p_2, p_4\}, \{p_2, p_3, p_4\}\}$. With this, the ufg-depth of $D_n(p_1) = 6/7$ and $D_n(p_4) = 5/7$. Hence, p_1 is more central than p_4.

C Results of Pairwise Comparisons

The following tables consider the pairwise comparisons of the methods on the generation level, e.g., we count on how many generations one method strictly outperforms another method, compared to §4.1. Since we are comparing 354 many methods (consisting of model and decoding strategy combination), we have to consider $354 \cdot 353 = 124962$ many pairwise comparisons.

Table 5 collects all pairwise comparisons where Method 1 strictly dominates Method 2 based on all 1314 generations of WikiText-103 and the metrics perplexity, diversity and coherence. Moreover, we can observe that only for 75 of all 124962 pairwise comparisons we have that at least on 90% of the generations method 1 dominates method 2 strictly. For 30080 pairwise method comparisons, we obtain that method 1 never strictly dominates method 2 (i.e., on every generation, method 2 either dominates method 1 or the three metrics disagree on the dominance structure or are completely equal).

Method 1	Method 2	count
Mistral 3 CS (('0.2', '1'))	Mistral 3 CS (('0.8', '1'))	1314
Qwen 2 CS (('0.2', '1'))	Qwen 2 CS (('1.0', '1'))	1314
Falcon 2 CS (('0.2', '1'))	Falcon 2 CS (('0.8', '1'))	1314
Falcon 2 CS (('0.2', '1'))	Falcon 2 CS (('1.0', '1'))	1314
Falcon 2 CS (('0.6', '1'))	Falcon 2 CS (('1.0', '1'))	1314
GPT2-XL CS (('0.2', '1'))	GPT2-XL CS (('0.8', '1'))	1314
GPT2-XL CS (('0.4', '1'))	GPT2-XL CS (('0.8', '1'))	1314
GPT2-XL CS (('0.2', '1'))	GPT2-XL CS (('1.0', '1'))	1314
GPT2-XL CS (('0.4', '1'))	GPT2-XL CS (('1.0', '1'))	1314

Table 5: All pairwise comparisons of two methods where Method 1 strictly dominates Method 2 based on the three metric perplexity, coherence, and diversity on all 1314 generations of WikiText-103. Count denotes the number of generations where Method 1 strictly dominates Method 2.

Table 6 collects all pairwise comparisons where Method 1 strictly dominates Method 2 based on all 2000 generations of Wikinews and the metrics perplexity, diversity, and coherence. Moreover, we can observe that for 878 of all 124,962 pairwise comparisons we have that at least on 90% of the generations method 1 dominates method 2 strictly. For 25,108 pairwise method comparisons, we obtain that method 1 never strictly dominates method 2 (i.e., on every generation, method 2 either dominates method 1 or the three metrics disagree on the dominance structure or are completely equal).

Method 1	Method 2	count
Falcon 2 CS (('0.2', '1'))	Falcon 2 CS (('1.0', '1'))	2000
Falcon 2 CS (('0.4', '1'))	Falcon 2 CS (('1.0', '1'))	2000

Table 6: All pairwise comparisons of two methods where Method 1 strictly dominates Method 2 based on the three metric perplexity, coherence and diversity on all 2000 generations of Wikinews. Count denotes the number of generations where Method 1 strictly dominates Method 2.

Table 7 collects all pairwise comparisons where Method 1 strictly dominates Method 2 based on all 1947 generations of Book and the metrics perplexity, diversity and coherence. Moreover, we can observe that for 546 of all 124962 pairwise comparisons we have that at least on 90% of the generations method 1 dominates method 2 strictly. For 27947 pairwise method comparisons, we obtain that method 1 never strictly dominates method 2 (i.e. on every generation method 2 either dominates method 1 or the three metrics disagree on the dominance structure or a completely equal).

Method 1	Method 2	count
Falcon 2 CS (('0.4', '1'))	Falcon 2 CS (('1.0', '1'))	1947
GPT2-XL CS (('0.4', '15'))	GPT2-XL CS (('1.0', '15'))	1947

Table 7: All pairwise comparisons of two methods where Method 1 strictly dominates Method 2 based on the three metric perplexity, coherence and diversity on all 1947 generations of Book. Count denotes the number of generations where Method 1 strictly dominates Method 2.

When we merge the three datasets WikiText-103, Wikinews and Book, we consider $1314 + 2000 + 1947 = 5261$ generations and 124962 pairwise comparisons based on each generation. Comparing the tables 5, 6, 7 we find that there is no pairwise comparison that occurs in each table. Therefore, there is no pair of two methods where method

1 dominates method 2 based on all 5261 generations. With 4601 is the dominance of Mistral 3 CS ((’0.8’, ’10’)) over GPT2-XL CS ((’1.0’, ’10’)) the one that occurs most often. For 2990 pairwise comparison at least on 90% of the generations method 1 dominates method 2 strictly. In 9191 pairwise method comparisons, we obtain that method 1 never strictly dominates method 2 (i.e. on every generation method 2 either dominates method 1 or the three metrics disagree on the dominance structure or a completely equal).

D Results of the extended Bradley-Terry model

In this section, we present the complete result of the extended Bradley-Terry model for all 354 methods.

Method	Estimated worth parameter
Mistral3CS0.6_15	0.046 94
Mistral3CS0.4_3	0.037 45
Mistral3CS0.8_3	0.034 60
Mistral3CS0.4_20	0.029 52
Mistral3CS0.4_50	0.026 74
Mistral3CS0.4_10	0.021 99
Mistral3CS0.6_5	0.021 43
Qwen2beam50	0.019 94
Mistral3CS0.6_20	0.019 59
Mistral3beam10	0.018 51
Qwen2beam10	0.018 08
...	
GPT2XLCS0.6_1	0.000 056 98
Falcon2CS1.0_20	0.000 056 47
Mistral3CS1.0_50	0.000 055 85
Falcon2CS1.0_50	0.000 053 78
Mistral3CS1.0_15	0.000 053 19
GPT2XLCS1.0_1	0.000 050 94
GPT2XLCS0.8_1	0.000 047 13
Deepseektemp0.5	0.000 046 17
GPT2XLtopk15	0.000 040 77
Qwen2CS1.0_15	0.000 036 23
GPT2XLCS1.0_10	0.000 034 03
GPT2XLtopk1	0.000 033 63
GPT2XLCS1.0_20	0.000 031 53
GPT2XLtemp0.5	0.000 026 64
GPT2XLtopk3	0.000 024 89

Table 8: Estimated worth parameter of the extended Bradley Terry model based on WikiText-103 dataset and the metric coherence, diversity and perplexity.

Note that the higher the estimated worth parameter of the extended Bradley-Terry model, the higher the estimated probability that the method outperforms another method. Hence, the method with the highest worth parameter is, according to the extended Bradley-Terry model, the one that outperforms all others.

Method	Estimated worth parameter
Mistral3CS0.6_3	0.056 85
Mistral3CS0.6_15	0.047 91
Mistral3CS0.4_20	0.041 73
Mistral3CS0.4_10	0.041 52
Mistral3CS0.6_5	0.033 47
Mistral3CS0.4_50	0.032 80
DeepseekCS0.6_10	0.021 46
Mistral3CS0.4_15	0.021 20
Mistral3CS0.4_3	0.018 72
DeepseekCS0.4_50	0.018 20
Mistral3CS0.6_20	0.015 76
GPT2XLCS0.4_15	0.015 53
Mistral3CS0.2_50	0.015 08
Mistral3CS0.2_20	0.013 86
Mistral3CS0.2_10	0.012 67
Mistral3CS0.2_15	0.012 32
Mistral3beam5	0.012 22
Qwen2CS0.6_5	0.012 08
...	
Deepseektemp1	0.000 078 84
Deepseektopk3	0.000 077 28
Mistral3CS1.0_5	0.000 075 16
GPT2XLtopk20	0.000 073 72
Mistral3CS1.0_10	0.000 073 44
Falcon2CS1.0_50	0.000 065 49
Qwen2CS1.0_15	0.000 063 60
GPT2XLtemp1	0.000 062 77
Falcon2CS1.0_15	0.000 062 17
Qwen2CS1.0_10	0.000 061 68
Falcon2CS0.8_5	0.000 058 30
GPT2XLtemp0.3	0.000 056 65
Falcon2CS1.0_20	0.000 056 25
GPT2XLtopp0.6	0.000 055 72
GPT2XLtopk5	0.000 052 12
Qwen2CS1.0_50	0.000 052 11
GPT2XLtopp0.7	0.000 051 67
GPT2XLtopk3	0.000 049 41
GPT2XLCS1.0_10	0.000 049 34
Mistral3CS1.0_15	0.000 047 53
GPT2XLCS1.0_5	0.000 044 59
GPT2XLCS1.0_20	0.000 041 33

Table 9: Estimated worth parameter of the extended Bradley-Terry model based on Wikinews dataset and the metric coherence, diversity and perplexity.

For reasons of clarity and comprehensibility, we decided to show here only a snippet, but the full result can be easily and fast obtained by the already stored results in GitHub-repository. Table 8 denotes the worth parameter based on WikiText-103, Table 9 on Wikinews, Table 10 on Books and all three datasets combined can be seen in Table 11. All computations are based on the metrics of perplexity, coherence, and diversity.

Method	Estimated worth parameter
Mistral3CS0.6_10	0.037 29
Mistral3CS0.4_50	0.027 66
Mistral3CS0.6_5	0.027 65
Mistral3CS0.4_10	0.025 90
DeepseekCS0.8_15	0.020 91
Mistral3CS0.4_5	0.020 12
Mistral3CS0.4_15	0.018 89
Falcon2CS0.6_20	0.017 53
DeepseekCS0.6_15	0.016 64
Falcon2CS0.4_20	0.015 55
Qwen2CS0.6_10	0.013 32
Mistral3beam15	0.012 37
Qwen2CS0.4_50	0.012 18
Qwen2beam5	0.011 75
Deepseekbeam5	0.011 75
Mistral3CS0.6_15	0.010 95
Mistral3CS0.6_50	0.010 88
Falcon2beam15	0.010 17
Mistral3beam3	0.009 950
Deepseekbeam15	0.009 685
Deepseekbeam20	0.009 523
Mistral3beam20	0.009 489
Mistral3beam5	0.009 439
...	
DeepseekCS1.0_50	0.000 089 67
Mistral3CS1.0_15	0.000 086 30
GPT2XLCS1.0_3	0.000 085 26
GPT2XLCS0.4_3	0.000 085 26
Qwen2temp0.9	0.000 084 48
Mistral3CS1.0_50	0.000 082 85
GPT2XLCS0.4_5	0.000 082 68
GPT2XLtopp0.6	0.000 078 19
GPT2XLtopk10	0.000 070 44
Falcon2CS1.0_50	0.000 064 77
GPT2XLCS1.0_5	0.000 063 46
GPT2XLCS0.4_20	0.000 059 06
Mistral3CS1.0_20	0.000 056 02
GPT2XLtopk3	0.000 050 49
GPT2XLCS1.0_20	0.000 042 92

Table 10: Estimated worth parameter of the extended Bradley-Terry model based on Book dataset and the metric coherence, diversity, and perplexity.

Method	Estimated worth parameter
Mistral3CS0.4_10	0.038 41
Mistral3CS0.4_5	0.037 66
Mistral3CS0.6_10	0.021 74
Mistral3CS0.4_50	0.020 71
Mistral3CS0.6_15	0.017 05
Mistral3CS0.2_50	0.016 50
Mistral3CS0.6_50	0.016 24
Mistral3beam50	0.014 53
Mistral3beam10	0.013 82
Mistral3beam3	0.013 15
Mistral3beam20	0.013 12
Qwen2beam5	0.012 86
Mistral3CS0.4_1	0.012 60
Mistral3CS0.4_15	0.011 63
Mistral3beam5	0.011 55
DeepseekCS0.6_50	0.011 46
Mistral3CS0.6_20	0.011 31
GPT2XLbeam20	0.010 88
Mistral3CS0.2_3	0.010 81
Mistral3CS0.2_15	0.010 05
Qwen2CS0.6_50	0.009 991
Qwen2beam20	0.009 966
Qwen2CS0.4_50	0.009 659
Mistral3CS0.2_10	0.009 592
Qwen2beam3	0.009 403
LLama3beam20	0.008 993
Mistral3CS0.2_5	0.008 868
Mistral3CS0.6_5	0.008 842
Mistral3CS0.6_1	0.008 508
LLama3beam10	0.008 505
LLama3beam3	0.008 160
Qwen2beam50	0.007 920
LLama3beam5	0.007 636
Qwen2CS0.4_20	0.007 613
Qwen2beam15	0.007 445
Falcon2CS0.6_50	0.007 364
Qwen2beam10	0.007 307
Mistral3CS0.4_3	0.007 242
Qwen2CS0.4_15	0.007 236
GPT2XLCS0.6_10	0.007 113
Mistral3CS0.8_5	0.006 781
Falcon2beam15	0.006 526
LLama3beam50	0.006 246
LLama3beam15	0.006 175
Mistral3beam15	0.006 097
Deepseekbeam10	0.006 073
Mistral3CS0.2_1	0.006 015
Falcon2beam5	0.005 898
DeepseekCS0.8_15	0.005 789
Qwen2CS0.4_5	0.005 717
Falcon2CS0.4_50	0.005 41
Qwen2CS0.2_1	0.005 382
Deepseekbeam3	0.005 328
Qwen2CS0.2_50	0.005 189
Mistral3topp0.7	0.004 943
Falcon2CS0.4_20	0.004 924
Qwen2CS0.2_15	0.004 791
Qwen2CS0.6_20	0.004 779
DeepseekCS0.4_20	0.004 730
GPT2XLbeam5	0.004 724
Mistral3CS0.2_20	0.004 709
Falcon2CS0.2_20	0.004 658
DeepseekCS0.8_10	0.004 637
Falcon2beam50	0.004 589
Deepseekbeam50	0.004 513
Falcon2beam3	0.004 435
Falcon2beam10	0.004 345
Falcon2CS0.4_3	0.004 321
Deepseekbeam15	0.004 298
Falcon2CS0.4_15	0.004 280
Falcon2CS0.4_10	0.004 212
Deepseekbeam5	0.004 125
DeepseekCS0.6_15	0.004 079
Falcon2CS0.6_20	0.003 949
Falcon2CS0.4_1	0.003 893
Qwen2CS0.2_5	0.003 890
Mistral3CS0.4_20	0.003 880
Qwen2CS0.2_20	0.003 746
Falcon2CS0.6_3	0.003 744
Falcon2CS0.6_10	0.003 690
Falcon2CS0.2_50	0.003 651
Falcon2CS0.6_15	0.003 643
Falcon2CS0.2_15	0.003 562
DeepseekCS0.2_10	0.003 527
Falcon2CS0.2_10	0.003 514
DeepseekCS0.2_20	0.003 507

Method	Estimated worth parameter
Qwen2CS0.2_10	0.003 504
Falcon2CS0.2_3	0.003 451
Falcon2beam20	0.003 394
GPT2XLCS0.6_5	0.003 319
DeepseekCS0.2_15	0.003 257
GPT2XLCS0.4_50	0.003 225
Falcon2CS0.2_1	0.003 174
DeepseekCS0.2_3	0.003 153
Deepseekbeam20	0.003 123
Falcon2CS0.4_5	0.003 10
DeepseekCS0.4_10	0.002 934
Falcon2CS0.2_5	0.002 919
Qwen2CS0.4_1	0.002 782
DeepseekCS0.4_50	0.002 636
Qwen2CS0.6_10	0.002 627
Qwen2CS0.8_1	0.002 605
GPT2XLCS0.6_50	0.002 549
GPT2XLbeam10	0.002 522
GPT2XLbeam3	0.002 481
Qwen2CS0.4_10	0.002 480
DeepseekCS0.2_1	0.002 478
Mistral3CS0.6_3	0.002 468
GPT2XLbeam15	0.002 457
GPT2XLCS0.6_50	0.002 549
GPT2XLbeam10	0.002 522
GPT2XLbeam3	0.002 481
Qwen2CS0.4_10	0.002 480
DeepseekCS0.2_1	0.002 478
Mistral3CS0.6_3	0.002 468
GPT2XLbeam15	0.002 457
GPT2XLbeam50	0.002 453
Mistral3topp0.8	0.002 387
Qwen2CS0.6_5	0.002 329
Falcon2CS0.6_5	0.002 317
Qwen2CS0.4_3	0.002 307
DeepseekCS0.2_50	0.002 247
Mistral3topp0.6	0.002 193
Qwen2CS0.6_1	0.002 175
Qwen2CS0.2_3	0.002 161
Falcon2CS0.8_10	0.002 132
Falcon2CS0.8_20	0.002 118
DeepseekCS0.6_1	0.002 094
Mistral3CS0.8_10	0.002 046
DeepseekCS0.4_1	0.002 034
DeepseekCS0.8_20	0.002 019
DeepseekCS0.8_3	0.001 956
GPT2XLCS0.6_20	0.001 950
LLama3temp0.9	0.001 922
GPT2XLCS0.2_50	0.001 921
DeepseekCS0.4_15	0.001 886
GPT2XLCS0.8_1	0.001 874
Falcon2CS0.6_1	0.001 852
DeepseekCS1.0_20	0.001 845
GPT2XLCS0.6_1	0.001 839
GPT2XLCS0.8_15	0.001 816
GPT2XLCS0.4_10	0.001 800
Mistral3CS0.8_1	0.001 785
GPT2XLCS0.6_3	0.001 765
Falcon2temp0.1	0.001 763
Mistral3temp0.5	0.001 761
DeepseekCS0.6_5	0.001 738
LLama3CS1.0_15	0.001 703
LLama3CS0.2_15	0.001 680
GPT2XLCS0.2_5	0.001 660
Deepseektopp0.6	0.001 656
Qwen2topp0.6	0.001 654
LLama3topk15	0.001 619
GPT2XLCS0.8_5	0.001 603
GPT2XLtemp1	0.001 581
Mistral3temp0.3	0.001 557
GPT2XLCS0.2_10	0.001 536
GPT2XLCS0.2_15	0.001 514
LLama3temp0.3	0.001 498
Falcon2topp0.9	0.001 477
DeepseekCS0.6_10	0.001 469
LLama3temp0.7	0.001 464
GPT2XLCS0.2_3	0.001 456
Falcon2topk20	0.001 453
LLama3CS0.2_5	0.001 452
Mistral3topk15	0.001 445
Mistral3temp0.9	0.001 429
Qwen2topp0.95	0.001 419
LLama3CS0.6_5	0.001 408
LLama3CS0.8_5	0.001 403
Mistral3topk5	0.001 397
GPT2XLCS0.4_15	0.001 388

Method	Estimated worth parameter
Qwen2topk1	0.001 352
Deepseektemp0.7	0.001 341
LLama3CS0.4_5	0.001 300
Qwen2CS0.6_3	0.001 296
Falcon2topp0.7	0.001 291
Mistral3topk50	0.001 290
Qwen2CS0.6_15	0.001 279
GPT2XLCS0.2_1	0.001 268
GPT2XLCS0.2_20	0.001 253
LLama3CS0.8_50	0.001 245
Falcon2temp0.3	0.001 222
DeepseekCS0.8_50	0.001 205
LLama3CS1.0_5	0.001 204
Mistral3topp0.9	0.001 192
Qwen2topk15	0.001 186
Falcon2temp1	0.001 177
LLama3CS0.8_15	0.001 173
LLama3CS0.4_50	0.001 167
Qwen2temp0.1	0.001 162
GPT2XLCS0.6_15	0.001 162
DeepseekCS0.4_3	0.001 157
Falcon2topk3	0.001 149
Falcon2CS0.8_3	0.001 141
DeepseekCS1.0_10	0.001 113
LLama3temp0.5	0.001 112
Falcon2topk1	0.001 107
LLama3CS1.0_50	0.001 105
DeepseekCS0.2_5	0.001 089
GPT2XLCS0.4_1	0.001 086
LLama3CS0.6_50	0.001 070
Falcon2topp0.8	0.001 066
LLama3topp0.9	0.001 063
LLama3CS0.6_10	0.000 982 0
Qwen2topp0.7	0.000 969 7
LLama3CS0.4_15	0.000 965 9
LLama3CS0.2_20	0.000 964 1
LLama3CS0.8_10	0.000 959 6
LLama3CS0.4_1	0.000 959 2
GPT2XLCS0.4_5	0.000 958 4
LLama3CS0.8_20	0.000 958 0
Deepseektopk20	0.000 946 3
Mistral3topk20	0.000 927 1
LLama3CS0.6_20	0.000 915 4
Mistral3topk1	0.000 903 3
LLama3CS0.6_3	0.000 902 9
LLama3CS0.2_1	0.000 899 8
Mistral3topk10	0.000 893 4
LLama3CS1.0_1	0.000 889 8
Falcon2CS0.8_50	0.000 887 2
LLama3CS0.8_3	0.000 879 8
LLama3CS0.8_1	0.000 875 4
Falcon2topk50	0.000 872 7
Qwen2CS1.0_1	0.000 871 0
LLama3CS0.2_3	0.000 870 1
LLama3CS1.0_10	0.000 868 3
LLama3CS1.0_3	0.000 867 6
LLama3CS1.0_20	0.000 855 5
Qwen2CS0.8_15	0.000 855 1
Qwen2CS1.0_15	0.000 853 5
LLama3CS0.2_10	0.000 851
Qwen2topp0.8	0.000 849 0
Qwen2temp0.3	0.000 848 9
LLama3topk5	0.000 848 5
Qwen2topk50	0.000 824 3
GPT2XLCS0.4_3	0.000 823 7
LLama3temp0.1	0.000 801 7
Mistral3CS1.0_20	0.000 783 8
LLama3CS0.6_1	0.000 778 7
Qwen2temp0.7	0.000 775 9
Deepseektemp1	0.000 769 5
Falcon2topk10	0.000 741 9
Deepseektopk3	0.000 739 6
Deepseektopk10	0.000 729 7
Mistral3CS1.0_5	0.000 728 9
DeepseekCS1.0_3	0.000 709 0
Qwen2CS0.8_50	0.000 708 7
Mistral3CS0.8_20	0.000 700 6
Falcon2CS0.8_15	0.000 697 9
LLama3CS0.2_50	0.000 691 3
GPT2XLCS0.4_20	0.000 690 4
LLama3topk50	0.000 677 0
Qwen2temp1	0.000 668 9
Falcon2topp0.95	0.000 647 0
LLama3CS0.4_20	0.000 645 5
LLama3topk20	0.000 641 9
LLama3topk3	0.000 641 4
Falcon2topp0.6	0.000 639 5
LLama3topp0.8	0.000 638 9
Qwen2CS0.8_20	0.000 630 9
Mistral3temp0.1	0.000 627 0
LLama3topk1	0.000 625 3
LLama3CS0.4_3	0.000 624 0
Falcon2CS1.0_3	0.000 621 4
LLama3CS0.6_15	0.000 616 3
Qwen2topk20	0.000 615 8

Method	Estimated worth parameter
GPT2XLCS0.8_3	0.000 612 7
Mistral3CS0.8_50	0.000 608 9
Deepseektopk15	0.000 606 3
Falcon2CS1.0_5	0.000 605 5
DeepseekCS1.0_15	0.000 605 3
DeepseekCS0.8_5	0.000 600 0
DeepseekCS0.6_20	0.000 594 9
GPT2XLtopp0.95	0.000 587 7
Qwen2topp0.9	0.000 586 6
LLama3CS0.4_10	0.000 576 7
Deepseektemp0.3	0.000 573 3
LLama3topk10	0.000 571 7
DeepseekCS0.6_3	0.000 558 6
GPT2XLCS0.8_10	0.000 554 1
Mistral3CS1.0_1	0.000 545 8
Deepseektopp0.7	0.000 544 8
LLama3topp0.95	0.000 539 0
Mistral3CS0.8_15	0.000 530 6
GPT2XLtopk1	0.000 529 7
Mistral3topk3	0.000 520 7
Falcon2CS0.8_5	0.000 520 4
Falcon2CS1.0_10	0.000 513 8
Qwen2temp0.5	0.000 505 4
GPT2XLtopp0.7	0.000 499 9
Qwen2CS0.8_10	0.000 487 5
Qwen2topk5	0.000 485 7
GPT2XLCS0.8_20	0.000 480 4
Mistral3topp0.95	0.000 467 1
DeepseekCS0.4_5	0.000 452 2
DeepseekCS1.0_5	0.000 440 4
Falcon2CS1.0_20	0.000 437 5
Qwen2topk10	0.000 436 5
Mistral3temp1	0.000 435 0
GPT2XLtopk5	0.000 426 0
Qwen2topk3	0.000 421 3
Qwen2CS0.8_5	0.000 419 1
GPT2XLtemp0.3	0.000 414 0
LLama3temp1	0.000 409 9
Falcon2temp0.7	0.000 391 6
Falcon2topk15	0.000 388 1
Falcon2temp0.5	0.000 385 6
LLama3topp0.6	0.000 380 3
LLama3topp0.7	0.000 378 4
Falcon2topk5	0.000 376 0
Deepseektemp0.5	0.000 354 5
GPT2XLtemp0.7	0.000 352 1
Mistral3CS0.8_3	0.000 348 0
Deepseektopp0.95	0.000 342 9
Qwen2CS0.8_3	0.000 339 1
Deepseektopk50	0.000 338 5
Deepseektopp0.9	0.000 334 8
Falcon2CS0.8_1	0.000 330 2
Deepseektopp0.8	0.000 329 5
GPT2XLtopk50	0.000 329 1
GPT2XLtopp0.9	0.000 328 7
GPT2XLtemp0.9	0.000 314 9
Qwen2CS1.0_3	0.000 310 9
DeepseekCS0.8_1	0.000 305 6
Mistral3temp0.7	0.000 297 8
GPT2XLCS1.0_3	0.000 297 5
GPT2XLtopk3	0.000 292 3
GPT2XLCS1.0_1	0.000 287 3
Qwen2temp0.9	0.000 285 3
Deepseektopk5	0.000 282 0
Mistral3CS1.0_15	0.000 274 5
Mistral3CS1.0_10	0.000 268 4
Falcon2CS1.0_15	0.000 265 1
Mistral3CS1.0_3	0.000 256 0
GPT2XLtemp0.5	0.000 249 4
Qwen2CS1.0_5	0.000 246 5
GPT2XLtemp0.1	0.000 244 0
GPT2XLCS0.8_50	0.000 241 6
Deepseektemp0.1	0.000 239 2
Falcon2temp0.9	0.000 237 7
GPT2XLCS1.0_50	0.000 233 5
DeepseekCS1.0_50	0.000 231 9
Qwen2CS1.0_50	0.000 224 2
Falcon2CS1.0_1	0.000 222 6
Qwen2CS1.0_10	0.000 222 5
DeepseekCS1.0_1	0.000 222 1
Mistral3CS1.0_50	0.000 212 5
Deepseektopk1	0.000 200 3
Qwen2CS1.0_20	0.000 198 6
Falcon2CS1.0_50	0.000 196 7
GPT2XLtopk10	0.000 187 9
Deepseektemp0.9	0.000 162 1
GPT2XLCS1.0_15	0.000 149 0
GPT2XLtopk15	0.000 134 1
GPT2XLCS1.0_10	0.000 124 6
GPT2XLtopk20	0.000 120 7
GPT2XLtopp0.8	0.000 118 7
GPT2XLtopp0.6	0.000 111 4
GPT2XLCS1.0_5	0.000 097 67
GPT2XLCS1.0_20	0.000 081 80

Table 11: Estimated worth parameter of the extended Bradley-Terry model based on WikiText-103, Wikinews, and Book datasets together and the metric coherence, diversity, and perplexity (2/2).

E Discussion of the Ufg-depth Results

At first glance, this result seems to contradict the number of observations of the partial orders, since the most frequent order, 646 out of 1314, has the lowest depth, and the one with the highest depth is observed only once. But let us take a closer look at the definition of the ufg-depth. The ufg-depth considers subsets of observed partial orders S with size greater than 2, where, in a first step, the number of occurrences is ignored (i.e. not every subset of partial orders is considered, for details see (Blocher et al., 2024)). Then, in a second step, the ufg-depth of a partial order is the proportion of the set S that supports that partial order (e.g. the partial order lies between the intersection and union of S). This proportion is weighted by the proportion of the number of observations corresponding to the partial orders in S. For this dataset, we have that almost all subsets of partial orders do not agree on any dominance structure. Thus, the empty partial order is supported by almost all subsets and, therefore, has such a high depth. Summing things up, the reasons for the low depth value of the most frequent observation are 1) that the number of observations is only considered as a weight and not directly, and 2) that the only subsets S that support this partial order are those that contain the partial order itself in S. Since the partial order corresponding to the highest ufg-depth does not have much in common with other observed partial orders, this set S always implies many other also observed partial orders.[5]

F Results of Q*Text

Based on the Q*Text metric introduced in §5, we can induce a total ordering of decoding methods. Tables 12, 14, 16 and 18 illustrate the results for the most dominant decoding models, strategies, hyperparameters and methods, respectively. On the other hand, We observe in Tables 13, 15, 17 and 19 the results for the least dominant decoding models, strategies, hyperparameters and methods.

Alignment with extended Bradley-Terry In this section, we explore the alignment between the extended Bradley-Terry model and Q*Text through various decoding methods.

[5]Note that this observation can also be made for the second (280 out of 1314) and third (208 out of 1314) most observed partial orders .

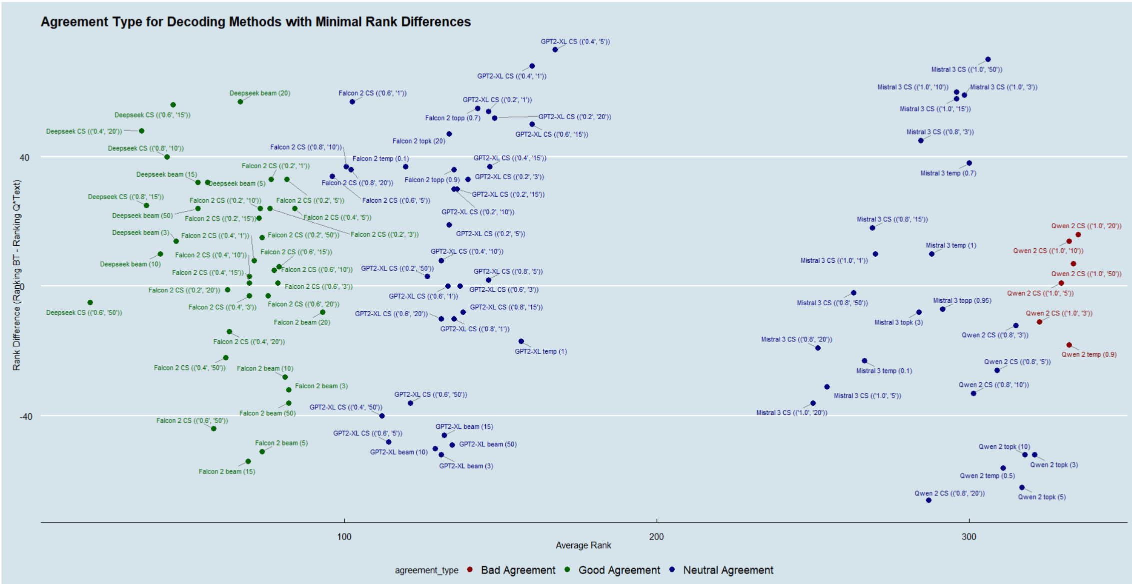

Figure 3: Decoding methods with the smallest rank discrepancies between the extended Bradley-Terry model and Q*Text. Green instances represent decoding methods where both rankings agree on high performance; blue instances indicate agreement on neutrality; and red instances signify agreement on lower quality.

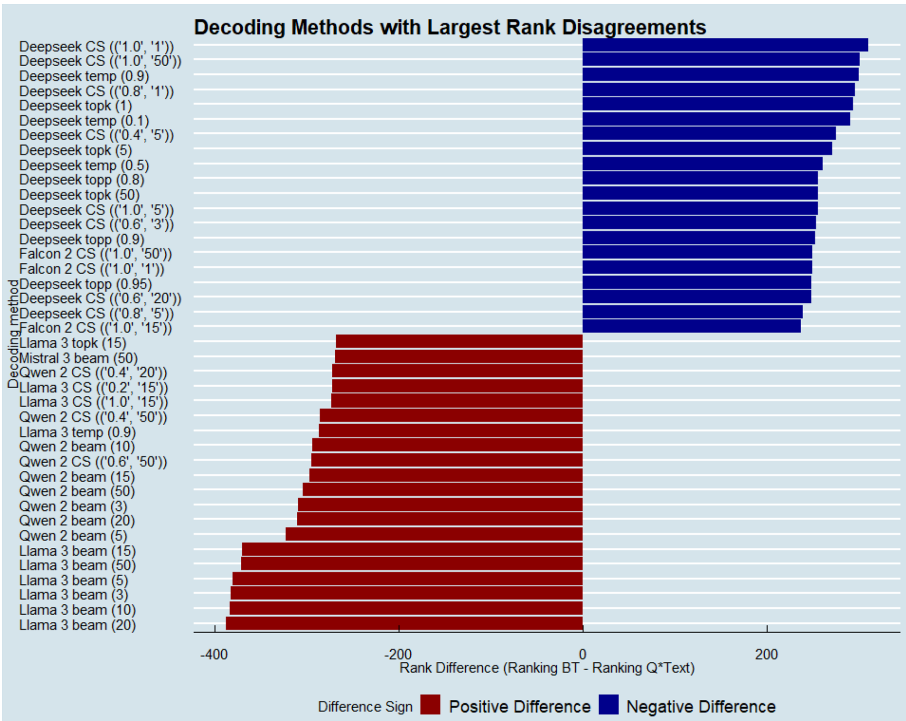

Figure 4: Decoding methods with the largest rank discrepancies between the extended Bradley-Terry model and Q*Text. Here, the extended Bradley-Terry model notably favors low-diversity methods, such as BS, while Q*Text tends to rank highly diverse methods higher. This highlights the differing emphases of each approach on diversity in decoding strategies.

Most Dominant Model	Count	Proportion
Falcon 2	2195	42%
Mistral 3	1471	28%
Qwen 2	904	17%
Deepseek	617	12%
GPT2-XL	55	1%
LLama 3	19	0%
Total	5261	100%

Table 12: Most dominant models based on Q*Text results.

Least Dominant Model	Count	Proportion
GPT2-XL	4050	77%
Qwen 2	703	13%
Llama 3	259	5%
Mistral 3	106	2%
Deepseek	80	2%
Falcon 2	63	1%
Total	5261	100%

Table 13: Least dominant models based on Q*Text results.

Most Dominant Strategy	Count	Proportion
CS	5095	97%
temp	135	3%
topp	16	0%
topk	12	0%
beam	3	0%
Total	5261	100%

Table 14: Most dominant strategies based on Q*Text results.

Least Dominant Strategy	Count	Proportion
CS	4567	87%
beam	652	12%
temp	34	1%
topk	5	0%
topp	3	0%
Total	5261	100%

Table 15: Least dominant strategies based on Q*Text results.

Most Dominant Hyperparameter	Count	Proportion
('0.8', '1')	2138	41%
('1.0', '1')	830	16%
('0.6', '1')	805	15%
('0.8', '5')	360	7%
('0.8', '10')	216	4%
('0.6', '10')	163	3%
('0.8', '3')	89	2%
('0.4', '3')	86	2%
('0.6', '5')	71	1%
0.7	70	1%
('0.8', '15')	64	1%
('0.6', '3')	60	1%
('0.4', '10')	55	1%
0.1	39	1%
('0.2', '10')	34	1%
('0.2', '3')	26	0%
0.3	22	0%
('0.8', '20')	18	0%
('0.4', '1')	17	0%
('0.4', '5')	13	0%
('0.6', '20')	12	0%
('0.6', '15')	11	0%
('1.0', '3')	8	0%
0.5	6	0%
3	6	0%
0.9	6	0%
0.8	6	0%
('0.6', '50')	5	0%
10	5	0%
('0.2', '1')	4	0%
('0.2', '20')	2	0%
('0.2', '5')	2	0%
('0.4', '20')	2	0%
20	2	0%
0.6	2	0%
('0.4', '15')	2	0%
('1.0', '5')	1	0%
50	1	0%
('0.2', '15')	1	0%
15	1	0%
Total	5261	100%

Table 16: Most dominant hyperparameters based on Q*Text results.

Least Dominant Hyperparameter	Count	Proportion
('1.0', '50')	4439	0.84
50	366	0.07
10	99	0.02
15	64	0.01
20	62	0.01
5	40	0.01
('1.0', '20')	39	0.01
('1.0', '15')	30	0.01
('0.8', '50')	27	0.01
3	22	0
0.1	20	0
('0.2', '1')	14	0
0.3	9	0
0.5	5	0
('0.4', '15')	5	0
1	4	0
('0.6', '1')	3	0
('0.4', '50')	3	0
0.7	1	0
0.6	1	0
('0.2', '10')	1	0
0.95	1	0
('0.6', '5')	1	0
('0.8', '10')	1	0
('0.6', '20')	1	0
('0.4', '5')	1	0
('0.4', '3')	1	0
('0.2', '15')	1	0
Total	5261	100%

Table 17: Least dominant hyperparameters based on Q*Text results.

Most Dominant Method	Count	Proportion
Falcon 2_CS (('0.8', '1'))	1083	21%
Mistral 3_CS (('0.8', '1'))	656	12%
Mistral 3_CS (('0.6', '1'))	629	12%
Falcon 2_CS (('1.0', '1'))	510	10%
Falcon 2_CS (('0.8', '5'))	335	6%
Qwen 2_CS (('0.8', '1'))	317	6%
Deepseek_CS (('0.6', '1'))	160	3%
Qwen 2_CS (('0.8', '10'))	148	3%
Deepseek_CS (('1.0', '1'))	141	3%
Qwen 2_CS (('1.0', '1'))	112	2%
Falcon 2_CS (('0.6', '10'))	99	2%
Deepseek_CS (('0.8', '1'))	76	1%
Deepseek_CS (('0.4', '3'))	70	1%
Falcon 2_CS (('0.8', '10'))	68	1%
Falcon 2_CS (('0.6', '5'))	67	1%
Qwen 2_CS (('0.8', '15'))	63	1%
Deepseek_CS (('0.6', '10'))	58	1%
Qwen 2_CS (('0.4', '10'))	48	1%
Mistral 3_temp (0.7)	45	1%
GPT2-XL_CS (('1.0', '1'))	42	1%
Qwen 2_CS (('0.8', '3'))	41	1%
Deepseek_CS (('0.8', '3'))	37	1%
Qwen 2_CS (('0.2', '10'))	32	1%
Mistral 3_CS (('0.6', '3'))	31	1%
Mistral 3_CS (('1.0', '1'))	30	1%
Mistral 3_temp (0.1)	29	1%
Qwen 2_CS (('0.6', '3'))	20	0%
Falcon 2_CS (('0.6', '1'))	19	0%
Deepseek_CS (('0.2', '3'))	19	0%
Deepseek_CS (('0.4', '1'))	17	0%
Qwen 2_CS (('0.8', '20'))	15	0%
Qwen 2_CS (('0.8', '5'))	15	0%
Qwen 2_CS (('0.4', '3'))	15	0%
Mistral 3_CS (('0.8', '3'))	14	0%
Qwen 2_CS (('0.6', '15'))	12	0%
Mistral 3_temp (0.3)	12	0%
Mistral 3_CS (('0.4', '5'))	11	0%
Deepseek_CS (('0.6', '3'))	11	0%
GPT2-XL_CS (('0.8', '1'))	10	0%
Falcon 2_temp (0.7)	10	0%
Qwen 2_topp (0.7)	9	0%
Qwen 2_CS (('0.6', '10'))	9	0%
Qwen 2_temp (0.7)	9	0%
Mistral 3_CS (('0.8', '5'))	8	0%
Qwen 2_temp (0.3)	7	0%
Qwen 2_CS (('0.2', '3'))	7	0%
Qwen 2_temp (0.9)	7	0%
Deepseek_CS (('0.4', '10'))	7	0%
Qwen 2_temp (0.1)	7	0%
Mistral 3_CS (('0.6', '20'))	6	0%
Deepseek_CS (('0.8', '5'))	6	0%
Deepseek_CS (('0.6', '5'))	6	0%
Qwen 2_topk (3)	6	0%
Qwen 2_CS (('1.0', '3'))	6	0%
Deepseek_temp (0.5)	5	0%
Falcon 2_CS (('0.8', '20'))	5	0%
Deepseek_CS (('0.2', '1'))	5	0%
Qwen 2_topp (0.8)	5	0%
Qwen 2_topk (10)	5	0%
Deepseek_temp (0.1)	5	0%
LLama 3_temp (0.3)	4	0%
Total	5261	100%

Table 18: Most dominant methods based on Q*Text results.

Least Dominant Method	Count	Proportion
GPT2-XL_CS (('1.0', '50'))	3821	73%
Qwen 2_CS (('1.0', '50'))	561	11%
LLama 3_beam (50)	130	2%
GPT2-XL_beam (50)	95	2%
Qwen 2_beam (50)	53	1%
Mistral 3_beam (50)	51	1%
LLama 3_beam (10)	38	1%
GPT2-XL_beam (10)	38	1%
Deepseek_CS (('1.0', '50'))	34	1%
Qwen 2_CS (('1.0', '20'))	29	1%
GPT2-XL_CS (('1.0', '15'))	29	1%
LLama 3_beam (20)	27	1%
LLama 3_beam (15)	26	0%
Deepseek_beam (50)	22	0%
LLama 3_beam (5)	18	0%
Mistral 3_CS (('1.0', '50'))	16	0%
Qwen 2_beam (10)	15	0%
Falcon 2_beam (50)	15	0%
Qwen 2_CS (('0.8', '50'))	15	0%
Mistral 3_beam (15)	14	0%
GPT2-XL_beam (20)	10	0%
GPT2-XL_beam (5)	10	0%
GPT2-XL_beam (3)	9	0%
Falcon 2_CS (('1.0', '20'))	9	0%
GPT2-XL_CS (('0.2', '1'))	8	0%
Qwen 2_beam (15)	8	0%
Mistral 3_beam (20)	8	0%
Qwen 2_beam (20)	7	0%
Deepseek_beam (20)	7	0%
Falcon 2_CS (('1.0', '50'))	7	0%
Falcon 2_CS (('0.8', '50'))	7	0%
LLama 3_temp (0.1)	6	0%
Deepseek_beam (15)	6	0%
GPT2-XL_beam (15)	5	0%
GPT2-XL_CS (('0.4', '15'))	5	0%
Falcon 2_beam (15)	5	0%
Qwen 2_beam (3)	5	0%
GPT2-XL_temp (0.1)	5	0%
GPT2-XL_CS (('0.8', '50'))	4	0%
Mistral 3_beam (5)	4	0%
Mistral 3_beam (3)	4	0%
Mistral 3_temp (0.1)	3	0%
LLama 3_temp (0.3)	3	0%
GPT2-XL_temp (0.3)	3	0%
Falcon 2_temp (0.1)	3	0%
Mistral 3_beam (10)	3	0%
Falcon 2_beam (20)	3	0%
GPT2-XL_CS (('0.6', '1'))	3	0%
Deepseek_beam (10)	3	0%
Falcon 2_beam (5)	3	0%
Qwen 2_beam (5)	3	0%
Mistral 3_temp (0.3)	2	0%
Qwen 2_temp (0.1)	2	0%
Falcon 2_beam (10)	2	0%
Deepseek_topk (1)	2	0%
LLama 3_beam (3)	2	0%
LLama 3_CS (('0.2', '1'))	2	0%
Qwen 2_CS (('0.2', '1'))	2	0%
Deepseek_beam (5)	2	0%
Falcon 2_topk (1)	2	0%
Falcon 2_temp (0.5)	2	0%
GPT2-XL_temp (0.5)	2	0%
Qwen 2_topp (0.7)	1	0%
Qwen 2_temp (0.3)	1	0%
LLama 3_CS (('0.6', '20'))	1	0%
LLama 3_temp (0.5)	1	0%
Deepseek_temp (0.1)	1	0%
Falcon 2_CS (('0.4', '5'))	1	0%
GPT2-XL_CS (('0.2', '10'))	1	0%
GPT2-XL_topp (0.95)	1	0%
LLama 3_CS (('0.8', '50'))	1	0%
LLama 3_CS (('0.6', '5'))	1	0%
LLama 3_CS (('0.8', '10'))	1	0%
Deepseek_CS (('0.4', '50'))	1	0%
Qwen 2_CS (('0.4', '50'))	1	0%
LLama 3_topp (0.6)	1	0%
GPT2-XL_topk (3)	1	0%
Falcon 2_CS (('0.4', '50'))	1	0%
Falcon 2_CS (('0.4', '3'))	1	0%
Deepseek_CS (('1.0', '15'))	1	0%
LLama 3_CS (('0.2', '15'))	1	0%
Falcon 2_CS (('0.2', '1'))	1	0%
Falcon 2_beam (3)	1	0%
Deepseek_CS (('1.0', '20'))	1	0%
Mistral 3_CS (('0.2', '1'))	1	0%
Total	5261	100%

Table 19: Least dominant methods based on Q*Text results.

G Q*Text Hyperparameters

Line	Pseudocode: Q*Text Hyperparameter Tuning **Input:** Perplexity, Coherence and Diversity scores (P, C, D)
1	`P_norm = (max(P) - P) / (max(P) - min(P))`
2	`C_norm = (C - min(C)) / (max(C) - min(C))`
3	`D_norm = (D - min(D)) / (max(D) - min(D))`
4	`θ = [1,1,1,0.5,0.5,0.5,1,1,1]`
5	`bounds_w = [[0.1,5],[0.1,5],[0.1,5]]`
6	`bounds_μ = [[0,1],[0,1],[0,1]]`
7	`bounds_α = [[0.1,10],[0.1,10],[0.1,10]]`
8	`for trial in range(max_trials):`
9	`  θ_new = θ + random_normal(0, 0.1)`
10	`  θ_new = clip(θ_new, bounds)`
11	`  for i in range(N):`
12	`    penalty_p = exp(-α_1(P_norm[i]-μ_1)^2)`
13	`    penalty_c = exp(-α_2(C_norm[i]-μ_2)^2)`
14	`    penalty_d = exp(-α_3(D_norm[i]-μ_3)^2)`
15	`    QText[i] = (w_1P_norm[i]penalty_p +`
16	`            w_2C_norm[i]penalty_c +`
17	`            w_3D_norm[i]penalty_d) / (w_1+w_2+w_3)`
18	`  ρ = spearman_corr(QText, Human)`
19	`  if ρ > best_ρ: θ_best = θ_new`
20	`return θ_best`

Table 20: Q*Text Optimization Algorithm

Algorithm explanation: Lines 1-3 normalize metrics to [0,1]. Lines 5-7 define parameter bounds for weights ($w_i \in [0.1, 5.0]$), targets ($\mu_i \in [0.0, 1.0]$), and penalties ($\alpha_i \in [0.1, 10.0]$), this bound definition aims at (i) preventing zero weights while allowing one metric to dominate, (ii) match the normalized metric range, and (iii) ensure positive penalties with reasonable strength. Lines 9-10 perturb parameters with Gaussian noise and clip to bounds. The optimization maximizes Spearman correlation ρ with human ratings.

Parameter	**Symbol**	**Value**
Metric Weights		
Perplexity Weight	w_1	0.586
Coherence Weight	w_2	0.834
Diversity Weight	w_3	3.853
Gaussian Target Values (μ)		
Perplexity Target	μ_1	0.458
Coherence Target	μ_2	0.000
Diversity Target	μ_3	0.854
Gaussian Penalty Strength (α)		
Perplexity Penalty	α_1	2.579
Coherence Penalty	α_2	1.496
Diversity Penalty	α_3	7.370

Table 21: Optimal Q*Text Hyperparameters (Spearman $\rho_s = 0.5545$)

Parameter Interpretation. The optimized parameters reveal insights about text quality assessment.

Diversity dominance: The substantially higher weight for diversity ($w_3 = 3.853$) compared to perplexity ($w_1 = 0.586$) and coherence ($w_2 = 0.834$) indicates that lexical variety is the most discriminative factor for human preferences in our dataset.

Target preferences: The optimal targets suggest humans prefer moderate perplexity levels ($\mu_1 = 0.458$), minimal coherence constraints ($\mu_2 = 0.000$), and high diversity ($\mu_3 = 0.854$).

Penalty sensitivity: The high diversity penalty strength ($\alpha_3 = 7.370$) enforces strict adherence to the diversity target, while the moderate perplexity penalty ($\alpha_1 = 2.579$) and lenient coherence penalty ($\alpha_2 = 1.496$) allow more variation in these two dimensions.

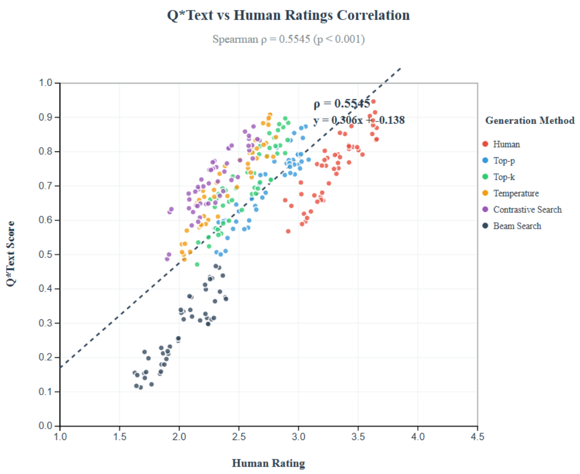

Figure 5: Correlation between Q*Text scores and human ratings across six text generation methods. Each point represents a text sample, colored by generation method. The dashed line shows the linear regression fit. Q*Text achieves a moderate positive correlation (Spearman $\rho = 0.5545$, $p < 0.001$) with human evaluations, demonstrating its effectiveness in capturing human preferences for text quality.