Neural Networks 196 (2026) 108249

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Abstractive summarization through the prism of decoding strategies

Giacomo Frisoni [a,1,*], Luca Ragazzi [a,1], David Cohen [a,1], Gianluca Moro [a,1], Antonella Carbonaro [a,1], Claudio Sartori [a,1]

[a] University of Bologna, Department of Computer Science and Engineering, Via dell'Università, 50, Cesena, 47522, Emilia-Romagna, Italy

ARTICLE INFO

Keywords:
Decoding strategies
Abstractive summarization
Natural language generation
Autoregressive language models.

ABSTRACT

In natural language generation, abstractive summarization (AS) is advancing rapidly due to transformer-based language models (LMs). Although decoding strategies significantly influence generated summaries, their significance is often overlooked. Given the abundance of token selection heuristics and associated hyperparameters, the community needs guidance to make well-informed decisions based on the specific task and target metrics. To address this gap, we conduct a comparative assessment of the effectiveness and efficiency of decoding-time techniques for short, long, and multi-document AS. We explore over 3,500 combinations involving three widely used million-scale autoregressive encoder–decoder LMs, two billion-scale decoder-only LMs, six datasets, and nine decoding settings. Our findings highlight that optimized decoding choices can lead to substantial performance improvements. Alongside human evaluation, we quantitatively measure effects using ten automatic metrics, covering dimensions such as semantic similarity, factuality, compression, redundancy, and carbon footprint. To set the stage for differentiable selection and optimization of decoding options, we introduce PRISM, a first-of-its-kind dataset that pairs AS gold input–output examples with our LM predictions across a diverse range of decoding options. **The code and data are publicly available at** https://github.com/disi-unibo-nlp/prism.

1. Introduction

Abstractive summarization (AS) is one of the most notable and challenging tasks of natural language generation (NLG), aimed at condensing and rephrasing the main points of textual documents (Sharma & Sharma, 2023). With the advent of transformer-based solutions, autoregressive language models (LMs) have repeatedly demonstrated their capabilities to generate human-like summaries (Zhang et al., 2022a). In this rapidly evolving research area, the AS process is typically broken down into two macro-steps: (i) training a neural network to estimate the next-token probability distributions given the input and previously predicted output tokens, (ii) applying an decoding strategy to control how tokens are selected and put together at inference time. Therefore, decoding strategies are considered one of the most significant determinants of AS output quality, also responsible for linguistic properties (Holtzman et al., 2020), prediction n-arity (Vijayakumar et al., 2018), reproducibility (Xu et al., 2023), extrinsic hallucinations (Massarelli et al., 2020; van der Poel et al., 2022), and low information coverage (Meister et al., 2023).

Unfortunately, up to now, AS contributions primarily rely on the conservative use of default decoding settings inherited from previous work (Shen et al., 2022). Sometimes, the choices of the decoding algorithm are presented without much discussion (Guo et al., 2022; Zhang et al., 2022b) or are omitted (González et al., 2022). The lack of systematic practice in thoroughly examining the impact of decoding raises concerns about its underestimation (Gong & Yao, 2023; Ji et al., 2023), driven, among other things, by the increasing number of heuristics and the complexity of text evaluation. Thus, researchers call for large studies to clarify the best decoding practices (Zarrieß et al., 2021). In one of the few available studies, (Meister et al., 2022) have shown that decoding methods exhibit various behaviors depending on the task. This indicates that making general claims in favor of one approach over another may lack a solid foundation. However, an in-depth examination focused solely on AS is still pending.

Contributions. Our paper addresses this gap, performing the first side-by-side investigation into the effect of decoding strategies for AS (Fig. 1), which includes short, long, and multi-document settings. We extensively evaluate 9 well-established decoding heuristics across 3 state-of-the-art

* Corresponding author.
E-mail addresses: giacomo.frisoni@unibo.it (G. Frisoni), l.ragazzi@unibo.it (L. Ragazzi), david.cohen@studio.unibo.it (D. Cohen), gianluca.moro@unibo.it (G. Moro), antonella.carbonaro@unibo.it (A. Carbonaro), claudio.sartori@unibo.it (C. Sartori).

[1] Co-first authorship with equal contribution.

https://doi.org/10.1016/j.neunet.2025.108249
Received 23 December 2024; Received in revised form 9 September 2025; Accepted 20 October 2025
Available online 3 November 2025

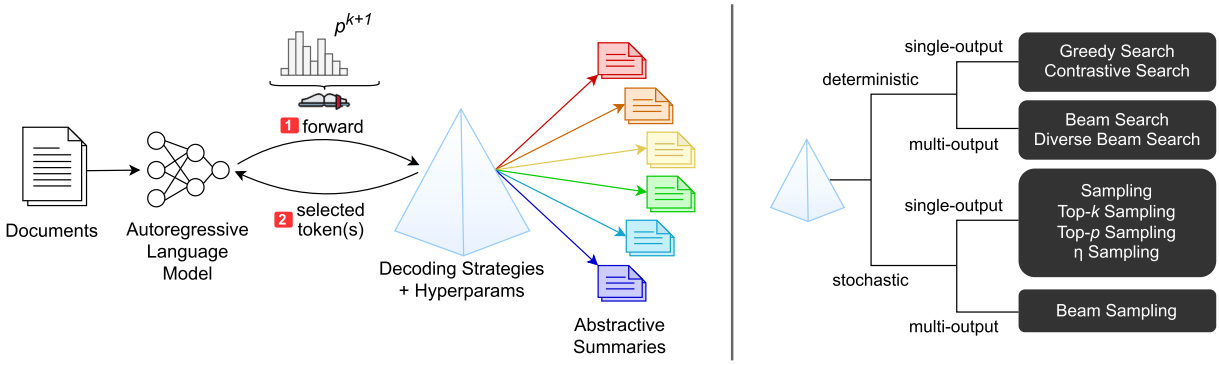


Fig. 1. Conceptual division between modeling and decoding in the neural abstractive summarization pipeline (left). Decoding choices determine the predicted summaries. The taxonomy (right) illustrates the decoding strategies assessed in this study; output n-arity refers to a single generation process.

representative encoder–decoder models and 6 widely used datasets from different domains, exploring a broad spectrum of hyperparameters. In addition to human assessment, we carefully select 10 widely-recognized automatic evaluation metrics from the literature to rigorously examine the relationship between decoding strategies and summary quality. Moreover, we judge efficiency by monitoring the carbon footprint and inference time. Finally, we compare summarization behavior of standard small encoder–decoder models with recent open-source large language models (LLMs).

To guide our investigation, we pose the following research questions:

- **RQ1:** Is there an absolute best decoding method for AS?
- **RQ2:** Are decoding methods sensitive to AS type?
- **RQ3:** To what extent does proper decoding affect LM metrics?
- **RQ4:** Which decoding method provides the best effectiveness-efficiency trade-off?
- **RQ5:** Which hyperparameter values best suit a particular AS quality attribute?
- **RQ6:** Do our findings from encoder-decoder models transfer to recent decoder-only LLMs?

Overall, this work provides a blueprint for the effective use of decoding algorithms and helps AS practitioners make confident choices that suit their needs. Our computational dedication shines through the genesis of PRISM, an innovative dataset in which gold input–output AS examples are accompanied by a large set of LM predictions and their decoding metadata. By aggregating token selections derived from multiple decoding heuristics on top of the LM probability distributions, PRISM introduces a significant opportunity in the literature. It pioneers the prospect of training neural networks to replicate the non-differentiable inner workings of decoding strategies. This is a considerable step toward automatizing their choice and optimizing their hyperparameters for a given benchmark, freeing users from the burdens of space exploration. Broadly, PRISM unlocks new analysis possibilities, helping the community refine the NLG metrics, devise new decoding strategies, and approximate existing ones.[1]

2. Related work

Abstractive Summarization. Transformers have established a strong foundation for AS (Kalyan et al., 2021), designing new large-scale architectures (Chowdhery et al., 2023; Guo et al., 2022), attention mechanisms (Huang et al., 2021; Phang et al., 2023), and pretraining objectives (He et al., 2023; Wan & Bansal, 2022). Starting in 2019,

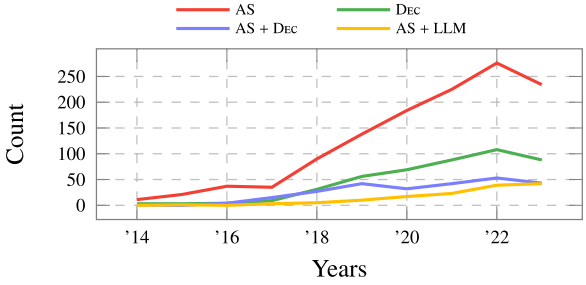


Fig. 2. Annual rate SCOPUS comparison between conference papers on AS, decoding, their intersection, and LLMs. See Appendix A for exact search queries.

AS supports a continuous flow of 140+ yearly publications (Fig. 2).[2] Many of these works show interest in decoding strategies (Ragazzi et al., 2024a), referring to them in the title, abstract, or keywords (cf. the blue line in Fig. 2). Significant progress has been achieved even in low-resource (Moro & Ragazzi, 2022; Moro et al., 2023d) and multi-document (Moro et al., 2022; Ragazzi et al., 2025) scenarios. The latest trends consider the capture of structural properties for neural document modeling (Cao & Wang, 2022) or knowledge injection (Frisoni et al., 2022, 2023). Despite the progress, the need to choose a suitable heuristic remains crucial, motivating our study. Regarding the training modality, reinforcement learning sequence-level rewards have emerged to directly optimize NLG metrics as an alternative to token-level training signals (Frisoni et al., 2023; Ramamurthy et al., 2023). In contrast, this work primarily focuses on popular million-scale encoder–decoder models trained under maximum likelihood regimes. Significantly, despite the rise of decoder-only architectures driven by LLMs, recent findings continue to support the closer alignment of encoder–decoder models with AS tasks (Fu et al., 2023).[3]

Decoding Studies. There is a significant lack of research on the generative effects governed by the available next-token selection strategies. Previous work focused on improving understanding of decoding strategies, providing general overviews (Zarrieß et al., 2021), analyses of human production variability (Giulianelli et al., 2023), or evaluations related to open-ended generation with incomplete optimization processes (Das & Balke, 2022; Holtzman et al., 2020; Su et al., 2022). Only two systematic benchmarks focus on behavior differences between

[1] Code, data, and predictions will be publicly released in case of acceptance (CC-BY-NC-SA 4.0).

[2] We direct the interested reader to (Sharma & Sharma, 2023) and (Syed et al., 2021) for a complete overview of the AS field.

[3] Following a broad literature review, we prioritized the currently dominant encoder–decoder models for AS. However, given the limited exploration of billion-scale networks (15.8 % of works since 2020; see Fig. 2), we also included a comparative analysis with recent, state-of-the-art decoder-only LLMs to assess the transferability of our findings.

tasks and output diversity (Ippolito et al., 2019; Meister et al., 2022). Most comparative studies originate primarily from papers that introduce novel decoding techniques to provide either theoretical justification or empirical evidence for their effectiveness. In addition to the widely adopted strategies such as greedy search, beam search, sampling, and their parameterized variants (e.g., top-k, top-p), several alternative paradigms have also been proposed in the literature. These include approaches that integrate learned discriminators into decoding procedures to guide sequence generation toward human-like outputs, such as Discriminative Adversarial Search (Scialom et al., 2020), or leverage structured search algorithms such as Monte Carlo Tree Search, as adapted for NLG in cooperative decoding settings to balance exploration and exploitation (Scialom et al., 2021b). Although the selection of the decoding strategy is widely known to be task-dependent, community efforts primarily emphasize open-ended generation, leaving several questions unanswered in application fields such as AS. In this area, Beam Search is often treated as the de facto standard, with limited exploration of alternative decoding strategies (Beltagy et al., 2020; Lewis et al., 2020; Xiao et al., 2022). Despite the latest heuristics, we aim to reassess this conventional approach. Furthermore, research efforts that already incorporate AS often draw task-level conclusions based on the analysis of a single dataset and a limited research scope, using concise source documents for the sake of experimental speed (Meister et al., 2022; Su & Collier, 2023). Additionally, efficiency data on runtime and CO_2 emissions are rarely mentioned. To the best of our knowledge, we are the first to (i) thoroughly examine the decoding results for AS in its various forms with respect to input length and n-arity (i.e., number of input documents to summarize), (ii) present findings and recommendations based on a literature-supported grid search tuning for decoding hyperparameter spaces, and (iii) release a pioneering decoding-oriented dataset that opens new research directions. Table 1 summarizes the extensive scale of our analysis. Compared to previous work, our task-specific investigation is counterbalanced by the high diversity of heuristics and their settings. In studies that developed a systematic benchmark for decoding heuristics, the exploration of hyperparameter space was notably limited. Specifically, (Meister et al., 2022) tested 11 variants, while (Ippolito et al., 2019) examined only 9. In contrast, our research has significantly expanded this exploration, testing a total of 3532 combinations of hyperparameters (see Table 3), which provides a more comprehensive understanding and potentially more robust findings. Consequently, we consider this paper to be the most extensive decoding study, particularly with respect to the diversity of AS models evaluated. For example, previous research by (Meister et al., 2022; Su et al., 2022), and (Su & Collier, 2023) considered exclusively one AS model each: BART (Lewis et al., 2020) and GPT-2 (Radford et al., 2019), respectively.

3. Method

Here, we detail the methodologies used in our study. We begin by providing an overview of the investigated AS settings (Section 3.1) and the role of decoding strategies within autoregressive summary generation (Section 3.2). Next, we outline the inclusion criteria for the decoding strategies evaluated in our experiments (Section 3.3).

3.1. Preliminary

We define the problem of AS with the following setup. For single-source tasks, the input is a document $\boldsymbol{x} = \{x_1, \dots, x_{|\boldsymbol{x}|}\}$, where each $x_i \in \boldsymbol{x}$ is a token. For multi-source scenarios, the input is a set $\mathcal{C} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_z\}$ of documents.

Short Document Summarization (SDS). $\boldsymbol{x}$ is a brief text, generally shorter than 1024 tokens, which is a hyperparameter representing the typical input size for transformer-based LMs with quadratic complexity (Lewis et al., 2020; Zhang et al., 2020).

Long Document Summarization (LDS). The number of input tokens can be very large (e.g., > 10,000). Quadratic LMs would ignore important summary-worthy information, leading to quality degradation (Moro & Ragazzi, 2023; Moro et al., 2023c). Therefore, they are replaced by efficient transformers with linear complexity that can process up to 16,384 tokens (Beltagy et al., 2020; Guo et al., 2022; Huang et al., 2021; Phang et al., 2023).

Multi-Document Summarization (MDS). $\mathcal{C}$ is a set consisting of multiple documents related to a topic (e.g., newspaper articles detailing the same recent event). Although there are some promising solutions for handling document relationships (Li & Xu, 2023; Moro et al., 2023e), $\boldsymbol{x}$ is generally created by concatenating the documents in $\mathcal{C}$ to form a single long textual input (DeYoung et al., 2021; Xiao et al., 2022), treating the summarization problem as in LDS.

3.2. Probabilistic summary generation

We consider autoregressive encoder–decoder models, with trainable weights θ, that define the conditional probability $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ of a summary $\boldsymbol{y} = \{y_1, \dots, y_{|\boldsymbol{y}|}\}$ given a variable-length input sequence $\boldsymbol{x}$. Depending on the AS setting, the tokens in $\boldsymbol{x}$ originate from one or more source documents, while the tokens in $\boldsymbol{y}$ are drawn from a finite vocabulary $\mathcal{V}$. The output space of possible summaries is $\mathcal{Y} := \{\text{BOS} \circ \boldsymbol{v} \circ \text{EOS} \mid \boldsymbol{v} \in \mathcal{V}^*\}$, where $\circ$ denotes string concatenation and $\mathcal{V}^*$ is the Kleene closure of $\mathcal{V}$; valid outputs are enclosed by special tokens "begin-of-sequence" and "end-of-sequence." The models follow a local normalization scheme, factorizing the probability of $\boldsymbol{y}$ as follows:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{|\boldsymbol{y}|} p_\theta(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \tag{1}$$

where each $p_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{<t})$ is a distribution over $\bar{\mathcal{V}} := \mathcal{V} \cup \{\text{EOS}\}$ and $\boldsymbol{y}_0 :=$ BOS. Commonly, weights θ are learned by minimizing cross-entropy loss $\mathcal{L}(\theta, C)$ between predicted tokens and ground-truth ones in a training corpus C (negative log-likelihood under p). To penalize overconfident output and combat overfitting, $\mathcal{L}$ can be enhanced by label smoothing, redistributing a certain probability mass of the true token uniformly across all other tokens (Gao et al., 2020). In our setting, the purpose of decoding is to generate a summary hypothesis, $\boldsymbol{y}^*$. Ideally, it should be defined as the one that maximizes the conditional probability, $\boldsymbol{y}^* = \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}} p_\theta(\boldsymbol{y}|\boldsymbol{x})$. This optimization problem is known as maximum a posteriori. As the number of summaries in the symbolic space increases as $|\mathcal{V}|^{|\boldsymbol{y}|}$, the exact search is NP-hard. Moreover, even if a solution were feasible, it would be far from high-quality text (Eikema & Aziz, 2020). Thus, decoding is approximated with heuristic methods.

3.3. Decoding strategies

We test the full range of heuristics supported by HuggingFace as of April 2023,[4] except those aimed at forcing or excluding the generation of specific tokens, which are outside the scope of our AS study. Such algorithms can be configured to satisfy different needs of NLG and fall into two main categories: *deterministic* and *stochastic*. The first category optimizes for summary continuations with high probabilities, while the second introduces an element of randomness into the generation process. Table 2 presents our decoding landscape. Note that all algorithms terminate when y_t or the hypotheses in $\mathcal{Y}_t$ reach EOS for some $t < \max\_length$. We do not consider other early stopping rules because they do not affect the quality of generation (Meister et al., 2020).

4. Experimental setup

We plan a series of generative experiments $(S_\mathcal{H}, \mathcal{D}, \mathcal{M})$, where $S_\mathcal{H}$ denotes a decoding strategy with fixed hyperparameters, $\mathcal{D}$ is the dataset

[4] https://huggingface.co/docs/transformers/internal/generation_utils

Table 1
Comparison of related work on decoding strategies.

Source	Decoding Strategies	Hyperparameters*	Task**	Metrics	Sys†
(Holtzman et al., 2020)	Greedy Search	/	OEG	Perplexity, Self-BLEU, Repetition, Zipf Coefficient	✗
	Beam Search	$b \in \{4, 8, 16\}$			
	Sampling	$\tau \in [0.1, 0.2, \dots, 1.0]$			
	Top-k Sampling	$k \in 5 \times 2^{[1,2,\dots,12]}$			
	Top-p Sampling	$\approx p \in [0.1, 0.2, \dots, 0.9, 0.95]$			
(Meister et al., 2022)	Greedy Search	/	OEG, MT, SDS, DG, SG	BLEU, COMET ROUGE, BLEURT DIST-n, ENT-n n-GRAM DIV. Self-BLEU Repetition	✔
	Beam Search	$b \in \{5, 10\}$			
	Diverse Beam Search	Δ = Hamming, $\lambda = 0.7$, $\lvert g \rvert = b = 5$			
	Sampling	/			
	Top-k Sampling	$k = 30$			
	Top-p Sampling	$p = .85$			
(Ippolito et al., 2019)	Beam Search	$b = 10$	DG, IC	Perplexity, DIST-n, ENT-n, SPICE	✔
	Diverse Beam Search	Δ = Hamming, $\lambda = 0.8$, $\lvert g \rvert = b = 10$			
	NPAD Beam Search	$b = 10$, $\sigma_0 = 0.3$			
	Clustered Beam Search	$b = 10$, $c = 5$			
	Top-g Capping Beam Search	$b = 10$, $g\_c = 3$			
	Iterative Beam Search	$b = 10$, $i = 5$			
	Sampling	$\tau \in \{0.5, 0.7, 1.0\}$			
(Leblond et al., 2021)	Greedy	/	MT	BLEU, BERTScore	✗
	Beam Search	$b \in [2, 4, \dots, 10, 20]$, logits $\tau \in [0.6, 0.8, \dots, 1.4]$ normalization $\tau \in [0.4, 0.6, \dots, 1.0]$			
	Value-Guided Beam Search	$b = 6$, logits $\tau \in [0.6, 0.8, \dots, 1.4]$, $\alpha_{lc} \in [0, 0.1, \dots, 0.9, 0.95, 1.0]$			
	Monte Carlo Tree Search	logits $\tau \in \{0.9, 1.1, 1.3\}$, $c_{puct} \in [1.0, 2.0, \dots, 6.0, 8.0]$			
	Sampling + Ranking Variants	$\tau \in [0.15, 0.25, \dots, 0.95]$			
(Su & Collier, 2023; Su et al., 2022)	Greedy	/	OEG, DG, SDS, CG, MT	Perplexity, Accuracy, Repetition, n-GRAM DIV., MAUVE, Sem. Coherence	✗
	Contrastive Search	$k \in [2, 3, \dots, 10]$, $\alpha \in [0.1, 0.2, \dots, 1]$			
	Beam Search	$b \in \{4, 5, 10\}$			
	Typical Sampling	$\tau = 0.95$			
	Top-k Sampling	$k = 50$			
	Top-p Sampling	$p = .95$			
Ours	Greedy	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$	SDS, LDS, MDS	ROUGE, BERTScore, Perplexity, Coverage, Density, Compression, UNR, NID, BARTScore-F, Carburacy, Runtime	✔
	Contrastive Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $k \in [20, 30, \dots, 60]$, $\alpha \in [0.2, 0.4, \dots, 1]$			
	Beam Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $b \in [2, 3, \dots, 10]$			
	Diverse Beam Search	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, Δ = Hamming, $\lambda \in [0.2, 0.4, \dots, 1.0]$, $\lvert g \rvert = b \in [2, 3, \dots, 10]$			
	Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$			
	Top-k Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$, $k \in [20, 30, \dots, 60]$			
	Top-p Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\tau \in \{0.8, 0.9, 1.0\}$, $k \in$ [None, 20, 30, … , 60] $p \in 0.4, 0.6, 0.8$			
	Beam Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $b \in [2, 3, \dots, 10]$, $p = .9$			
	η Sampling	no_repetition_ngram_size $\in \{2, 3, 4, 5\}$, $\eta \in \{0.0003, 0.0006, 0.0009, 0.002, 0.004\}$			

* b = beam size, τ = temperature, k/p = top-k/p thresholds for selecting candidate pools, Δ = diversity measure, λ = diversity penalty, $\lvert g \rvert$ = number of sub-groups, σ_0 = decoder-level random noise, c = number of clusters, g_c = top parent hypotheses considered at each time step, i = number of iterations, α_{lc} = linear combination weights, η = entropy-dependent cutting-off.
** OEG = Open-Ended Generation, MT = Machine Translation, DG = Dialogue Generation, SG = Story Generation, IC = Image Captioning, CG = Code Generation, SDS = Short Document Summarization, LDS = Long Document Summarization, MDS = Multi-Document Summarization.
† ✔= Publication focused on proposing a systematic benchmark to compare decoding methods; ✗= Otherwise (e.g., new heuristic).

Table 2
Summary of the decoding strategies benchmarked in this AS study. References occur in table are cited here (Su et al., 2022) [36], (Lowerre, 1976) [55] (Vijayakumar et al., 2018) [4], (Graves, 2013) [56], (Fan et al., 2018) [57], (Han et al., 2019) [58], (Holtzman et al., 2020) [3], (Caccia et al., 2020) [59], (Hewitt et al., 2022) [60].

DECODING STRATEGIES[*,†]		
Greedy Search	**Contrastive Search** [36]	**Beam Search** [55]
It selects the most probable token at each t (locally-optimal decision): $y_t = \underset{y \in \bar{\mathcal{V}}}{\text{argmax}} \log p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ (for $t > 0$)	It extends Greedy Search by jointly considering the model confidence (i.e., semantic coherence) and the similarity w.r.t. the previous context: $y_t = \underset{y \in \bar{\mathcal{V}}^{(k)}}{\text{argmax}} \{(1-\alpha) \times p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) - \alpha \times (\max\{s(h_y, h_{y_j}) : 1 \le j \le t-1\})\}$ (for $t > 0$) $\bar{\mathcal{V}}^{(k)}$ is the set of top-k predictions from the LM's probability distribution ($k \in \mathbb{Z}_+$). The second term is a degeneration penalty governed by $\alpha \in [0, 1]$; it is calculated with the cosine similarity $s(\cdot, \cdot)$ between the candidate representation h_y and the prior tokens. When $\alpha = 0$, Contrastive Search degenerates to Greedy Search.	A pruned breadth-first search algorithm that expands the hypotheses in a greedy left-to-right way, retaining the top-b candidates at each t: $Y_t = \underset{\substack{Y' \subseteq \mathcal{B}_t \\ \lvert Y' \rvert = b}}{\text{argmax}} \mathcal{S}(Y'_t)$ (for $t > 0$) $\mathcal{B}_t$ denotes all possible roll-out extensions $\{\boldsymbol{y}_{t-1} \circ y \mid y \in \bar{\mathcal{V}}$ **and** $\boldsymbol{y}_{t-1} \in Y_{t-1}\}$. $\mathcal{S} : \mathcal{Y} \to \mathbb{R}$ is a scoring function on $Y \subseteq \mathcal{Y}$. By default, $\mathcal{S}(Y) = \sum_{y \in Y} \log p_\theta(\boldsymbol{y} \mid \boldsymbol{x})$. $\boldsymbol{y}*$ is chosen from the final set Y_T.
Diverse Beam Search [4]	**Sampling** [56]	**Top-k Sampling** [57]
In vanilla Beam Search, as t and b increase, the candidate summaries of a single beam gradually occupy the top positions. This inner functioning results in a high overlap among hypotheses that often differ only by punctuation and slight morphological variations [58], indicating poor coverage of the search space. Diverse Beam Search splits a beam into sub-groups sequentially optimized and adds a penalty in $\mathcal{S}$ to encourage inter-group dissimilarity: $\mathcal{S}(Y_t^{(g)}) = \sum_{\boldsymbol{y} \in Y} \log p_\theta(\boldsymbol{y} \mid \boldsymbol{x}) - \lambda \sum_{g' < g} \Delta\left(\boldsymbol{y}, Y_t^{(g')}\right)$ $\Delta(\boldsymbol{y}, Y_t^{(g')})$ is a diversity measure whose strength is controlled by λ, and g is a sub-group.	Instead of approximating $\boldsymbol{y}^*$, it makes a random selection at each t, thus giving a non-zero chance to every token following the model distribution ($\sim$): $y_t \sim p_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ (for $t > 0$) Softmax can be re-calibrated through a temperature parameter $\tau \in (0, 1]$. By decreasing τ (1 = no effect), we skew the distribution towards likely tokens and reduce the mass in the unreliable tail: $p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) = \frac{\exp(\frac{u}{\tau})}{\sum_j \exp(\frac{u_j}{\tau})}$ where $u \in U$ are logits.	It limits the sampling space to the top-k probable tokens, regardless of the distribution shape. Here, the decoding maximizes: $\sum_{y \in \bar{\mathcal{V}}^{(k)}} p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$ where $\bar{\mathcal{V}}^{(k)} \subseteq \bar{\mathcal{V}}$.
Top-p (Nucleus) Sampling [3]	**Beam Sampling** [59]	**η Sampling** [60]
Instead of assuming a fixed-sized shortlist, it dynamically expands and contracts the candidate pool according to the shape of the distribution (e.g., fewer contenders if sharp). When many next tokens are plausible, the allowed set reflects that. Formally, it samples from the smallest subset of tokens whose cumulative probability mass exceeds a chosen threshold $p \in (0, 1]$: $\sum_{y \in \bar{\mathcal{V}}^{(p)}} p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) \ge p$	It is a variant of Beam Search where, at each t, the b next tokens for constructing $\mathcal{B}$ are sampled conditioned on the current hypotheses (no local optimum).	It samples tokens above an entropy-dependent probability threshold to avoid unnecessary cutting-offs. Eligible tokens come from: $\{y \in \bar{\mathcal{V}} \mid p_\theta(y \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) > \eta\}$ $\eta = \min(\epsilon, \alpha \exp(-e_{\boldsymbol{x}, \boldsymbol{y}_{<t}}))$ where $\exp(-e_{\boldsymbol{x}, \boldsymbol{y}_{<t}})$ is the expected next token probability given the entropy $e_{\boldsymbol{x}, \boldsymbol{y}_{<t}}$, scaled by a constant α. To expose a single hyperparameter, α is set to $\sqrt{\epsilon}$. The general principle is to only truncate tokens whose probabilities are low relative to the rest of the distribution or an absolute threshold.

[*] Basic notation. $\boldsymbol{x}$ = document(s), $\boldsymbol{y}$ = summary, θ = model weights, t = decoding step, $\bar{\mathcal{V}} := \mathcal{V} \cup \{\text{EOS}\}$, p_θ = model probability distribution, $\mathcal{Y}$ = hypothetical summaries, b = beam size, $\boldsymbol{y}*$ = best summary.
[†] ● = deterministic, ● = stochastic.

that presents specific AS challenges and length constraints, and $\mathcal{M}$ is the autoregressive LM. In total, we executed 3532 runs (see Table 3) over 81 GPU days. To the best of our knowledge, we conduct the most detailed hyperparameter sweep in the NLG literature. We use grid search to simultaneously alter the generative variables of each heuristic, selecting range values based on insights from previous studies (see Appendix B).

Environment. Inference runs are performed on a workstation that has 4 Nvidia GeForce RTX3090 GPUs with 24 GB of dedicated memory each, 64 GB VRAM, and an Intel® Core™ i9-10900X1080 CPU @ 3.70GHz. Our code is based on PyTorch 1.10.2 (Paszke et al., 2019), the HuggingFace Transformers (Wolf et al., 2019) and Datasets (Lhoest et al., 2021) libraries. All decoding runs for autoregressive summary generation are subject to the "`generate()`" method of HuggingFace. We set the global seed to 42 to ensure reproducibility. We tracked all our executions with Weights & Biases[5] and monitored CO_2 emissions with CodeCarbon.[6] As for LLMs, we leverage the vLLM (Kwon et al., 2023) library to accelerate batched inference.

4.1. Datasets

We consider the test sets of 6 notable, English-language, domain-specific, and open-access corpora, 2 for each AS family. Table 4 lists dataset references.

[5] https://wandb.ai
[6] https://github.com/mlco2/codecarbon

Table 3
Hyperparameters for each decoding strategy. The total number of runs is 2656 for encoder-decoder models; 876 for decoder-only LLMs.

HYPERPARAMS*	Deterministic				Stochastic				
	Greedy	Contrastive	Beam Search	Diverse Beam Search	Sampling	Top-k Sampling	Top-p Sampling	η Sampling	Beam Sampling
`no_rep_ngram_size`†					{2, 3, 4, 5}				
`early_stopping`			✓	✓					✓
`diversity_penalty`				0.{2, 4, 6, 8}, 1.0					
`num_beams`			[2, 3, …, 10]	[2, 3, …, 10]					[2, 3, …, 10]
`temperature`					0.{8, 9}, 1.0	0.{8, 9}, 1.0	0.{8, 9}, 1.0		
`top_k`		[20, 30, …, 60]				[20, 30, …, 60]	[/, 20, …, 60]		
`top_p`							0.{4, 6, 8}		0.9
`penalty_alpha`		0.{2, 4, 6, 8}, 1.0							
`eta_cutoff`								$\{4, 2\}\times10^{-3}$, $\{9, 6, 3\}\times10^{-4}$	
`do_sample`					✓	✓	✓	✓	✓
# Encoder-decoder Runs	16‡	400	144	720	48	240	864	80	144
# Decoder-only LLM Runs	12‡				36	180	648		

* {…} = sets, […] = series, ✓ = true, / = none.
† Considered only for encoder-decoder models. orange and cyan initialization schemes are tested on 2 and 4 datasets, respectively.
‡ Encoder-decoder calculation example. Note that we test one specialized model for each dataset.
(2 `no_rep_ngram_size` settings × 2 datasets) + (3 `no_rep_ngram_size` settings × 4 datasets) = 16.
Decoder-only LLM calculation example. Note that we test two foundational non-specialized LLMs for each dataset.
2 models × 6 datasets = 12.

Table 4
List of the utilized datasets and relative length constraints.

Dataset	URL	Input	Output	
		Max Len	Min Len	Max Len
XSum	https://huggingface.co/datasets/xsum	1024	20	100
CNN/DM	https://huggingface.co/datasets/cnn_dailymail	1024	20	100
PubMed	https://huggingface.co/datasets/ccdv/pubmed-summarization	4096	100	256
arXiv	https://huggingface.co/datasets/ccdv/arxiv-summarization	4096	100	256
Multi-News	https://huggingface.co/datasets/multi_news	4096	100	256
Multi-LexSum	https://huggingface.co/datasets/allenai/multi_lexsum	4096	50	256

- *SDS*. **XSum** (Narayan et al., 2018), a dataset of 2010–2017 BBC articles, designed for highly abstractive one-sentence summarization. **CNN/DM** (Nallapati et al., 2016), a collection of reports from different news outlets accompanied by short-sentence highlights written by journalists.
- *LDS*. **PubMed** and **arXiv** (Cohan et al., 2018), two datasets of lengthy and structured scientific papers along with their abstracts.
- *MDS*. **Multi-News** (Fabbri et al., 2019), composed of news articles and human-written summaries from `newser.com`, which aggregates content from hundreds of US and international sources. **Multi-LexSum** (Shen et al., 2022), a collection of expert-authored summaries of court documents from federal civil rights lawsuits. Given the multi-granularity nature of summaries, we consider the D → S task, i.e., synthesizing the source texts into a short summary.

Data Preprocessing. Following best practices in summarization dataset building (Kornilova & Eidelman, 2019), our preprocessing pipeline involves (i) lowercasing, (ii) ASCII encoding, (iii) removal of extra spaces, URLs, special characters, and bullet points, (iv) HTML cleaning, (v) deletion of dash sequences, (vi) erasure of newlines and tabs, (vii) removal of empty sentences and non-alphabetic starters, (viii) ensuring space between sentences, (ix) word contraction expansion, and (x) normalization of quotes.

Data Sampling. Given the wide research space for decoding and evaluation, analyzing entire datasets is impractical due to computational constraints. Thus, we consider a representative subset from each corpus. A prudent preliminary step is to statistically determine the required sample size to detect significant metric effects across decoding runs, if they exist, with a specified degree of confidence (the metrics studied are listed in Table 9). We conducted multiple power analyses to help ensure that our study is neither underpowered (risking missing true effects) nor overpowered (wasting resources on unnecessarily large samples). A single analysis considers the following key concepts.

- Power level, the probability of correctly rejecting the null hypothesis when it is false. In our study, the null hypothesis is that there is no significant difference in performance between two decoding strategies for a given metric. For example, "There is no significant difference in ROUGE scores between Greedy decoding and Beam Search." So, rejecting the null hypothesis means concluding that there is indeed a significant metric-driven difference between the decoding strategies. Higher power reduces the risk of Type II errors (false negatives), but typically requires larger sample sizes. In our study, we set the power level to the commonly recommended value of 0.8, which means that we aim for an 80 % chance of identifying real performance differences. Using this power-level setting in our analysis, we determined a sample size that allowed us to draw reliable conclusions about the relative effectiveness of the decoding strategies explored.
- Significance level, the probability of incorrectly rejecting the null hypothesis when it is true. We set it to 0.05, i.e., 5 % chance of a Type I error (false positive).
- Effect size, the magnitude of the difference we expect to observe between the metric scores of diverse decoding strategies. We used Cohen's d, calculated by taking the difference in means between two groups and dividing it by the pooled standard deviation—the average variability in the groups being compared, i.e., $(\bar{x}_1-\bar{x}_2)/s$. It allows for a standardized comparison of metric-based contrasts across a couple of decoding strategies. Cohen's d is commonly interpreted with a small effect around 0.2, a medium effect near 0.5, and a large effect at 0.8 or greater. Smaller effect sizes indicate subtle differences that are harder to detect reliably. The smaller the effect size, the larger the required sample.

We followed an iterative method to balance statistical rigor with computational efficiency, running power analyses on a small sample of the target datasets and expanding the sample size based on the results. To estimate the required sample size, we selected a 1 % subset from each dataset using proportional stratified random sampling without replacement, ensuring the same distributional characteristics as the original datasets. We stratified based on the document and summary length; tertiles were calculated to assign {short, medium, long} classes. For MDS, we also considered the number of source documents. In the aggregated samples (421 documents in total), we conducted a prelim-

Table 5
Maximum required sample size per dataset.

Dataset	# Samples
XSum	100.52
Cnn/Dm	393.40
PubMed	114.34
arXiv	180.12
Multi-News	244.35
Multi-LexSum	79.88

inary small-scale study, generating summaries and computing metrics using the $(S_H, D, \mathcal{M})$-structured runs previously discussed.

Using this data, we computed the effect sizes for all combinations of metrics and decoding strategy pairs on the AS benchmarks of interest. We then conducted a series of power analyses designed for t-tests to estimate the required sample size for each metric–dataset combination, using the average absolute effect size across all decoding comparisons. We used the `statsmodels` Python library (v 0.14.0) for these analyses. Finally, we determined the larger sample size required for each dataset as a conservative approach (see Table 5). We observed that the metrics requiring the most samples on average are ROUGE and UNR. The detailed procedure is presented as pseudocode in Algorithm 1. This initial analysis provided valuable information on the effect sizes and variability within our data. The findings highlighted the need for a larger sample size to achieve the desired statistical power, prompting us to expand our sample accordingly. Given that the required larger sample size is approximately 393, we ensured this minimum threshold and, as a precaution, selected 10 % of instances for each dataset (e.g., 1134 for XSum, 1148 for Cnn/Dm). We expanded the dataset using stratified sampling following the same criteria. Due to the small size of the test set in Multi-LexSum (616 in total), we decided to retain it completely.

After preprocessing, a summary of the statistics of the datasets (before and after sampling) is presented in Table 6. The number of words is derived from `nltk.word_tokenize` (Bird, 2006).

Algorithm 1 Power analysis and sample size calculation.

```
Load dataset D containing decoding runs and metrics
Define set of metrics M
for each dataset d ∈ D do
    for each metric m ∈ M do
        for each pair of decoding strategies (s1, s2) do
            Calculate mean difference x̄_s1 − x̄_s2
            Calculate pooled standard deviation s_p
            Compute effect size ES = (x̄_s1 − x̄_s2) / s_p
        end for
        Compute average absolute effect size |ES| for (d, m)
        Calculate required sample size n using power analysis:
        n ← PowerAnalysis(|ES|, power = 0.8, α = 0.05)
    end for
    Compute maximum required sample size n̄_d across all metrics for dataset d
end for
```

4.2. Models

For our principal analysis within the scope of RQ1–RQ5, we use three comparable state-of-the-art transformer-based models in their large versions, for which original fine-tuned weights on the datasets in Section 4.1 already exist (see Table 7). **Bart** (Lewis et al., 2020) (SDS, 406M parameters) has quadratic memory complexity relative to input size, limiting it to sequences up to 1024 tokens. **Led** (Beltagy et al., 2020) (LDS, 447M parameters) uses sparse attention to achieve a linear input scale, processing up to 16,384 tokens. **Primera** (Xiao et al., 2022) (MDS, 447M parameters) adapts Led to multi-inputs through a summarization-specific pretraining objective, concatenating the sources with a special token and forming a single input of up to 4,096 tokens. According to (Xiao et al., 2022), the input for Primera is the concatenation of documents within the sets (in the same order), where truncation is applied to each document based on the input length limit divided by the number of documents, ensuring representation of all sources. To assess the generalizability to LLMs (RQ6), we considered two recent foundational, instruct-tuned models. **Qwen2.5-7B** (Yang et al., 2024) is part of the LLM family developed by Alibaba Cloud and is specifically optimized for enhanced reasoning capabilities. **LLaMA-3.1-8B** (Dubey et al., 2024) is developed by Meta and follows conventional pretraining without explicit optimization for reasoning enhancement. We note that, unlike encoder–decoder models, tested LLMs did not undergo a preliminary fine-tuning on the training sets of the target datasets. Additionally, we evaluate them in zero-shot condition without in-context examples.

4.3. Evaluation

Automatic. Estimating the quality of an AS system is a task that requires a level of complexity comparable to that of generating accurate summaries. Both tasks involve a deep understanding of the text and the ability to assess key aspects such as informativeness and factuality. To ensure good practice, we use a comprehensive set of relevant metrics that capture distinct attributes (Table 9). We quantify metric scores using HuggingFace Evaluate (von Werra et al., 2022), and use external official repositories where necessary.[7] Table 8 lists all hyperparameters.

We use the GPT-2 model's perplexity score to assess the naturalness of generated text.[8] Perplexity measures how well the model predicts a text sequence, with lower scores indicating more natural language. This allows us to compare the fluency of text produced under different hyperparameter configurations. We computed BERTScore with DeBERTa-large instead of the default RoBERTa-large, as recommended by the authors from version 0.3.11, due to its greater correlation with human judgment. We acknowledge the availability of factuality metrics such as SummaC (Laban et al., 2022), which is particularly suited for abstractive LDS (Koh et al., 2022). Yet, after preliminary tests, the computational demands were too time-consuming, leading us to prefer BARTScore. We also acknowledge alternative evaluation paradigms such as QuestEval (Scialom et al., 2021a), which estimates summary quality via question generation and answering, and MAUVE (Pillutla et al., 2021), which quantifies the divergence between the distributions of generated and reference texts. While both have shown promising results in specific settings, we focus here on widely adopted and standardized metrics to ensure comparability and reproducibility across decoding strategies.

Human. To effectively assess the merits of decoding, we conduct a thorough human evaluation. Inspired by (Fabbri et al., 2019; Narayan et al., 2018), and (Huang et al., 2023), we use a direct comparison strategy, which is more reliable, sensitive, and less labor-intensive than traditional rating scales. By asking evaluators to make relative judgments rather than absolute ones, we reduce their cognitive load, particularly when assessing complex dimensions of summary quality. This comparative technique allows us to capture nuanced differences that may not be apparent through individual assessments. We sample 20 documents from each dataset and select the three best decoding strategies for each dataset based on the average score of effectiveness-oriented metrics (ROUGE, BERTScore, and BARTScore). This approach allows us to focus on strategies that are more practical for the AS community while

[7] https://github.com/lil-lab/newsroom for Coverage, Density, and Compression. https://github.com/Wendy-Xiao/redundancy_reduction_longdoc for UNR and NID.

[8] We chose GPT-2 because it is the standard causal language model for perplexity evaluation in HuggingFace (von Werra et al., 2022).

Table 6
Dataset statistics before and after sampling, including test set size, number of sources per instance, and the words in the source and target texts. All values are averaged except "# Samples".

Before Sampling			Source		Target	After Sampling			Source		Target
	Dataset	# Samples	# Docs	# Words	# Words		Dataset	# Samples	# Docs	# Words	# Words
SDS	XSum	11,333	1	421.2	22.9	*SDS*	XSum	1134	1	414.1	22.8
	CNN/DM	11,490	1	761.3	57.4		CNN/DM	1148	1	753.8	58.0
LDS	PubMed	6658	1	3,070.4	206.7	*LDS*	PubMed	667	1	3,095.5	207.2
	arXiv	6440	1	5,733.2	161.6		arXiv	644	1	5,770.7	160.6
MDS	Multi-News	5621	2.8	2,026.3	245.9	*MDS*	Multi-News	555	2.8	1,947.9	248.0
	Multi-LexSum	616	10.3	88,122.9	126.5		Multi-LexSum	616	10.3	88,122.9	126.5

Table 7
List of the utilized models.

Model	Specialization Dataset	URL
BART-large	XSum	https://huggingface.co/facebook/bart-large-xsum
	CNN/DM	https://huggingface.co/facebook/bart-large-cnn
LED-large	PubMed	https://huggingface.co/patrickvonplaten/led-large-16384-pubmed
	arXiv	https://huggingface.co/allenai/led-large-16384-arxiv
PRIMERA-large	Multi-News	https://huggingface.co/allenai/PRIMERA-multinews
	Multi-LexSum	https://huggingface.co/allenai/primera-multi_lexsum-source-short
LLAMA-3.1-8B	/	https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
QWEN2.5-7B	/	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Table 8
Hyperparameters of NLG metrics.

Metric	Hyperparameters
ROUGE	rouge_types=["rouge1","rouge2","rougeLsum"], use_stemmer=True
BERTScore	lang="en", model_type="microsoft/deberta-large-mnli", idf=True, batch_size=32, rescale_with_baseline=True
Perplexity	model_id="gpt2", add_start_token=False
BARTScore	checkpoint="facebook/bart-large-cnn", batch_size=4
Carburacy	alpha=10, beta=100

keeping the evaluation manageable. While comparing identical strategies across datasets could offer insights, our approach of selecting top strategies per dataset aims to evaluate the most effective approaches for each specific data type, maximize evaluation efficiency, and identify both universally effective and context-specific strategies. This setting mirrors real-world optimization practices in the AS community. Three English-proficient AS researchers, external to the author group, evaluate the summaries produced by the top-performing strategies of each dataset. For each document, human experts independently label all possible pairs of artificial summaries, comparing one pair of strategies at a time. The names of the decoding strategies are kept hidden from the evaluators to ensure a blind assessment. In addition, we randomize the position of each strategy within a pair and the combination (pair) order to prevent bias. Experts judge each summary pair on four dimensions: *Recall*: whether the generated summary covers all the target content. *Precision*: whether the generated summary covers only the target content (i.e., no redundant or superfluous information). *Faithfulness*: whether the generated summary is factually consistent with the input document. *Fluency*: the grammaticality and coherence of the summaries. For each dimension, experts select the better summary or declare a "tie" if both are of equal quality, whether satisfactory or unsatisfactory. In total, each expert provides 1440 preference labels (6 datasets × 20 documents × 3 strategy-distinguished summary pairs × 4 rating dimensions). Thus, all possible pairs are judged by each expert, and each summary pair is annotated by all experts (4,320 total labels). The final score for each decoding strategy is calculated as the percentage of times its summaries are selected as best, minus the percentage of times they are not. Ties between summaries do not affect this calculation, effectively being disregarded in the scoring process. This method quantifies the relative performance of each strategy while neutralizing instances where the quality difference is indiscernible to human evaluators. Appendix C provides detailed human instructions and annotation examples.

5. Results

Our goal is to evaluate the impact of different decoding methods and identify any emerging patterns. Given the complexity of our study, we focus on comparing relative efficiency and effectiveness.

Fig. 3 presents our findings through radar plots, offering an overview of the metric scores from our 2656 inference runs of encoder–decoder models. To identify *inter-dataset* patterns and facilitate observations across radar plots, we follow (Cao et al., 2022) and apply $[0, 1]$ min–max rescaling to *all* runs collectively.[9] These scores are aggregated by decoding strategy and dataset using average pooling. To enhance visual comprehension, we employ distinct colored areas for each decoding strategy. Additionally, we use distinct marks (circles and triangles) to differentiate between deterministic and stochastic decoding strategies. Within each radar chart, decoding strategies are layered based on their cumulative rescaled scores across all metrics; for metrics where lower values are preferable (i.e., density, perplexity, NID), we use the negative of the rescaled score. Layering is done in ascending order; thus, the most prominent color in each radar represents the best overall strategy. Precise rescaled metric scores are provided in tabulated form in Appendix D.

Fig. 4 provides a comprehensive visualization of the non-rescaled metric score distributions using boxplots. Each box offers a snapshot of key statistical information for the metric scores produced by a decoding strategy in a given metric–dataset combination. Specifically, a box spans the interquartile range, illustrating the spread of the middle 50 % of the data and providing a measure of score dispersion. The lower and upper whiskers represent the minimum and maximum values, respectively. The diamond indicates the mean score, while the in-box line represents the median, providing insight into the symmetry or skewness of the distribution. The closer the median line is to the center of the box, the more

[9] Rescaling has been implemented solely for presentation purposes in the context of inter-dataset evaluations. Original metric scores are maintained. For power analysis (Section 4.1), differences in means and standard deviations were calculated before rescaling.

Table 9
A summary of the metrics used in this study to assess the generated summaries. References occur in table are cited here (Lin, 2004) [78], (Zhang et al., 2020) [79], (Jelinek et al., 1977) [80], (Moro et al., 2023b) [81], (Devlin et al., 2019) [82], (Radford et al., 2019) [45], (Grusky et al., 2018) [83], (Xiao & Carenini, 2020) [84], (Weizhe et al., 2021) [85].

METRICS*,†		
ROUGE $[0,1],\uparrow$ [78]	**BERTScore** $[-1,1],\uparrow$ [79]	**Perplexity** $[0,\infty[,\downarrow$ [80]
Unigrams (r_1), bigrams (r_2), and longest common subsequence (r_{Lsum}) lexical overlaps (%) between the inferred and gold summaries, i.e., a proxy for informativeness and fluency. Inspired by [81], we measure an aggregated judgment: $\mathcal{R} = \frac{\text{avg}(r_1, r_2, r_{Lsum})}{1+\sigma_r^2}$ (2) where σ_r^2 is the variance of 0-1 normalized ROUGE F1 scores. $\mathcal{R}$ penalizes model results with discrepant r_1, r_2, r_{Lsum} values.	Semantic recall formalized as: $\frac{\sum_{y_i \in \boldsymbol{y}} \text{idf}(y_i) \max_{\hat{y}_j \in \hat{\boldsymbol{y}}} \mathbf{e}(y_i)^\intercal \mathbf{e}(y_j)}{\sum_{y_i \in \boldsymbol{y}} \text{idf}(y_i)}$ where $\mathbf{e}(\cdot)$ is a BERT [82] token embedding.	The naturalness of a summary w.r.t. the data seen by a model (GPT-2 [45] in this paper) during training: $ppl(\boldsymbol{y}) = \exp\left(\frac{1}{t}\sum_i^t \log p_\theta(y_i \vert \boldsymbol{y}_{<i})\right)$.
Coverage $[0,1],\uparrow$ [83]	**Density** $[0,\vert x\vert_c],\downarrow$ [83]	**Compression** $[0,\vert x\vert],\uparrow$ [83]
The percentage of summary words within the source text: $\frac{1}{\vert \boldsymbol{y}\vert}\sum_{f \in \mathcal{F}(\boldsymbol{x},\boldsymbol{y})} \vert f\vert$ where $\mathcal{F}$ is the set of all fragments, i.e., extractive character sequences. When low, it suggests a high chance for unsupported entities and facts.	The average length of the extractive fragments. It is formulated as: $\frac{1}{\vert \boldsymbol{y}\vert_c}\sum_{f \in \mathcal{F}(\boldsymbol{x},\boldsymbol{y})} \vert f\vert_c^2$ where $\Vert_c$ is the character length. When low, it suggests that most summary sentences are not verbatim extractions from the sources (abstractive).	The document-summary word ratio: $\frac{\vert \boldsymbol{x}\vert}{\vert \boldsymbol{y}\vert}$
Unique N-gram Ratio (UNR) $[0,1],\uparrow$ [84]	**Normalized Inverse of Diversity (NID)** $[0,1],\downarrow$ [84]	**BARTScore-F** $[-\infty,0],\uparrow$ [85]
The summary n-grams uniqueness: $\frac{\text{count}(\text{uniq\_n\_gram}(\boldsymbol{y}))}{\text{count}(\text{n\_gram}(\boldsymbol{y}))}$ where we take n from $[1,3]$ and divide the average by variance.	It reckons redundancy by inverting the entropy of summary unigrams and applying length normalization: $1 - \frac{\text{entropy}(\boldsymbol{y})}{\log(\vert \boldsymbol{y}\vert)}$	The weights θ of a pretrained Bart model [43] are used to estimate how likely hypothesis and reference are reciprocal paraphrases (i.e., probability of generating one giving the other): $\sum_{t=1}^{\vert \boldsymbol{y}\vert} \log p(y_t \vert \boldsymbol{y}_{<t}, \boldsymbol{x}, \theta_B)$ We measure faithfulness by mapping sources and predicted summaries to $\boldsymbol{x}$ and $\boldsymbol{y}$.
Carburacy [81] $[0,1],\uparrow$		
Carbon-aware accuracy measure modeling both the AS model effectiveness and eco-sustainability: $\Upsilon = \frac{(\exp^{\mathcal{R}_\alpha})}{1+C\cdot\beta}$, where $\mathcal{R}$ is defined in Equation 2, C is the kg of CO_2 emissions produced by the model to process a single instance x at inference time, α and β are trade-off hyperparameters.		

* Blue text indicates the bound value and the general reading key (i.e., ↑ = higher is better, ↓ = lower is better).
† ● = reference-based (gold-summary dependence), ● = reference-free

Table 10
Pooled standard deviations for different metrics per dataset.

Dataset	$\mathcal{R}$	BERTScore	Perplexity	Coverage	Density	Compression	UNR	NID	BARTScore	Carburacy
XSum	3.2728	15.0274	5.3514	0.1551	0.4401	0.9841	0.0361	0.0626	2.5702	4.4817
Cnn/Dm	5.4050	17.5126	5.7900	0.0051	2.7847	0.0464	0.0184	0.0397	3.2044	5.6668
PubMed	6.9028	15.4085	7.8158	0.2595	10.8964	1.0158	0.0451	0.0481	2.8203	6.7917
arXiv	5.8003	12.6680	15.4260	0.2116	7.8669	9.5349	0.0375	0.0495	2.4604	5.5582
Multi-News	5.4603	14.9108	1.4984	0.1810	1.5428	0.6291	0.0356	0.0478	2.6767	5.4972
Multi-LexSum	1.6939	2.5097	13.9422	0.0050	0.0500	33.7744	0.0262	0.0111	0.2524	1.4835
AVG	4.4220	13.0062	8.6379	0.1362	3.6138	7.3308	0.0333	0.0434	2.3301	4.9132

symmetrical the distribution. This visualization enables researchers and practitioners to make informed decisions about strategy selection based not only on average performance, but also on consistency and potential extremes in different scenarios. By comparing boxplots across different metrics and datasets, the figure helps identify recurring patterns and dataset-specific trends. Moreover, Table 10 reports the pooled standard deviations for each metric per dataset.

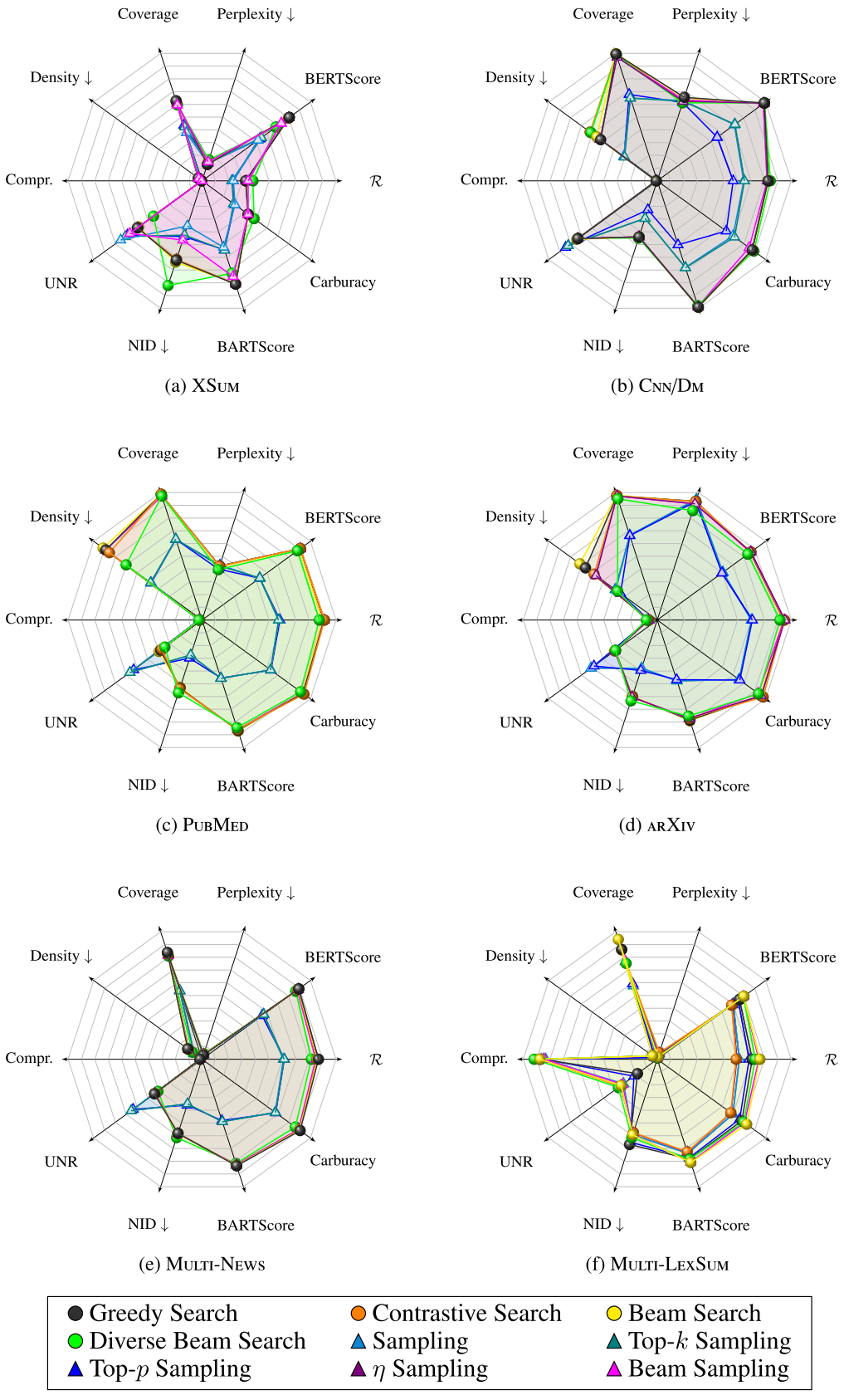
To provide a more nuanced view of decoding strategy performance, Fig. 5 and Fig. 6 present per-dataset rankings for $\mathcal{R}$ and Carburacy effectiveness. All inference runs for each dataset are sorted in descending order by their metric scores. Each dot represents an individual run, colored according to its decoding strategy. The x-axis shows the percentile ranking, while the y-axis displays the actual metric score. To facilitate quick interpretation of the results, each ranking includes a pie chart showing the percentage distribution of decoding strategies based on the number of runs that produced metric scores in the top 25 %. However, decoding strategies can differ significantly in the size of their hyperparameter search spaces, leading to varying numbers of experimental runs and probabilities of occurrence. For example, for a given dataset, there are only 16 runs for Greedy Search compared to 720 for Diverse Beam Search (see Table 3). To ensure a fairer comparison and account for discrepancies in the total run counts, we present an additional normalized pie chart showing the proportion of each strategy's runs that fall within the top quartile. These figures help researchers and practitioners identify which strategies are most likely to yield top results in terms of lexical overlap and efficiency across various configurations and datasets, which is crucial for real-world applications.

Fig. 7 summarizes the time required to generate a single summary in each dataset, depending on the selected decoding strategy. Efficiency distributions are shown using boxplots, following the same visual notation used in Fig. 4.

In the following subsections, we analyze the results, structuring our discussion around the five research questions that motivated our study.

5.1. Quantitative findings

Please note that the findings in this subsection are based on automatic metrics that serve as proxies for the quality dimensions of interest. While these metrics provide valuable insights, they should be interpreted as indirect measures of the complex qualities being examined. To



(a) XSum (b) Cnn/Dm

(c) PubMed (d) arXiv

(e) Multi-News (f) Multi-LexSum

Fig. 3. Metric-based comparison between artificial and gold summaries in the examined datasets (min–max rescaling across all runs). The dominant color in each radar denotes the best overall strategy (• = deterministic, ▲ = stochastic).

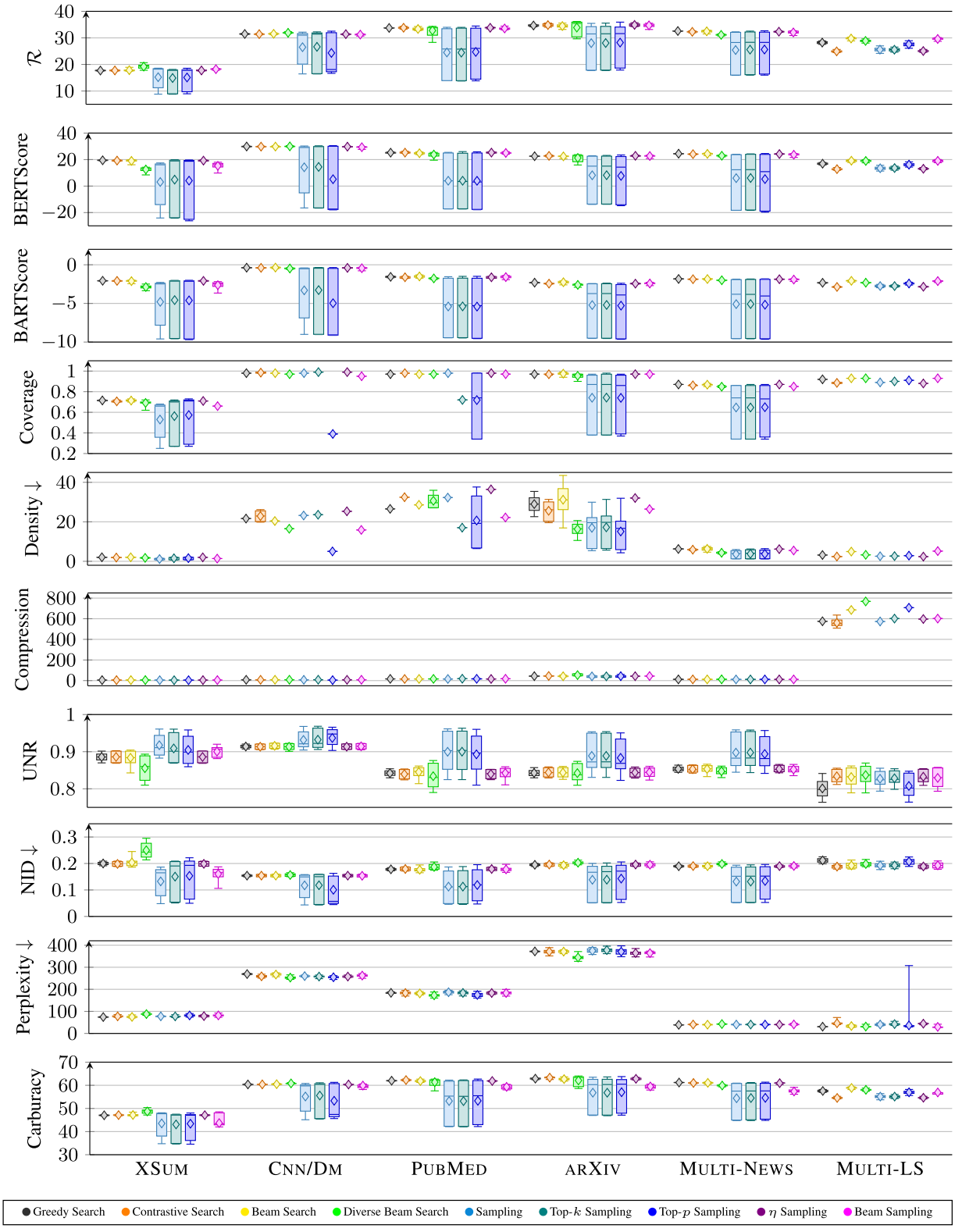


Fig. 4. Graphical metric score distributions grouped by datasets and strategies.

support our claims, we conduct paired t-tests comparing specific decoding strategies for key metrics and datasets.

RQ1 **Is there an absolute best decoding method for AS?**

There is no one-size-fits-all strategy for AS.

A glance at Fig. 3 reveals that there is no single superior decoding method when considering the average score across the evaluation metrics. This is evident from the variation in dominant colors observed across the different radar charts, which reflects the datasets. For example, Diverse Beam Search (green) performs best in PUBMED and ARXIV, while Beam Sampling (fuchsia) excels in XSUM.

Looking at the non-pooled results in Fig. 5 and considering the top-25 % ranking in terms of $\mathcal{R}$ (the standard AS metric), we see a clear

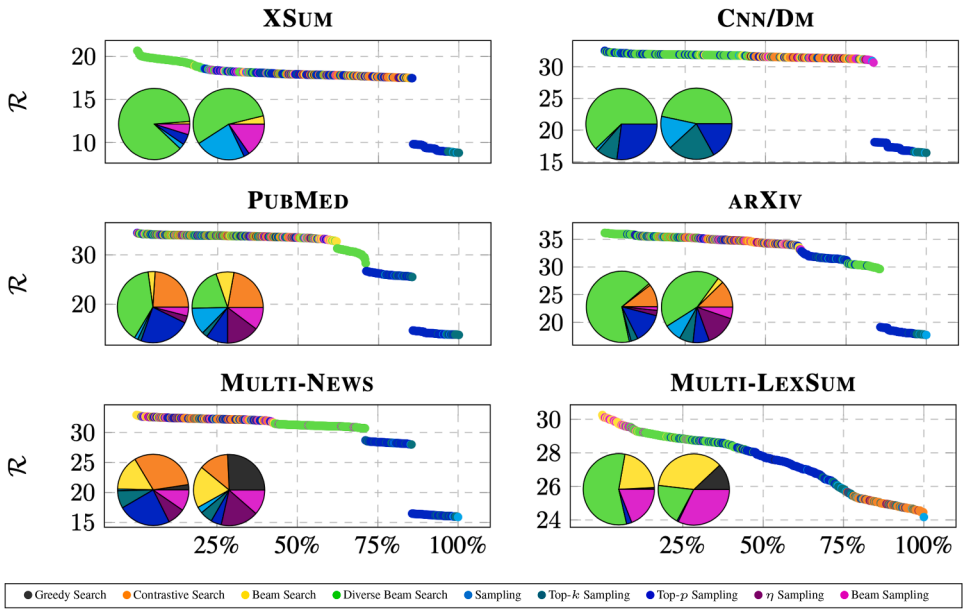


Fig. 5. Per-dataset decoding strategy ranking according to $\mathcal{R}$. The left pie chart illustrates the distribution of decoding strategies within the top 25 %. The right pie chart shows the normalized occurrences based on their total number of runs.

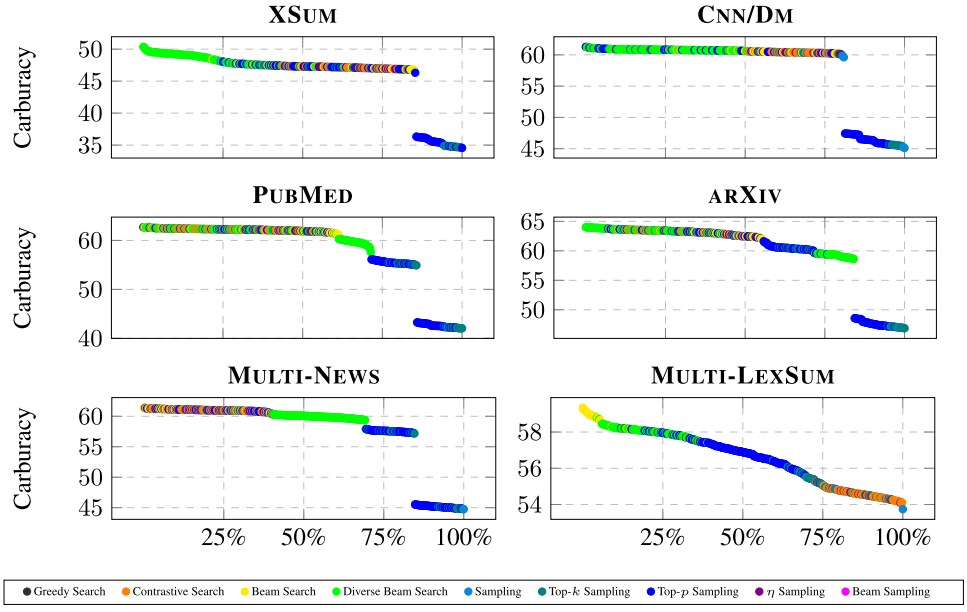


Fig. 6. Per-dataset decoding strategy ranking according to Carburacy.

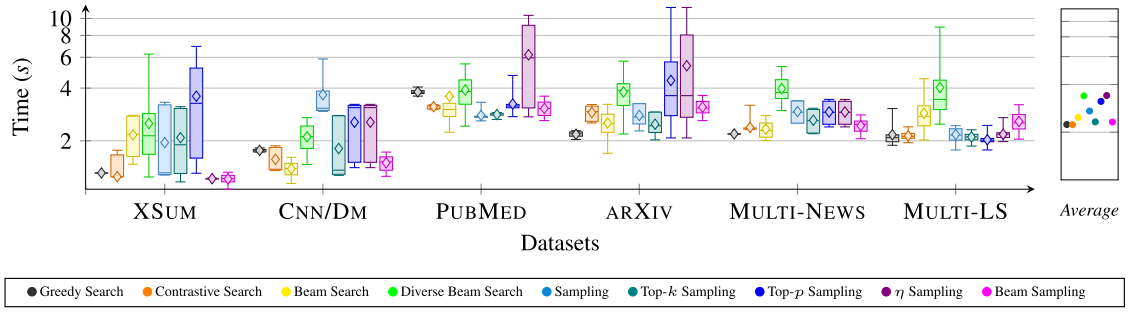


Fig. 7. Decoding time complexity (in seconds) for summarizing a single instance of sampled test sets. For each strategy, the average time across all samples is also reported.

prominence of deterministic methods. Specifically, deterministic strategies account for 87.95 % on XSUM, 62.16 % on CNN/DM, 66.4 % on PUBMED, 78.4 % on ARXIV, 49.6 % on MULTI-NEWS, and 78.57 % on MULTI-LEXSUM (cf. pie charts under the per-dataset rankings). Although less consistent, well-calibrated stochastic methods perform well and can be highly competitive with deterministic ones. Surprisingly, Top-p Sampling shows a notable presence in the highest-performing quartile across several datasets: 23.2 % for PUBMED, 24 % for MULTI-NEWS, and 27 % for CNN/DM. Similarly, Beam Sampling is particularly effective in MULTI-NEWS (9.6 %) and MULTI-LEXSUM (19 %).

Diverse Beam Search, when adequately tuned, proves its merit regardless of the AS type. With the exception of MULTI-NEWS, it is the leading strategy in the top $\mathcal{R}$ quartile of each dataset, with frequency rates ranging from 39.2 % to 86.75 %.

RQ2 **Are decoding methods sensitive to AS type?**

The effectiveness of various decoding strategies shows distinct trends across datasets, which appear to be influenced by the AS type.

The preference for a decoding strategy varies between datasets but tends to be more uniform within the same AS family. This pattern is evident in the average scores across all metrics (see the colored areas in Fig. 3). SDS favors Greedy Search and Beam Sampling. LDS highlights the effectiveness of Diverse Beam Search. MDS tasks benefit most from Beam Search. A closer inspection of the top-performing decoding runs, specifically ranked by the $\mathcal{R}$ metric (Fig. 5), further supports our findings. Diverse Beam Search and well-tuned Top-p Sampling achieve the highest scores in SDS. In particular, Diverse Beam Search demonstrates statistically significant superior performance in several key metrics for both XSUM and CNN/DM datasets. For XSUM, it shows significant improvements in $\mathcal{R}$ ($p = 3.94e^{-15}$), BARTScore ($p = 0.0086$), perplexity ($p = 9.99e^{-49}$), and NID ($p = 1.09e^{-31}$). For CNN/DM, significant improvements are observed in $\mathcal{R}$ ($p = 8.55e^{-9}$), BERTScore ($p = 9.82e^{-8}$), BARTScore ($p = 3.13e^{-7}$), and NID ($p = 1.64e^{-8}$). These results were obtained by performing one-tailed t-tests comparing Diverse Beam Search against all other methods for each metric in each dataset, with the null hypothesis that Diverse Beam Search does not perform better.

The effectiveness of Constrastive Search increases markedly in LDS, achieving high ranks in both PUBMED and ARXIV; on average, its frequency in the top quartile moves from 0 % (SDS) to 17 % (LDS).

Again, Beam Search and Beam Sampling prove to be the most stable and reliable options for MDS. Beam Search and Beam Sampling demonstrate significantly lower variance across most metrics in both MULTI-NEWS and MULTI-LEXSUM datasets, indicating higher stability and reliability. For MULTI-NEWS, all metrics show extremely significant results ($p < 1e^{-16}$ for all metrics, $p = 0.019$ for perplexity). For MULTI-LEXSUM, highly significant results are observed for $\mathcal{R}$, BERTScore, BARTScore, and perplexity (all $p < 1e^{-16}$), with UNR also showing significance ($p = 0.042$). These results were obtained by performing F-tests comparing the variance of combined Beam Search and Beam Sampling methods against all other methods for each metric in each dataset, with lower variance indicating higher stability.

These results raise warnings about claims made in prior publications with task-agnostic or single-AS-type settings (Su & Collier, 2023).

We found `temperature=0.8` in Top-k Sampling and Top-p Sampling caused the negative metric score clusters in XSUM, CNN/DM, PUBMED, ARXIV, and MULTI-NEWS (cf. isolated points in the ranking tails). The worst runs in MULTI-LEXSUM are due to η Sampling and Contrastive Search. These patterns are consistent in benchmarks belonging to the same AS type, such as PUBMED and ARXIV. A deviation is observed in MULTI-LEXSUM, which involves extremely long inputs (averaging over 88K words, as shown in Table 6). In this case, truncation narrows the gap between strategies, negatively affecting Contrastive Search. The hypothesized reason is the α>0 hyperparameter, which assigns less weight to model confidence—a critical factor for handling uncertainty.

RQ3 **To what extent does proper decoding affect LM metrics?**

Changing the decoding strategy is not just about decimals.

The choice of decoding heuristic can substantially affect LM scores within a given benchmark dataset. This choice can result in variations of up to 20 $\mathcal{R}$ points in PUBMED, 9 BARTScore points in CNN/DM, and 278 Perplexity points in MULTI-LEXSUM. Furthermore, depending on the hyperparameters chosen, Sampling, Top-k Sampling, and Top-p Sampling display the highest degree of result variability (cf. the box height in Fig. 4). For example, in ARXIV, their summaries can vary by up to 17 $\mathcal{R}$ points (syntax), 7.5 BARTScore points (factuality), and 60 Coverage points (shifting from abstractive to highly extractive, thus impacting all other metrics). In contrast, we observe a homogeneous pattern in MULTI-LEXSUM due to truncation.

RQ4 **Which decoding method provides the best effectiveness–efficiency trade-off?**

Deterministic methods generally provide the most balanced option in terms of output quality and computational efficiency.

Fig. 7 shows that Greedy Search and Contrastive Search are the fastest strategies, with an average of 2.1 seconds per instance. Diverse Beam Search, Top-p Sampling, and η Sampling are the slowest, with an average of 3.7 seconds. The latency of Diverse Beam Search, Sampling, Top-k Sampling, Top-p Sampling, and η Sampling greatly depends on the hyperparameters and the specific dataset. The widest range of processing times for these strategies within a single dataset are: Diverse Beam Search (±5.9 sec on MULTI-LEXSUM), Sampling (±2.9 sec on CNN/DM), Top-k Sampling (±2 sec on XSUM), Top-p Sampling (±9.5 sec on ARXIV), η Sampling (±9.5 sec on ARXIV). Beyond text length, Diverse Beam Search shows increased latency with higher beam sizes and group numbers. Sampling methods such as Top-k, Top-p, and η have longer processing times with more diverse token selections (higher k, p, and temperature values, or lower cut-offs). Beam Search latency increases with larger beam sizes, while Contrastive Search slows down with higher α values, which intensify the calculation of the degeneration penalty.

Regarding CO_2 emissions at inference time, the best average $\mathcal{R}$–efficiency trade-off, as measured by Carburacy, varies across datasets. Diverse Beam Search performs best for XSUM and CNN/DM (cf. the green line in Fig. 3 for Carburacy), with an average p-value of 0.012 and 0.032 compared to all other strategies. Contrastive Search excels for PUBMED and ARXIV, with an average p-value of 0.010 and 0.016. Beam Search leads for MULTI-LEXSUM (p 0.0048), while Greedy decoding performs best for MULTI-NEWS (p 0.0075). These p-values were obtained using independent t-tests comparing the best-performing strategy against each other strategy for each dataset. The detailed Carburacy ranking in Fig. 6 supports these results.

RQ5 **Which hyperparameter values best suit a particular AS quality attribute?**

Not all quality attributes are easy to control at decoding time.

High beam size, high `no_repeat_ngram_size`, and low diversity penalty promote factuality and semantic consistency. For $\mathcal{R}$, it is strongly recommended to maintain a large beam size, while using a temperature of 0.8 in Top-k Sampling and Top-p Sampling often results in clusters of degeneration. Interestingly, no strategy or hyperparameter consistently favors the naturalness of text in a predictable way. However, we observe a strong positive correlation between Perplexity and Density scores (0.73 Pearson coefficient). Appendix E provides a thorough examination of how metrics respond to fine-grained variations in hyperparameters. Appendix F reports the best hyperparameter values across all datasets and includes a per-dataset evaluation.

Among the deterministic methods, Diverse Beam Search is characterized by high redundancy. MDS consistently yields a lower perplexity than the other task families, reflecting the difficulty of current models in achieving high naturalness. The extractive nature of a summary in LDS scenarios can vary greatly depending on the strategy implemented. As expected, the length of the target is strongly correlated with the $\mathcal{R}$ score, implying a higher probability of generating n-grams of overlap with longer references, especially in LDS scenarios.

Given the widespread use of ROUGE as a standard metric in AS and NLG research (Grusky, 2023), we present the optimal hyperparameter

Table 11
Hyperparameters for the top three $\mathcal{R}$ decoding strategies per dataset.

Dataset	no_rep	num_beams	div_penalty	temperature	penalty_alpha	top_k	top_p	cutoff
XSUM								
1. Contrastive Search	2	-	-	-	1.0	20	-	-
2. Top-k Sampling	2	-	-	0.9	-	50	-	-
3. Top-p Sampling	2	-	-	0.9	-	50	0.6	-
CNN/DM								
1. Diverse Beam Search	3	5	0.2	-	-	-	-	-
2. Top-k Sampling	3	-	-	0.9	-	50	-	-
3. Top-p Sampling	3	-	-	0.9	-	-	0.4	-
PUBMED								
1. Contrastive Search	5	-	-	-	0.8	30	-	-
2. Top-p Sampling	5	-	-	1.0	-	50	0.8	-
3. Beam Sampling	5	3	-	-	-	-	-	-
ARXIV								
1. Diverse Beam Search	5	10	0.6	-	-	-	-	-
2. Top-p Sampling	5	-	-	1.0	-	20	0.6	-
3. η Sampling	5	-	-	-	-	-	-	0.0003
MULTI-NEWS								
1. Contrastive Search	5	-	-	-	0.2	30	-	-
2. Beam Search	4	10	-	-	-	-	-	-
3. Beam Sampling	5	9	-	-	-	-	-	-
MULTI-LEXSUM								
1. Beam Search	3	8	-	-	-	-	-	-
2. Diverse Beam Search	3	7	0.2	-	-	-	-	-
3. Beam Sampling	3	8	-	-	-	-	-	-

Table 12
Average FLOPs computational performance of evaluated decoding strategies.

Greedy	Contrastive	Beam Search	DBS	Sampling	Top-k	Top-p	η Sampling	Beam Sampling
$1.7e^{17}$	$1.7e^{17}$	$1.9e^{17}$	$2.6e^{17}$	$2.1e^{17}$	$1.8e^{17}$	$2.4e^{17}$	$2.6e^{17}$	$1.8e^{17}$

configurations for the three decoding strategies that achieved the highest $\mathcal{R}$ scores in our 2656 experimental runs of encoder–decoder models (Table 11). This information serves as valuable reference points for researchers and practitioners in the field.

Extra findings.

- *Redundancy is ubiquitous, mainly in MDS:* Our scores contradict (Meister et al., 2022), who refer to redundancy as a rare phenomenon in AS. We observe a tendency for recurring tokens as the input size increases, peaking with MULTI-LEXSUM (-60.97 % UNR avg.). As expected, short summaries (i.e., in XSUM and CNN/DM) are less redundant than longer ones (+46.59 % UNR avg.) (Xiao & Carenini, 2020).
- *Stochastic vs. deterministic:* The addition of randomness increases the chance of unconventional summaries and contradictions. In five out of six datasets, Sampling, Top-k Sampling, and Top-p Sampling exhibit greater sample variance compared to all other strategies. As expected, they have fewer repetitions (+27.63 % UNR avg.) (Fan et al., 2018; Holtzman et al., 2020) but tend more to factual flaws (-40.58 % BARTScore avg.) (Basu et al., 2021; Su et al., 2022). Notably, η Sampling and Beam Sampling stand out in the stochastic sphere, suffering less from hallucinations (+40.23 % BARTScore avg.). Together with deterministic methods, they produce the highest $\mathcal{R}$ (+29.65 % avg.) and BERTScore (+35.34 % avg.).

RQ6 **Do our findings from encoder–decoder models transfer to recent decoder-only LLMs?**

We tested both LLMs—LLAMA-3.1-8B and QWEN2.5-7B—on all six datasets using a subset of decoding strategies: Greedy, Sampling, Top-k, and Top-p, evaluated across all combinations of selected hyperparameters (see Table 3 for the total number of runs). Our choices were guided by practical considerations: we used vLLM, a widely adopted tool for efficient LLM inference, and thus limited our experiments to the decoding methods it supports, which reflect the most commonly used strategies in practice. Similarly, we did not consider the `no_repeat_ngram_size` hyperparameter, as it is not supported by vLLM. Beam Search was excluded after experimentation due to its tendency to produce empty outputs. See Fig. 10 for the prompt used. See Table 13 for the results.

Consistency with previous findings.

Similar to encoder-decoder models, greedy decoding, even without training, remains one of the most preferable and robust strategies overall. On average, it consistently outperforms stochastic methods across most datasets, especially in SDS. It also yields low perplexity, confirming its tendency to produce fluent, coherent summaries.

Among sampling-based methods, top-p again performs better than standard sampling. As in prior results, it shows more stable performance across hyperparameter variations, as evidenced by consistently lower uncertainty (see blue highlights in Table 3).

Key deviations.

Differently from encoder–decoder models—where deterministic decoding strategies emerged as the most reliable choice for practitioners unable to conduct broad hyperparameter sweeps—the landscape shifts when considering decoder-only LLMs. Ranking decoding runs by dataset based on average $\mathcal{R}$, BERTScore, and BARTScore reveals that greedy decoding and top-p sampling consistently occupy the top two positions. This trend holds irrespective of the AS type, suggesting that both strategies are robust defaults. Given the greater stability in metric scores as hyperparameters vary, top-p sampling stands out as a more recommendable approach than it is for encoder–decoder models. Consequently, we advise deprioritizing sampling and top-k sampling in favor of greedy and top-p, effectively narrowing the search space. As for hyperparameter values, temperature 0.8 appears most frequently among best-performing settings across datasets—a finding that again contrasts with recommendations for encoder–decoder setups.

One major difference concerns perplexity. While it was previously observed to be stable across decoding strategies and hyperparameters, decoder-only LLMs show significantly higher variance under stochastic decoding—particularly in LDS and MDS settings. This suggests that fluency in decoder-only LLMs is more sensitive to decoding settings.

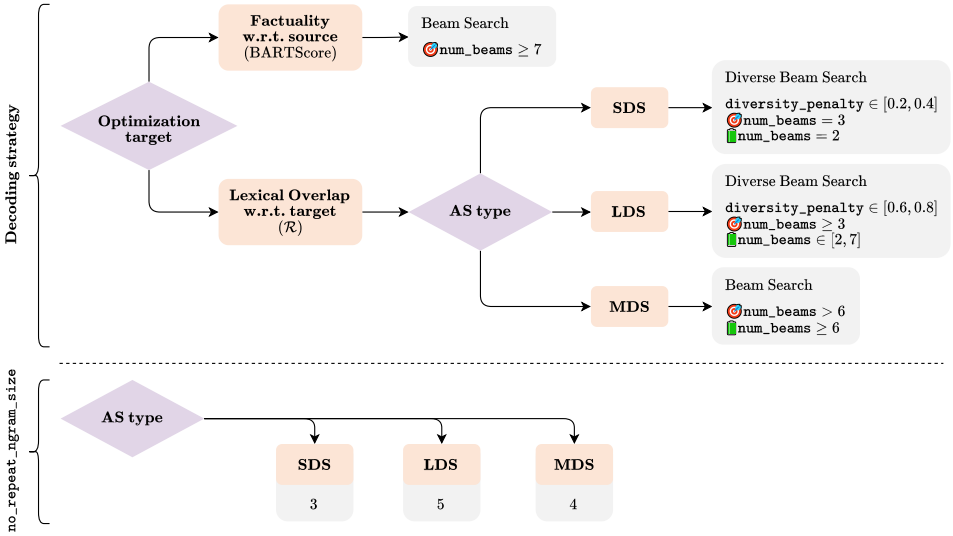


Fig. 8. Visual guideline for the recommended decoding strategies and hyperparameter settings depending on the use case scenario. Optimization target = "What are you interested in?"; AS type = "What is your Abstractive Summarization scenario?". The hit symbol denotes hyperparameter value recommendations for achieving maximum effectiveness on the proxy metric, while the battery symbol signifies the optimal effectiveness–efficiency trade-off as proposed by (Moro et al., 2023b).

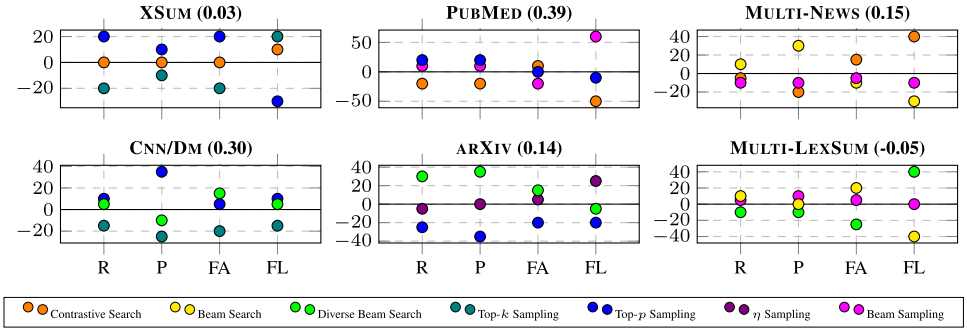


Fig. 9. Win - Lose (%) human evaluation results on four quality dimensions: Recall (R), Precision (P), Faithfulness (FA), and Fluency (FL). The average Kendall's τ coefficients among all inter-annotator agreements are given in brackets.

When comparing LLMs with encoder–decoder models using the reference-free BARTScore metric—which evaluates similarity to the input document rather than a gold summary—we do not observe a clear advantage for LLMs. This suggests that, despite their scale, decoder-only models do not consistently produce summaries that are better aligned with the source content under this type of evaluation.

While encoder–decoder models showed more consistent patterns across architectures, decoder-only models diverge more. For instance, LLAMA-3.1 consistently outperforms QWEN2.5, but the performance gap depends on the decoding strategy used—narrower under greedy decoding, wider under sampling. This suggests that the reasoning-enhanced design of QWEN2.5 does not provide a clear benefit in the summarization setting, where structured extraction and compression may outweigh the need for complex reasoning.

Both models consistently fail to handle the Multi-LexSum dataset, regardless of decoding strategy or hyperparameter choice. Outputs are often incomplete or degenerate (e.g., one-word summaries), suggesting that current open-source decoder-only LLMs are not well equipped to handle extremely long MDS tasks out of the box, and likely require fine-tuning or task-specific adaptation.

5.2. Qualitative findings

The annotation process took approximately six hours per judge. The results are presented in Fig. 9. The average Kendall's τ of 0.33, calculated between two annotators across all pair selection results, reflects the high competitiveness among the top three decoding strategies when correctly tuned, often leading to subjective summary preferences.

Top-p Sampling excels in AS with concise inputs. Although not reflected by the $\mathcal{R}$ metric on XSUM, Top-p Sampling gains 75 % human preference in SDS. However, its effectiveness diminishes as input length increases, favoring deterministic alternatives. Consistent with this trend, Beam Search prevails in MDS.

Fluency negatively correlates with Recall, Precision, and Factuality. Fluency opposes Recall, Precision, and Factuality more than 66 % of the time. Additionally, Fluency preferences depend on the dataset and do not always favor stochastic strategies.

5.3. Additional results

Experiment Tracking. The cumulative processing time of the GPU for generating all summaries amounts to approximately 65 days. Computing all metric scores requires about 16 days: 12 hours for $\mathcal{R}$, 18 hours for BERTScore, 3.6 hours for Perplexity, 198 hours for Coverage, Density, and Compression, 38 minutes for UNR and NID, 75 hours for BARTScore, and 10 seconds for Carburacy. Table 12 reports the FLOPs computational performance of the decoding strategies as an additional measure of processing power and efficiency.

Visual Guide. Wrapping up our core findings—based on average scores, ranking positions, and hyperparameter sensitivity—we identify a practical and easy-to-follow guideline for encoder–decoder models (Fig. 8),

System:
You are a helpful assistant that summarizes documents.

User:
Summarize the following document directly, without prefacing the response: `{document}`

Fig. 10. Prompt used to guide LLMs on the summarization task.

Table 13
Average decoder-only LLM scores categorized by dataset, metric, and decoding strategy. Best scores are shown in bold. Blue highlights indicate lower uncertainty (max deviation from the mean across runs).

	Strategy	$\mathcal{R}$	BERTScore	Perplexity ↓	Coverage	Density ↓	Compression	UNR	NID ↓	BARTScore
					XSum					
LLaMa-3.1	Greedy Search	12.93	8.69	**27.04**	**0.86**	3.56	4.01	**0.89**	**0.17**	**-1.62**
	Sampling	12.86 ± 0.04	8.70 ± 0.07	29.65 ± 1.06	0.84 ± 0.01	**3.12** ± 0.14	**4.10** ± 0.06	**0.89** ± 0.0	**0.17** ± 0.0	-1.80 ± 0.06
	Top-k Sampling	12.84 ± 0.04	8.71 ± 0.09	29.47 ± 0.85	0.84 ± 0.01	**3.12** ± 0.14	4.08 ± 0.05	**0.89** ± 0.0	**0.17** ± 0.0	-1.80 ± 0.05
	Top-p Sampling	**12.97** ± 0.06	**8.74** ± 0.13	27.54 ± 0.83	0.85 ± 0.01	3.49 ± 0.14	4.05 ± 0.04	**0.89** ± 0.0	**0.17** ± 0.0	-1.65 ± 0.05
Qwen2.5	Greedy Search	8.94	4.93	**22.56**	**0.81**	9.58	**2.55**	0.77	0.21	**-1.82**
	Sampling	8.76 ± 0.26	5.39 ± 0.46	37.31 ± 15.15	0.72 ± 0.04	**5.44** ± 1.72	2.54 ± 0.15	**0.84** ± 0.02	**0.17** ± 0.01	-2.72 ± 0.46
	Top-k Sampling	8.91 ± 0.16	**5.55** ± 0.26	28.03 ± 2.79	0.75 ± 0.03	6.84 ± 1.47	2.48 ± 0.05	0.83 ± 0.01	0.18 ± 0.0	-2.44 ± 0.23
	Top-p Sampling	**9.03** ± 0.27	5.41 ± 0.42	23.93 ± 3.13	**0.81** ± 0.04	9.47 ± 2.11	2.53 ± 0.09	0.80 ± 0.02	0.19 ± 0.01	-1.96 ± 0.35
					CNN/DM					
LLaMA-3.1	Greedy Search	**30.16**	**20.69**	**22.23**	**0.89**	3.55	8.64	0.89	**0.17**	**-1.69**
	Sampling	29.25 ± 0.28	19.87 ± 0.17	24.31 ± 0.71	0.87 ± 0.01	**3.12** ± 0.09	8.74 ± 0.04	**0.90** ± 0.0	**0.17** ± 0.0	-1.85 ± 0.05
	Top-k Sampling	29.27 ± 0.31	19.92 ± 0.17	24.19 ± 0.66	0.87 ± 0.01	**3.12** ± 0.09	**8.75** ± 0.04	**0.90** ± 0.0	**0.17** ± 0.0	-1.84 ± 0.05
	Top-p Sampling	29.92 ± 0.24	20.48 ± 0.13	22.61 ± 0.47	**0.89** ± 0.01	3.49 ± 0.09	8.71 ± 0.02	0.89 ± 0.0	**0.17** ± 0.0	-1.71 ± 0.04
Qwen2.5	Greedy Search	**29.58**	**19.68**	**24.34**	**0.90**	6.82	8.61	0.89	**0.17**	**-1.70**
	Sampling	26.65 ± 1.40	17.30 ± 1.52	33.79 ± 12.55	0.82 ± 0.03	**3.16** ± 0.50	**8.86** ± 0.51	**0.90** ± 0.0	**0.17** ± 0.0	-2.36 ± 0.30
	Top-k Sampling	27.04 ± 0.98	17.87 ± 0.87	28.81 ± 4.29	0.83 ± 0.02	3.27 ± 0.42	8.61 ± 0.05	**0.90** ± 0.0	**0.17** ± 0.0	-2.26 ± 0.17
	Top-p Sampling	29.04 ± 0.61	19.35 ± 0.29	24.51 ± 0.89	0.88 ± 0.02	5.40 ± 1.07	8.58 ± 0.04	0.89 ± 0.0	**0.17** ± 0.0	-1.83 ± 0.15
					PUBMED					
LLaMA-3.1	Greedy Search	**33.02**	**22.63**	**17.77**	**0.91**	4.25	**36.52**	0.79	**0.20**	**-2.15**
	Sampling	32.45 ± 0.28	21.59 ± 0.33	18.89 ± 0.61	0.90 ± 0.01	**3.68** ± 0.15	36.03 ± 0.07	**0.80** ± 0.0	**0.20** ± 0.0	-2.28 ± 0.04
	Top-k Sampling	32.76 ± 0.29	22.11 ± 0.41	17.89 ± 0.61	**0.91** ± 0.0	4.21 ± 0.12	36.06 ± 0.57	0.79 ± 0.01	**0.20** ± 0.0	-2.17 ± 0.05
	Top-p Sampling	32.49 ± 0.25	21.66 ± 0.31	18.85 ± 0.81	0.90 ± 0.01	3.69 ± 0.14	36.02 ± 0.15	**0.80** ± 0.0	**0.20** ± 0.0	-2.28 ± 0.04
Qwen2.5	Greedy Search	**30.57**	**19.77**	**37.39**	**0.89**	11.24	**188.57**	0.82	0.21	**-2.24**
	Sampling	28.38 ± 1.03	17.38 ± 1.35	69.98 ± 16.72	0.83 ± 0.03	**3.98** ± 0.77	187.49 ± 0.70	**0.84** ± 0.01	**0.19** ± 0.01	-2.79 ± 0.23
	Top-k Sampling	28.76 ± 0.72	18.05 ± 0.85	48.16 ± 17.35	0.84 ± 0.02	4.26 ± 0.69	187.10 ± 0.23	**0.84** ± 0.01	**0.19** ± 0.0	-2.68 ± 0.16
	Top-p Sampling	30.51 ± 0.38	19.63 ± 0.73	46.50 ± 26.79	0.87 ± 0.02	7.35 ± 2.70	187.33 ± 0.63	0.82 ± 0.0	0.20 ± 0.01	-2.35 ± 0.18
					ARXIV					
LLaMA-3.1	Greedy Search	**27.20**	**17.38**	32.97	**0.89**	3.70	116.19	0.79	**0.21**	**-2.66**
	Sampling	26.10 ± 0.31	15.65 ± 0.73	31.16 ± 0.58	0.86 ± 0.01	**3.19** ± 0.15	**119.86** ± 2.63	**0.81** ± 0.0	**0.21** ± 0.0	-2.95 ± 0.07
	Top-k Sampling	26.29 ± 0.41	15.96 ± 1.00	**30.75** ± 0.72	0.87 ± 0.02	3.23 ± 0.20	117.53 ± 0.95	**0.81** ± 0.0	**0.21** ± 0.0	-2.93 ± 0.05
	Top-p Sampling	26.29 ± 0.41	15.96 ± 1.00	**30.75** ± 0.72	0.87 ± 0.02	3.23 ± 0.20	117.53 ± 0.95	**0.81** ± 0.0	**0.21** ± 0.0	-2.93 ± 0.05
Qwen2.5	Greedy Search	19.96	9.01	**132.54**	0.76	4.31	**1309.20**	0.83	0.38	**-3.79**
	Sampling	18.75 ± 0.66	6.54 ± 0.46	142.51 ± 142.50	**0.82** ± 0.02	**3.03** ± 0.63	1213.56 ± 0.78	**0.87** ± 0.01	0.39 ± 0.04	-4.13 ± 0.26
	Top-k Sampling	19.11 ± 0.61	7.41 ± 1.37	151.53 ± 124.27	0.81 ± 0.06	3.10 ± 0.51	1213.21 ± 0.90	**0.87** ± 0.01	0.37 ± 0.07	-4.06 ± 0.18
	Top-p Sampling	**20.06** ± 0.37	**9.11** ± 1.93	206.64 ± 172.32	0.79 ± 0.06	4.27 ± 0.49	1212.93 ± 2.63	0.86 ± 0.0	**0.31** ± 0.12	-3.81 ± 0.06
					MULTI-NEWS					
LLaMA-3.1	Greedy Search	27.50	16.87	**59.33**	0.90	3.95	121.29	0.85	**0.18**	**-1.77**
	Sampling	27.29 ± 0.16	16.76 ± 0.15	104.84 ± 15.77	0.89 ± 0.01	**3.48** ± 0.18	**131.22** ± 0.31	**0.86** ± 0.0	**0.18** ± 0.0	-1.92 ± 0.06
	Top-k Sampling	27.34 ± 0.16	16.85 ± 0.12	142.59 ± 167.35	0.89 ± 0.01	3.49 ± 0.18	121.42 ± 0.32	**0.86** ± 0.0	0.19 ± 0.0	-1.91 ± 0.05
	Top-p Sampling	**27.55** ± 0.34	**17.00** ± 0.41	138.11 ± 137.03	**0.91** ± 0.01	3.99 ± 0.30	121.84 ± 9.21	0.85 ± 0.0	0.19 ± 0.0	-1.78 ± 0.05
Qwen2.5	Greedy Search	**27.99**	**15.24**	37.19	**0.90**	18.19	100.01	0.83	0.20	**-1.88**
	Sampling	25.69 ± 1.10	12.23 ± 1.98	54.50 ± 49.06	0.81 ± 0.04	**5.14** ± 1.84	**101.63** ± 4.06	**0.85** ± 0.01	**0.18** ± 0.01	-2.83 ± 0.47
	Top-k Sampling	26.34 ± 0.77	13.48 ± 0.92	44.57 ± 51.55	0.83 ± 0.02	6.22 ± 2.18	99.76 ± 0.42	**0.85** ± 0.01	0.19 ± 0.01	-2.58 ± 0.28
	Top-p Sampling	27.61 ± 0.72	15.05 ± 0.98	**32.66** ± 14.40	0.88 ± 0.03	9.45 ± 3.28	100.26 ± 0.45	0.84 ± 0.01	0.19 ± 0.01	-2.11 ± 0.26
					MULTI-LEXSUM					
LLaMA-3.1	Greedy Search	**3.71**	**-13.15**	1695.82	**0.94**	1.98	32251.22	**0.81**	/	**-4.90**
	Sampling	2.99 ± 0.15	-15.89 ± 0.90	1741.41 ± 994.52	0.75 ± 0.06	**1.23** ± 0.10	32988.66 ± 141.63	**0.81** ± 0.0	/	-5.44 ± 0.11
	Top-k Sampling	2.97 ± 0.15	-15.89 ± 0.94	1839.48 ± 1036.16	0.75 ± 0.06	1.24 ± 0.10	32483.49 ± 475.09	**0.81** ± 0.0	/	-5.42 ± 0.13
	Top-p Sampling	3.39 ± 0.34	-13.83 ± 0.93	**1020.63** ± 818.98	0.92 ± 0.04	1.79 ± 0.34	32961.58 ± 128.33	**0.81** ± 0.0	/	-5.12 ± 0.49
Qwen2.5	Greedy Search	**0.54**	**-19.01**	41376.66	0.92	1.62	72215.33	**0.49**	/	**-6.08**
	Sampling	0.36 ± 0.04	-20.22 ± 0.13	24335.42 ± 422.97	0.94 ± 0.02	**1.27** ± 0.32	83200.30 ± 2546.42	0.40 ± 0.02	/	-6.52 ± 0.22
	Top-k Sampling	0.38 ± 0.07	-20.39 ± 0.22	**23525.21** ± 384.33	**0.97** ± 0.01	1.39 ± 0.34	**86300.97** ± 152.16	0.36 ± 0.01	/	-6.22 ± 0.06
	Top-p Sampling	0.39 ± 0.09	-20.21 ± 0.48	31330.99 ± 34652.33	0.95 ± 0.03	1.70 ± 0.32	82882.25 ± 6073.07	0.39 ± 0.05	/	-6.19 ± 0.28

suggesting a small-size decoding space depending on the optimization target and AS type.

6. The PRISM dataset

Composition.

Building upon the experiments presented in the previous sections, we introduce PRISM, a first-of-its-kind dataset that collects over 2M artificial summaries generated using various decoding settings. PRISM includes an instance for each inference run, detailing all metadata (dataset, model, decoding strategy, hyperparameter values), average decoding time per instance (milliseconds), carbon emissions (kg), and metric scores. To further enhance the dataset's value, we include per-token probability distributions. The source files (approximately 10 GB) are stored in JSONL format and are publicly available for download through the HuggingFace Datasets platform.[10] For example, to access the summaries predicted by a Beam Search run, you need to install the `datasets` Python library and follow the instructions shown in Fig. 11. For space efficiency, we

[10] The Huggingface dataset will be public in case of acceptance.

```python
from datasets import load_dataset
# Download PRISM locally and load it as a Dataset
prism = load_dataset("PRISM")
# The first Beam Search run
run = prism["beam_search"][0]
# The predicted summaries
run["predictions"]
```

Fig. 11. PRISM HuggingFace Dataset.

separately release the gold document–summary AS pairs that each run relies on. Additional information on the project website will be updated regularly to incorporate any future changes, additions, or errata.[11]

Applications.

The potential applications of PRISM are extensive and diverse. Researchers can use this dataset to study new NLG metrics. Additionally, it provides a unique opportunity to benchmark decoding strategies against a multitude of established baselines. Beyond this, PRISM offers the ability to train LMs to emulate the token choice of one or more strategies for style control (Goyal et al., 2022) or automatic hyperparameter optimization (Chen et al., 2022). Decoding strategies are complex algorithms that are hard to incorporate into end-to-end networks due to their non-differentiable nature. In light of the limited success in designing exact differentiable versions of decoding strategies (e.g., Top-k Sampling (Jang et al., 2017), Beam Search (Collobert et al., 2019)), PRISM emerges as an indispensable asset for creating approximated modules. Future works may use our data (probability distributions, selected tokens, decoding information) to train differentiable strategy emulators. For example, a foundational model could be instruction fine-tuned with a cross-entropy loss to assign maximum probabilities to the tokens that a specific decoding strategy with a desired setting—defined in the prompt—would select if faced with the same probability distributions provided as input. Alternatively, several LoRA adapters (Hu et al., 2022) could be individually trained to alter the behavior of the base model, each toward a distinct token selection strategy; the resulting adapters could then be integrated into the Mixture-of-Experts layers of sparse models (Wu et al., 2024), switching between them or mixing their choices with a routing mechanism. This would, for the first time, enable the automation of the strategy choice and its hyperparameters' optimization (Chen et al., 2022).

7. Conclusion

The rapid growth of transformer-based summarizers is offset by poor control over decoding strategies, which exhibit cloudy task-specific qualities due to the frequent release of new models. In this paper, we demystify the role of decoding-time methods for abstractive summarization. Our full-scale study comprises thorough quantitative and qualitative breakdowns, covering various decoding setups, autoregressive models, datasets, and evaluation metrics. Empirical results demonstrate how generative heuristics and their hyperparameters can overturn predicted summaries, where optimal choices depend on target quality dimensions and the summarization type at hand (i.e., long, short, multi-document). Our findings reveal the uniqueness of abstractive summarization and the best procedures to follow, depending on the case, serving as cautionary notes. Our breath study and the data collected unlock new research avenues, raising expectations for a future marked by greater awareness of the decoding implications and their control. The limitations and next directions are presented in Appendix G.

CRediT authorship contribution statement

Giacomo Frisoni: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Luca Ragazzi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **David Cohen:** Validation, Software, Investigation, Formal analysis, Data curation; **Gianluca Moro:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization; **Antonella Carbonaro:** Writing – review & editing, Supervision, Project administration, Methodology, Formal analysis, Conceptualization; **Claudio Sartori:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research is partially supported by:
• Artificial Intelligence for Public Administration Connected (AI-PACT), CUP B47H22004450008 and B47H22004460001, PNRR, mission 4, component 2, investment 2.3;
• DigitAl lifelong pRevEntion (DARE), CUP B53C22006450001, code PNC0000002, Complementary National Plan PNC-I.1;
• Future Artificial Intelligence Research (FAIR), Spoke 8, PNRR—M4C2—Investment 1.3, Extended Partnership PE00000013;
• Chips JU TRISTAN, G.A. 101095947, European Commission and Italian MIMIT;
• AI analysis for risk assessment of empty lymph nodes in endometrial cancer surgery", the Fondazione Cassa di Risparmio in Bologna.

Appendix A. SCOPUS Queries

The bibliometric results reported in the main paper (Fig. 2) were obtained by executing the following queries on the SCOPUS search engine:

- **Abstractive Summarization**
  `TITLE-ABS-KEY(abstractive AND summarization);`
- **Decoding Strategies**
  `TITLE-ABS-KEY(((text AND generation) OR nlg) AND decoding);`
- **Abstractive Summarization + Decoding Strategies**
  `TITLE-ABS-KEY(summarization AND decoding);`
- **Abstractive Summarization + Large Language Models**
  `TITLE-ABS-KEY(abstractive AND summarization AND ((large AND`
  language AND (model OR models)) OR LLM)).

We consider Computational Science conference papers by appending the following: `AND (LIMIT-TO(SUBJAREA, ''COMP'')) AND (LIMIT-TO(DOCTYPE, ''cp'')).`

Appendix B. Decoding Hyperparameters

For *Contrastive Search*, we modulate α and k in $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ and $\{20, 30, 40, 50, 60\}$, respectively. For *Beam Search*, we choose the

[11] https://prism.github.io. The web page will be public in case of acceptance.

Table C.14
Example attached to annotation instructions.

Document
African elephants are the largest land animals on Earth. They are known for their distinctive large ears and tusks, which are actually elongated incisor teeth. These gentle giants live in herds led by a matriarch and have a complex social structure. African elephants face threats from poaching and habitat loss, which have led to a decline in their population.
Gold Summary
African elephants live in herds led by a matriarch. Despite their gentle nature, they face endangerment due to poaching and habitat loss, leading to a declining population. **Topics**: African elephants - Social structure - Threats they face.
Summary 1 (High Recall, High Precision, Low Faithfulness)
African elephants live in herds led by a matriarch. They are not endangered and face no threats. **Explanation**: Summary 1 has a *high Recall* because it includes all relevant information/topics of the gold summary; it has a *high Precision* because it includes only the relevant information/topics without adding irrelavant details; the *Faithfulness is low* because it includes incorrect information about their endangerment status.
Summary 2 (High Recall, Low Precision, Low Faithfulness)
African elephants, the Earth's largest land animals, are known for their large ears and tusks. They live individually and face threats from poaching and habitat loss, causing a decline in their population. **Explanation**: Summary 2 has a *high Recall* because it includes all relevant information/topics of the gold summary; it has a *low Precision* because it also includes additional information/topics that are not within the gold summary; the *Faithfulness is low* because it includes incorrect information about their social structure.

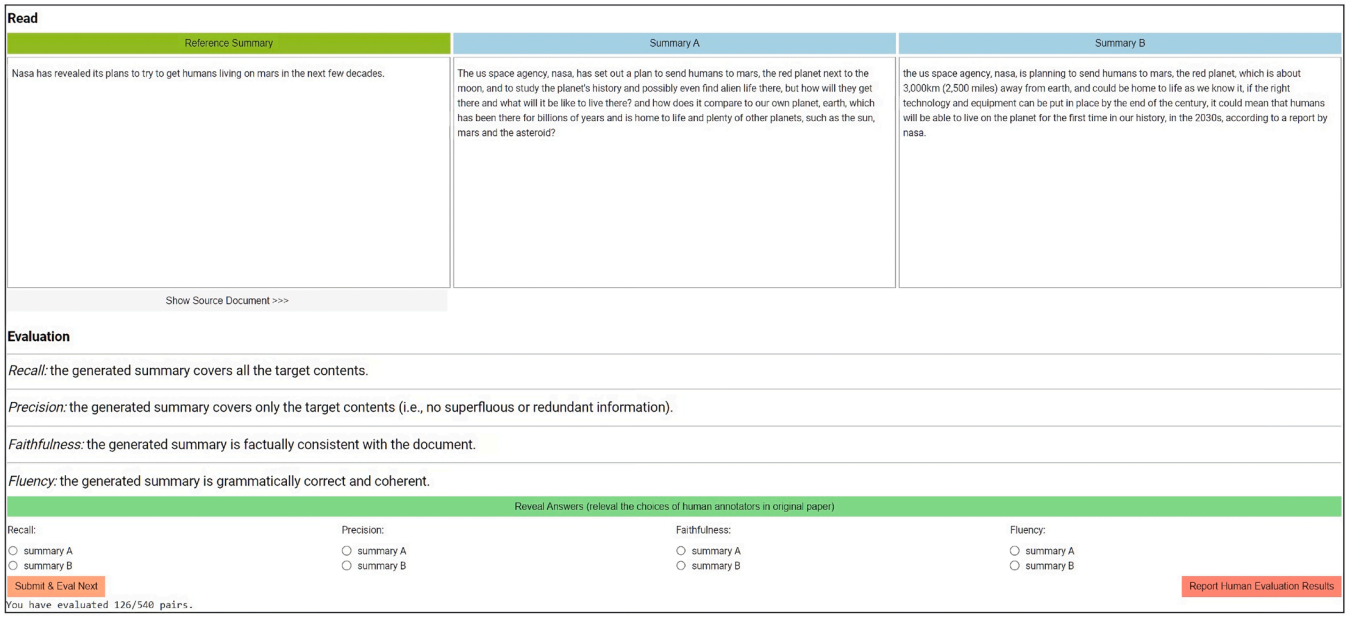


Fig. C.12. Screenshot of the interface used to perform the human evaluation.

beam size b from $[2, 10]$, allowing a trade-off between quality and performance. We keep b unchanged for *Diverse Beam Search*, implementing Δ with the Hamming distance, as suggested by (Vijayakumar et al., 2018), and consider a diversity penalty $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$; the number of subgroups is set equal to b. We maintain the same exploration range for b when trying out *Beam Sampling*. For *Top-k Sampling*, we pick k from $\{20, 30, 40, 50, 60\}$. For *Top-p Sampling*, supported by the results of (DeLucia et al., 2020), we choose p from $\{0.4, 0.6, 0.8\}$. For η *Sampling*, we search η-cutoff over $\{0.0003, 0.0006, 0.0009, 0.002, 0.004\}$ (Hewitt et al., 2022). For pure (ancestral), Top-k, and Top-p Sampling, we also perform an ablation on the temperature hyperparameter, exploring $\tau \in \{0.8, 0.9, 1.0\}$, the best values according to (Holtzman et al., 2020) and (Pasunuru et al., 2021). Where not specified, we maintain default hyperparameters depending on the model configuration.

Artificial summaries often suffer from high redundancy (Xiao & Carenini, 2020). Therefore, we investigate non-negligible word-level n-grams penalties as introduced by (Klein et al., 2017) and (Paulus et al., 2018) specifically for AS. In essence, we force the decoder to avoid outputting the same sequence of words more than once during testing by manually setting the probability of already hypothesized n-grams to 0. We extract n from $\{3, 4, 5\}$ for PUBMED, ARXIV, MULTI-NEWS, and MULTI-LEXSUM, and from $\{2, 3\}$ for XSUM (one-line output) and CNN/DM. This setup results in a minimum of 16 runs per decoding strategy.

As is common in AS, we define a minimum and maximum target size for each dataset (see Table 4), preventing the production of an EOS token before or after a certain threshold. These constraints are determined by examining the summary lengths in the validation tests.

Appendix C. Human Evaluation

Our setup, inspired by (Gu et al., 2022), is illustrated in Fig. C.12, We instruct the annotators to distinguish quality dimensions by presenting the example shown in Table C.14.

Appendix D. Normalized Tabulated Scores

For better interpretability, Table D.15 provides the tabulated version of Fig. 3. Data are categorized by dataset, metric, and decoding strategy, with a 0–10 rescaling applied for value magnitude and graphical clarity. We prioritize the relative relationships among the strategies rather than the precise score values achieved by each. Please note that the exact metric scores are openly available in our PRISM dataset.

Table D.15
Rescaled tabulated scores categorized by dataset, metric, and decoding strategy. For each dataset, the decoding strategies are sorted in decreasing order of their average scores.

Strategy	$\mathcal{R}$	BERTScore	Perplexity ↓	Coverage	Density ↓	Compression	UNR	NID ↓	BARTScore	Carburacy
XSum										
Beam Sampling	3.44	7.31	1.49	5.95	0.31	0.03	6.67	4.65	7.51	4.25
Greedy Search	3.25	8.05	1.28	6.28	0.35	0.04	5.96	6.22	8.11	4.24
Contrastive Search	3.27	8.01	1.37	6.17	0.32	0.04	5.97	6.16	8.09	4.27
η Sampling	3.27	8.01	1.40	6.08	0.35	0.03	5.96	6.17	8.10	4.26
Beam Search	3.30	7.97	1.30	6.27	0.35	0.04	5.85	6.37	8.06	4.28
Diverse Beam Search	3.80	6.82	1.65	5.96	0.28	0.05	4.48	8.21	7.25	4.80
Sampling	2.34	5.16	1.37	3.78	0.14	0.03	7.51	3.54	5.21	3.04
Top-k Sampling	2.22	5.48	1.37	4.23	0.23	0.03	7.09	4.25	5.45	2.90
Top-p Sampling	2.30	5.34	1.48	4.37	0.25	0.03	6.88	4.37	5.41	3.00
Cnn/Dm										
Greedy Search	8.28	9.88	6.55	9.94	5.22	0.07	7.33	4.40	9.91	8.77
η Sampling	8.25	9.86	6.24	9.70	5.26	0.07	7.32	4.40	9.89	8.76
Beam Search	8.30	9.87	6.49	9.98	5.58	0.08	7.42	4.40	9.93	8.80
Diverse Beam Search	8.44	9.90	6.12	9.86	6.12	0.08	7.30	4.49	9.80	8.92
Beam Sampling	8.20	9.79	6.36	9.71	5.28	0.08	7.36	4.38	9.84	8.48
Contrastive Search	8.26	9.87	6.27	9.93	6.20	0.09	7.31	4.40	9.89	8.77
Top-k Sampling	6.51	7.17	6.24	6.49	3.01	0.02	8.23	2.95	6.82	7.14
Sampling	6.46	7.12	6.30	6.49	3.08	0.02	8.20	2.94	6.79	6.99
Top-p Sampling	5.67	5.53	6.15	6.78	3.07	0.08	8.45	2.27	5.02	6.37
PubMed										
Diverse Beam Search	8.75	8.81	3.95	9.73	7.01	0.23	3.43	5.71	8.44	9.10
Contrastive Search	9.14	9.07	4.22	9.87	8.55	0.20	3.70	5.40	8.59	9.41
Greedy Search	9.11	9.06	4.25	9.86	8.89	0.19	3.82	5.34	8.65	9.31
Beam Search	8.97	8.99	4.17	9.86	9.13	0.19	3.98	5.28	8.71	9.26
Beam Sampling	9.04	9.03	4.23	9.73	8.88	0.19	3.85	5.34	8.62	9.24
η Sampling	9.13	9.07	4.23	9.75	8.90	0.18	3.71	5.39	8.59	9.27
Top-k Sampling	5.71	5.33	4.24	6.35	4.75	0.22	6.66	2.75	4.58	6.32
Sampling	5.74	5.33	4.32	6.34	4.71	0.21	6.64	2.75	4.57	6.33
Top-p Sampling	5.81	5.32	3.99	6.33	4.69	0.24	6.31	2.99	4.55	6.36
arXiv										
Diverse Beam Search	9.16	8.37	8.58	9.51	3.66	0.81	3.84	6.34	7.54	9.35
η Sampling	9.53	8.64	9.12	9.73	5.65	0.66	3.91	6.06	7.71	9.59
Contrastive Search	9.50	8.64	9.30	9.72	5.83	0.67	3.92	6.06	7.72	9.77
Beam Sampling	9.47	8.62	9.11	9.73	5.68	0.64	3.94	6.04	7.74	9.30
Greedy Search	9.45	8.59	9.32	9.73	6.62	0.65	3.86	6.02	7.85	9.62
Beam Search	9.37	8.57	9.29	9.79	7.14	0.63	3.90	5.97	7.91	9.55
Top-p Sampling	7.10	5.97	9.29	6.62	3.39	0.64	5.84	3.95	4.68	7.62
Sampling	7.02	6.05	9.40	6.64	3.82	0.62	6.11	3.78	4.76	7.56
Top-k Sampling	7.04	6.06	9.48	6.66	3.88	0.62	6.11	3.79	4.78	7.55
Multi-News										
Greedy Search	8.69	8.93	0.33	8.38	1.32	0.15	4.39	5.81	8.36	9.05
Beam Search	8.64	8.90	0.35	8.35	1.34	0.15	4.37	5.82	8.35	9.00
Contrastive Search	8.58	8.87	0.37	8.27	1.25	0.15	4.36	5.84	8.33	8.98
η Sampling	8.60	8.88	0.36	8.18	1.25	0.15	4.38	5.82	8.32	8.95
Beam Sampling	8.49	8.79	0.38	8.14	1.24	0.14	4.31	5.85	8.27	8.90
Diverse Beam Search	8.16	8.65	0.43	8.09	0.88	0.16	4.07	6.15	8.18	8.61
Top-k Sampling	6.13	5.69	0.37	5.36	0.75	0.14	6.54	3.50	4.90	6.77
Sampling	6.07	5.68	0.36	5.36	0.71	0.14	6.53	3.51	4.90	6.74
Top-p Sampling	6.14	5.55	0.36	5.40	0.74	0.14	6.36	3.60	4.81	6.80
Multi-LexSum										
Beam Search	7.66	7.99	0.18	9.40	0.42	8.75	3.33	5.93	8.10	8.25
Diverse Beam Search	7.32	7.96	0.12	7.52	0.43	9.19	3.58	6.15	7.85	7.98
Beam Sampling	7.59	7.97	0.08	7.70	0.40	8.45	3.20	5.93	8.07	8.05
Greedy Search	7.09	7.59	0.10	8.61	0.40	8.70	1.82	6.69	7.83	7.81
Contrastive Search	5.89	6.87	0.52	8.59	0.44	8.81	3.42	5.75	7.26	6.79
η Sampling	5.94	6.92	0.47	7.65	0.41	8.47	3.36	5.78	7.29	6.81
Top-p Sampling	6.86	7.47	0.26	5.92	0.23	8.46	2.15	6.51	7.73	7.61
Sampling	6.13	6.99	0.37	5.78	0.23	8.66	3.10	5.93	7.38	6.98
Top-k Sampling	6.12	7.02	0.42	5.75	0.22	8.33	3.16	5.93	7.38	6.98

Appendix E. On Hyperparameters Effect

To promote informed configuration of decoding strategies, we consider the relationship between each hyperparameter value and the primary metrics (i.e., $\mathcal{R}$, BERTScore, BARTScore, Perplexity).

Overall.

Unexpectedly, as the value of `no_repeat_ngram_size` (abbreviated as `no_rep`) increases, there is a decrease in $\mathcal{R}$ and BERTScore but a simultaneous increase in factuality (BARTScore). We hypothesize that this can be explained by the elevated abstractiveness of the datasets—a hypothesis reinforced by human evaluation. There is a tension between staying close to the source document and allowing for abstractive modification. Broadly speaking, we find that `no_rep`—a frequently overlooked parameter—plays a substantial role in shaping the output.

Greedy Search (Table G.17).

For SDS and LDS, an increase in `no_rep` is accompanied by a corresponding increase in the value of $\mathcal{R}$, except for XSum. MDS follows an inverse pattern: as `no_rep` increases, both $\mathcal{R}$ and BERTScore decline, while BARTScore increases. The data show a notable correlation across tasks, with only two outliers: $\mathcal{R}$ and BERTScore in XSum, and Perplexity in Multi-LexSum.

Contrastive Search (Table G.18).

It is challenging to pinpoint the selection guidelines for `top_k`, but lower values tend to achieve better results. Decreasing `no_rep` in XSum leads to improved $\mathcal{R}$ scores. In Multi-LexSum, choosing a low `penalty_alpha` is preferable.

Beam Search (Table G.19).

In SDS scenarios, factuality improves as both `no_rep` and beam size b increase. BERTScore also shows a positive trend with `no_rep` and b as well, peaking at $b = 4$ and $b = 5$. However, for XSum, there is a decline

in $\mathcal{R}$ with high values for no_rep and b. This trend is less pronounced in CNN/DM, which benefits from a mid-range b and no_rep= 3. This difference can likely be attributed to the varying length of the summaries. Specifically, since XSUM is much more concise, a strict non-repetition constraint can significantly affect lexical similarity. This opposition is also apparent in terms of Perplexity. In LDS scenarios, ARXIV and PUBMED experience an increase in $\mathcal{R}$ and BERTScore with high no_rep and low b. Again, the value of b sets a boundary on the attainable level of factuality. In MDS, there is a notable shift in the trajectory of $\mathcal{R}$, which increases with no_rep and b. BARTScore follows the same pattern as in SDS and LDS. BERTScore fluctuates more, showing dataset-specific behavior, but generally benefits from higher values of b, regardless of no_rep.

Diverse Beam Search (Table G.20).

In SDS, artificial summaries from XSUM achieve high $\mathcal{R}$ scores with increased no_rep, decreased b, and a low Λ value for diversity_penalty. Conversely, better semantic alignment is achievable with low no_rep, high b, and low Λ. Optimally tuned hyperparameters make Diverse Beam Search superior to standard Beam Search. In LDS, the $\mathcal{R}$ trend remains unchanged, and Λ has less impact on the syntactic surface. The effectiveness curve according to BERTScore changes significantly, favoring high no_rep, low b, and low Λ. In MDS, the results for MULTI-NEWS improve with higher b and Λ. In all task families, there is a consistent upward trend in BARTScore as b and no_rep values increase, except for XSUM.

Sampling (Table G.21).

In SDS and MDS, the behavior is well-established, suggesting the adoption of high temperatures to maximize $\mathcal{R}$, BERTScore, and BARTScore. The only outlier is MULTI-LEXSUM, likely due to the extreme length of its inputs, which leads to inaccurate summaries. Perplexity varies across datasets and shows a weak correlation with changes in hyperparameters.

Top-k Sampling (Table G.22) and Top-p Sampling (Table G.23).

Excluding MULTI-LEXSUM, $\mathcal{R}$, BERTScore and BARTScore improve as temperatures increase. Compared to temperature, top-k, top-p, and no_rep have minimal effect on the metrics. The disparity between Top-k Sampling and Top-p Sampling mainly arises from Perplexity in longer inputs, favoring the latter approach.

η Sampling (Table G.24). The behavior of η Sampling is difficult to predict and control. For BARTScore, results improve with high no_rep, while cut_off has a significant impact only on MULTI-LEXSUM. The rise in $\mathcal{R}$ often aligns with an increase in no_rep, while variations in cut_off exhibit a comparatively weaker impact.

Beam Sampling (Table G.25).

For XSUM, there is a clear positive correlation between b and the metrics $\mathcal{R}$, BERTScore, and BARTScore, with no_rep having a small influence. A similar pattern is observed for CNN/DM, where optimal performance is achieved with $b = 6$, followed by slight oscillations. In LDS, $\mathcal{R}$ and BERTScore have opposite trends in PUBMED and ARXIV. In MDS, $\mathcal{R}$ increases with b and no_rep, with few exceptions in MULTI-LEXSUM.

Appendix F. Best Hyperparameters and Decoding Strategies

Table F.16 presents the hyperparameters that lead to the best $\mathcal{R}$ scores (i.e., ≥ 0.9 quantile).

Appendix G. Limitations and Future Directions

Additional Decoding Strategies. We focused on broadly applicable decoding methods that implement the same likelihood objective as the models. However, some recent strategies introduce additional assumptions, goals, non-metric-agnostic outputs, and task-specific constraints to generation. Notably, FAME (Aralikatte et al., 2021) dynamically biases the decoder to generate summary tokens topically similar to the input, and PINOCCHIO (King et al., 2022) restricts Beam Search only to consider tokens likely supported by the source text. Future work should investigate their inclusion in the analysis.

Additional Models. We conducted a thorough selection process to identify representative encoder–decoder models within each AS family (short, long, multi-document), along with foundational instruct-tuned decoder-only alternatives. The chosen models are open-source and hold state-of-the-art results across several AS benchmarks, featuring checkpoints from reputable companies (Meta, AllenAI, Alibaba) with permissive license terms. Zooming out, our paper is centered on evaluating the models most commonly adopted by the AS community, with parameter counts ranging from 406M to 8B. Extending the study to larger open-source models and closed-source alternatives—such as GPT-4 (OpenAI, 2023)—would be a valuable direction for future research.

Additional Datasets. Regarding testbeds, we carefully selected two well-known representative public datasets per AS type. Future analysis should consider additional datasets from each AS family (e.g., BILLSUM (Kornilova & Eidelman, 2019) for SDS and GOVREPORT (Huang et al., 2021) for LDS). Future work should investigate other AS fields, such as dialogue summarization (Feng et al., 2022), and other NLG tasks, such as question answering (Moro et al., 2024).

Multi-Lingual Resources. The experimental analysis focuses only on English, influenced by the availability of datasets and fine-tuned models in the AS settings analyzed. English serves as a lingua franca in many domains, achieving a broader impact and enabling substantial research due to the wealth of resources. Shifting towards multi-lingual insights requires new models and datasets for each AS type (Moro et al., 2023a; Ragazzi et al., 2024b), effectively doubling the number of runs and computational days. We plan to extend our analysis to multi-lingual settings in future work.

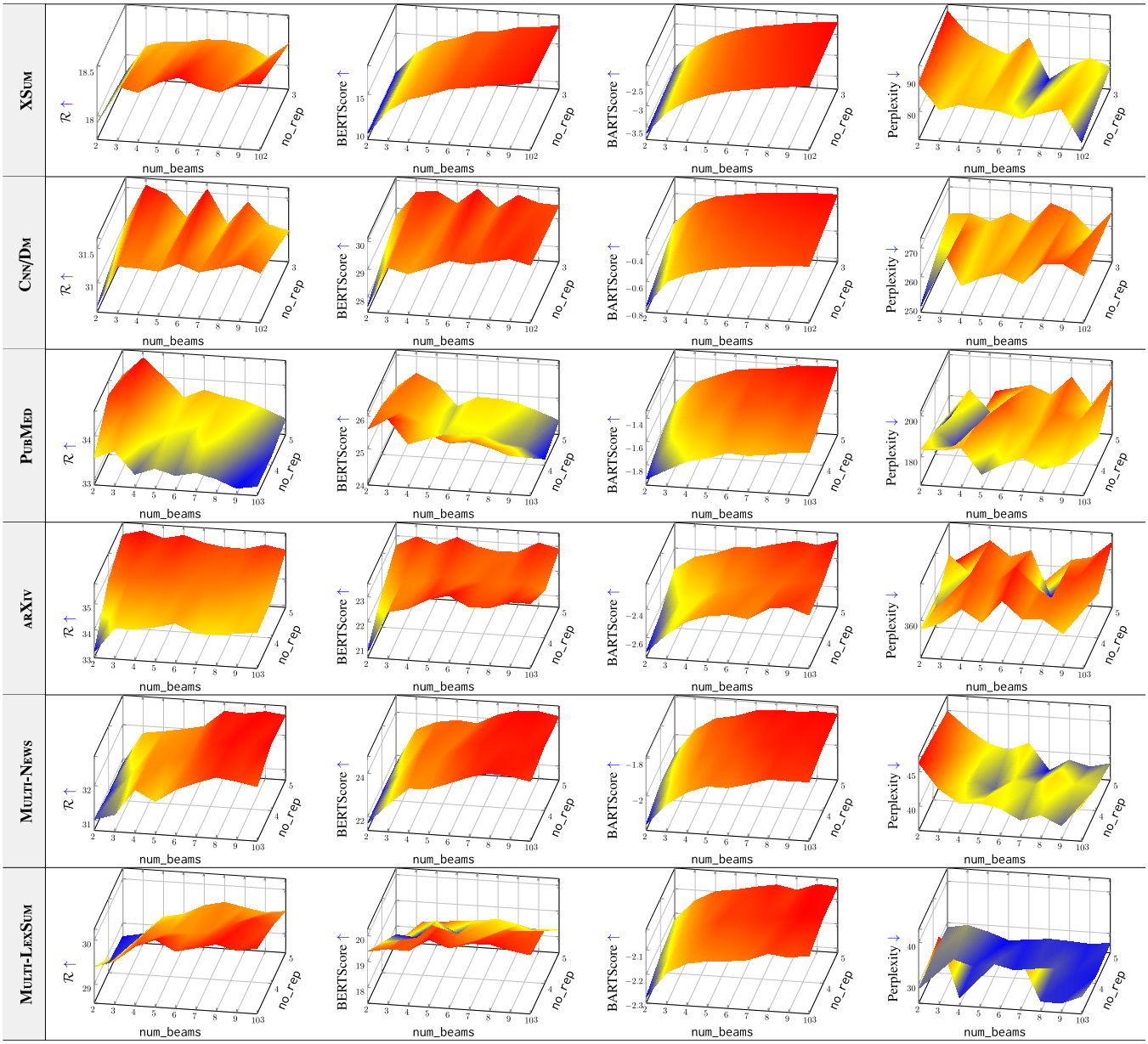
Increased Sample Size. We acknowledge that our experiments used a limited set of samples. Testing all configurations on entire datasets would greatly increase computational needs and runtimes, making it unsustainable with our hardware configuration. To mitigate this problem, we performed stratified sampling to accurately reflect the population and avoid the pitfalls of pure random sampling.

Novel Decoding Strategies. This research focuses on an exhaustive comparison of existing decoding methods alongside encyclopedic hyper-

Table F.16
Best $\mathcal{R}$ hyperparameters for decoding strategies across all datasets.

Greedy Search	Contrastive Search	Beam Search
no_repeat_ngram_size = 3	no_repeat_ngram_size = 5, top_k = 20, penalty_alpha=0.2	no_repeat_ngram_size = 3, num_beams = 9
Diverse Beam Search	**Sampling**	**Top-k Sampling**
no_repeat_ngram_size = 3, num_beams = 10, diversity_penalty = 0.2	no_repeat_ngram_size = 5, temperature = 1.0	no_repeat_ngram_size = 5, temperature = 1.0 top_k = 50
Top-p Sampling	**η Sampling**	**Beam Sampling**
no_repeat_ngram_size = 3, temperature = 1.0, top_k = 20, top_p = 0.4	no_repeat_ngram_size = 5, cut_off = 0.002	no_repeat_ngram_size = 3, num_beams = 10

Table G.17
Hyperparameters effect grouped by datasets – Greedy Search.



parameter permutations and evaluation axes. The goal is to understand the impact of well-established decoding strategies on AS tasks from a neutral perspective and provide fresh insights to the community. Unlike previous work, such as Top-p Sampling (Holtzman et al., 2020) and Contrastive Search (Su et al., 2022), we do not introduce another decoding technique with a limited evaluation of quality attributes (e.g., coherence for Contrastive Search, naturalness and diversity for Top-p Sampling). Instead, we aim for a horizontal assessment on an unprecedented scale, making the introduction of a new heuristic beyond the scope of our study.

Diversity Metrics. We acknowledge that there are multiple ways to summarize a document and that the lack of diversity in decoded solutions is a significant issue in the NLG sector. However, quality and diversity criteria are not equally important in all tasks, and improving one often comes at the cost of the other (Aralikatte et al., 2021). According to (Meister et al., 2022), AS goals prioritize accurate outputs over diverse ones. This contrasts with open-ended tasks such as dialogue generation, which aim to maintain an engaging conversation with a human partner and avoid repetitions.

Unified Metric. Unlike other benchmarks such as GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a), this study uses metrics to evaluate orthogonal quality dimensions in generated summaries, from prediction–target semantic similarity to source–target compression and factuality. These aspects encompass different goals, boundaries, and interpretation keys, making it neither viable nor sound to establish a single overall score. It is worth noting that, based on our literature analysis, no previous research has computed a set of 10 distinct NLG metrics to assess artificial summaries, as we have in PRISM. We hope our work will help researchers identify the finest decoding options for AS and set the stage for a new wave of heuristics.

Table G.18
Hyperparameters effect grouped by datasets – Contrastive Search.

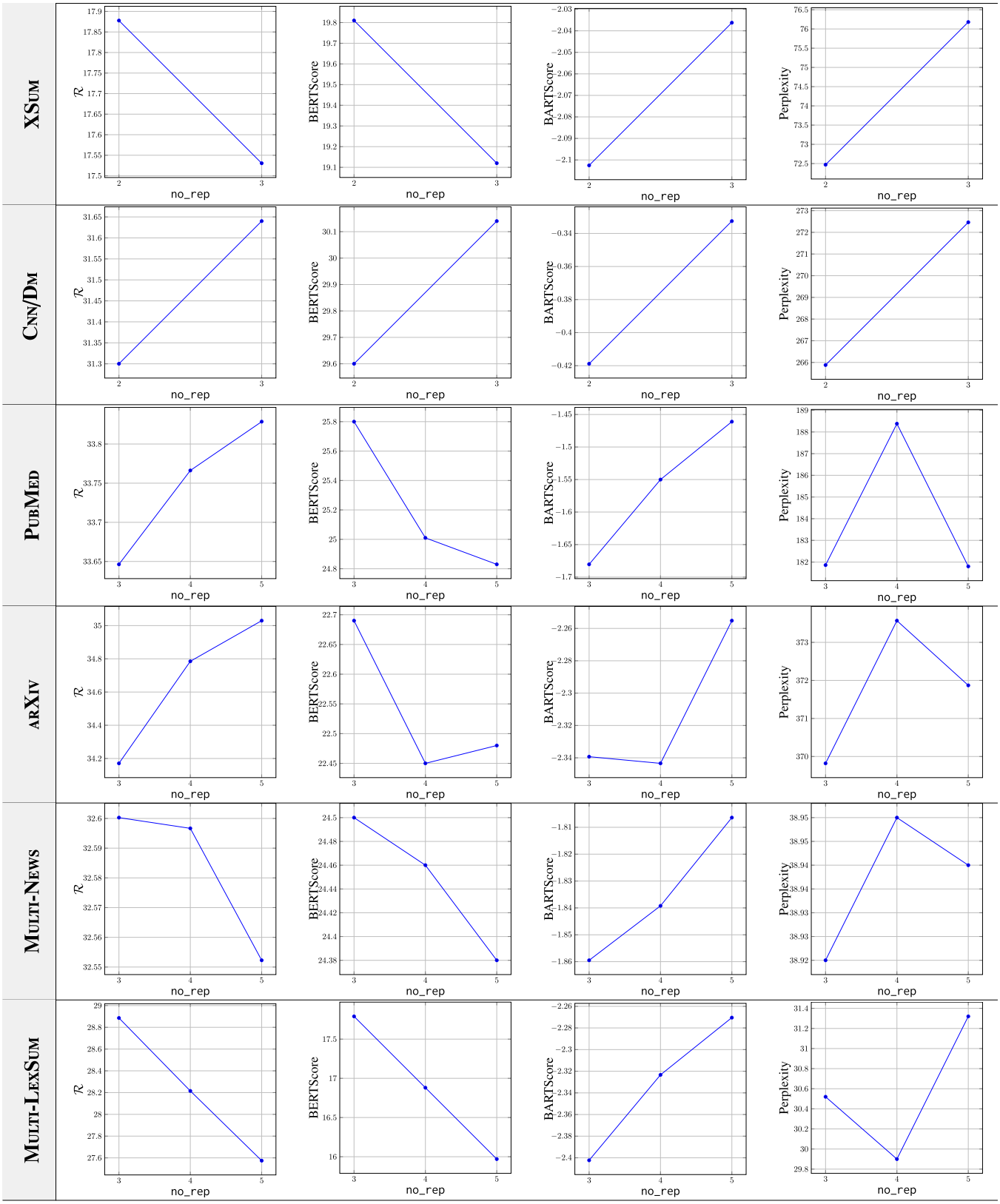


Table G.19
Hyperparameters effect grouped by datasets – Beam Search.

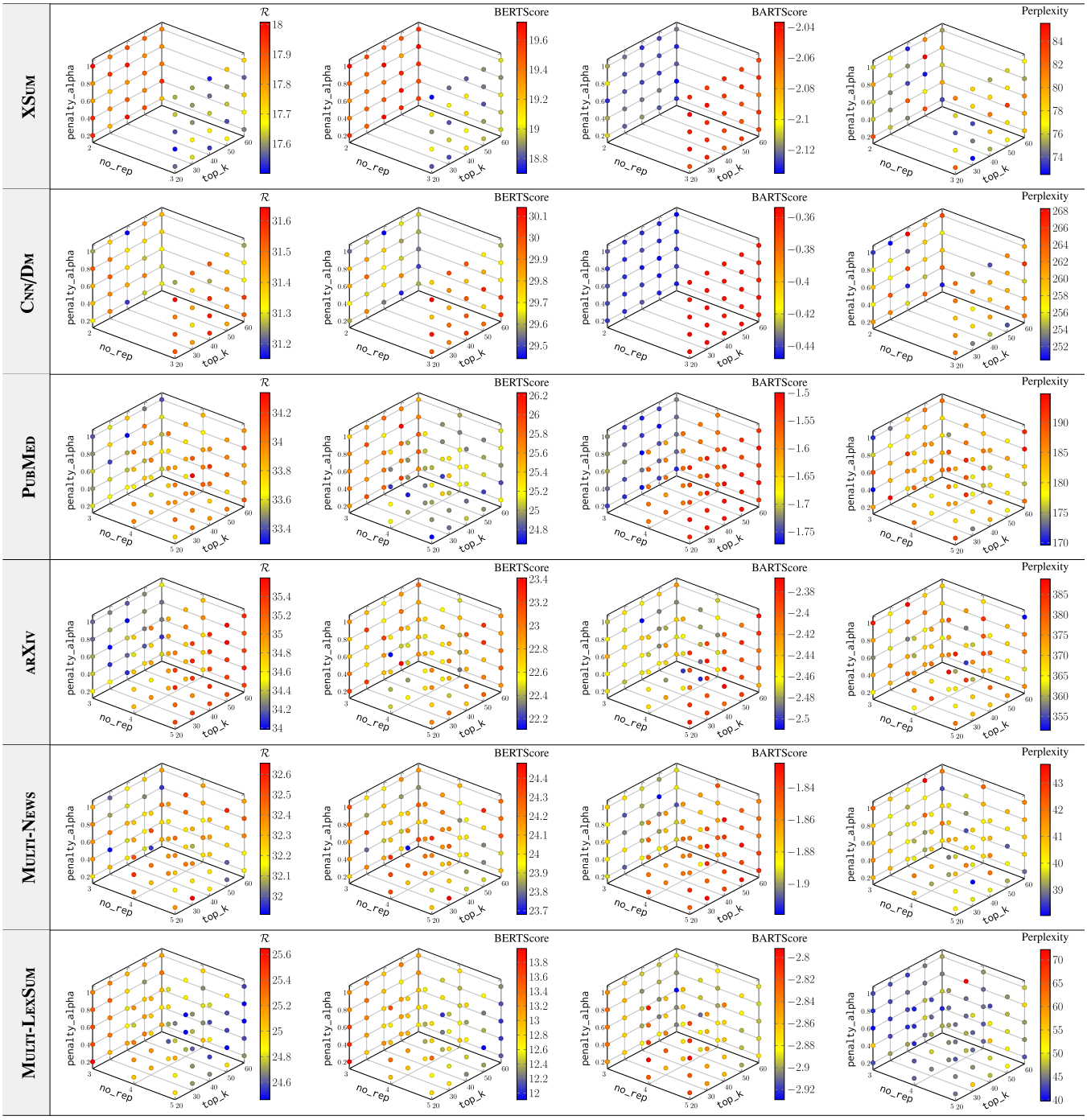


Table G.20
Hyperparameters effect grouped by datasets – Diverse Beam Search.

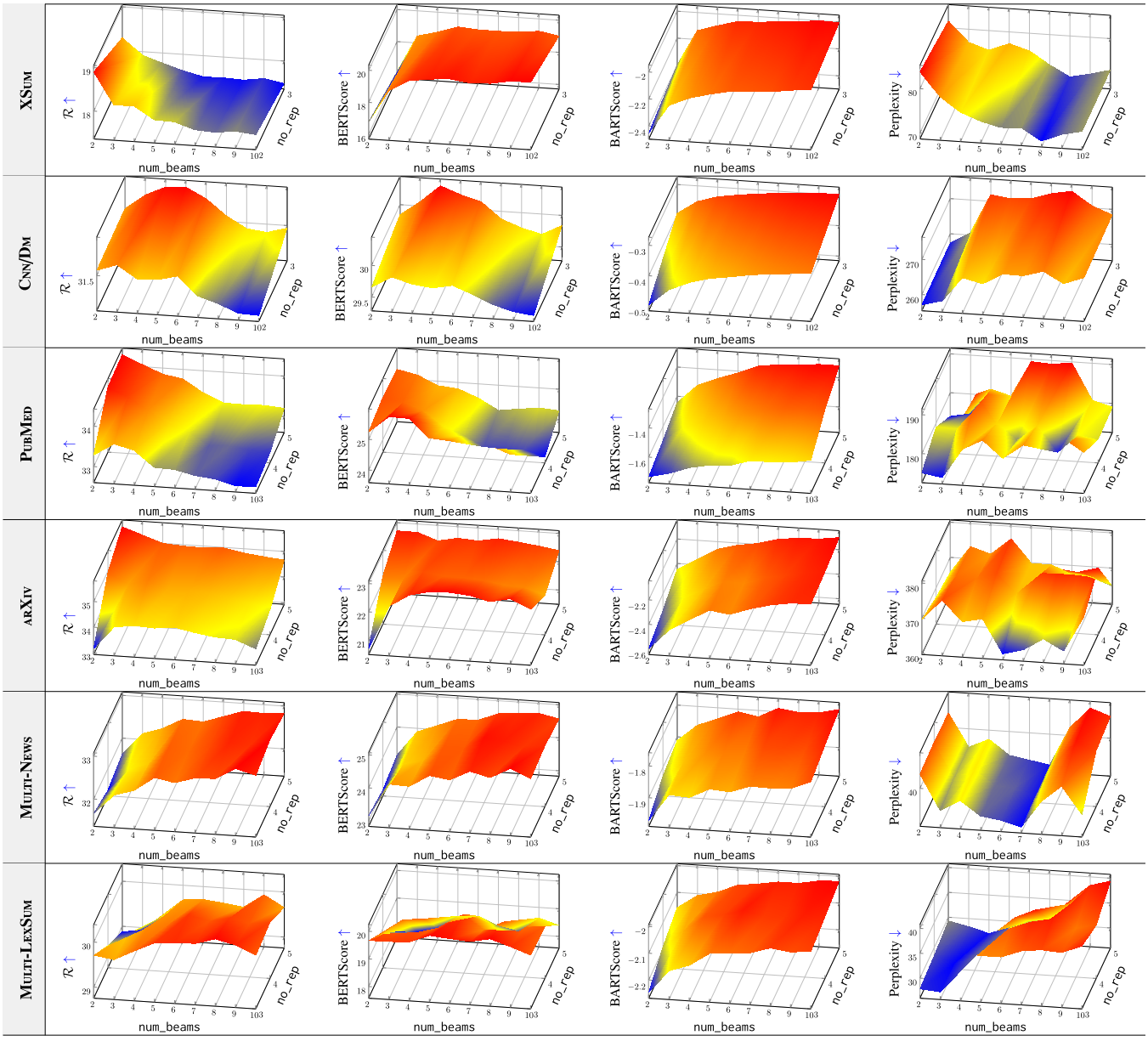


Table G.21
Hyperparameters effect grouped by datasets – Sampling.

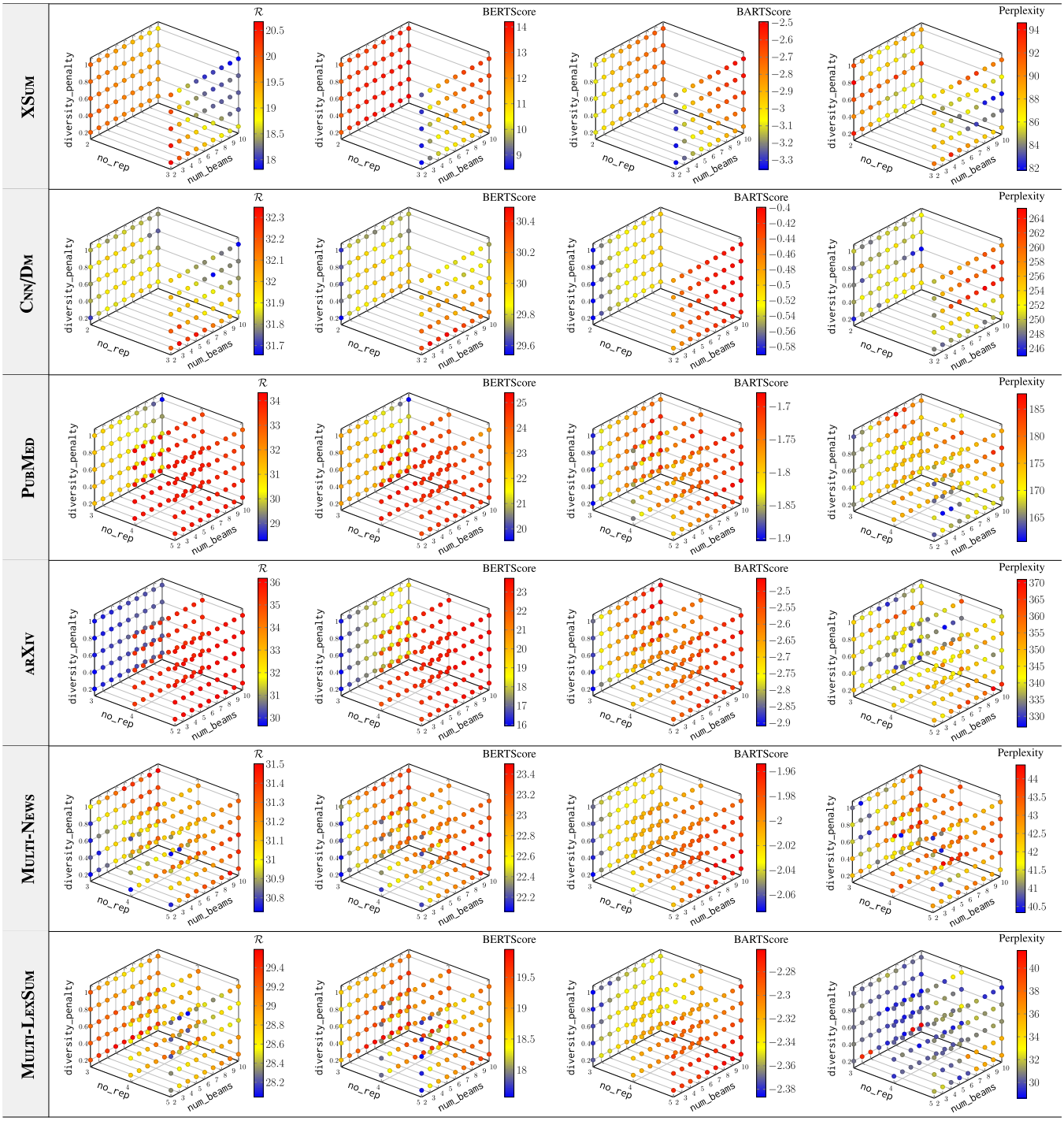


Table G.22
Hyperparameters effect grouped by datasets – Top-k Sampling.

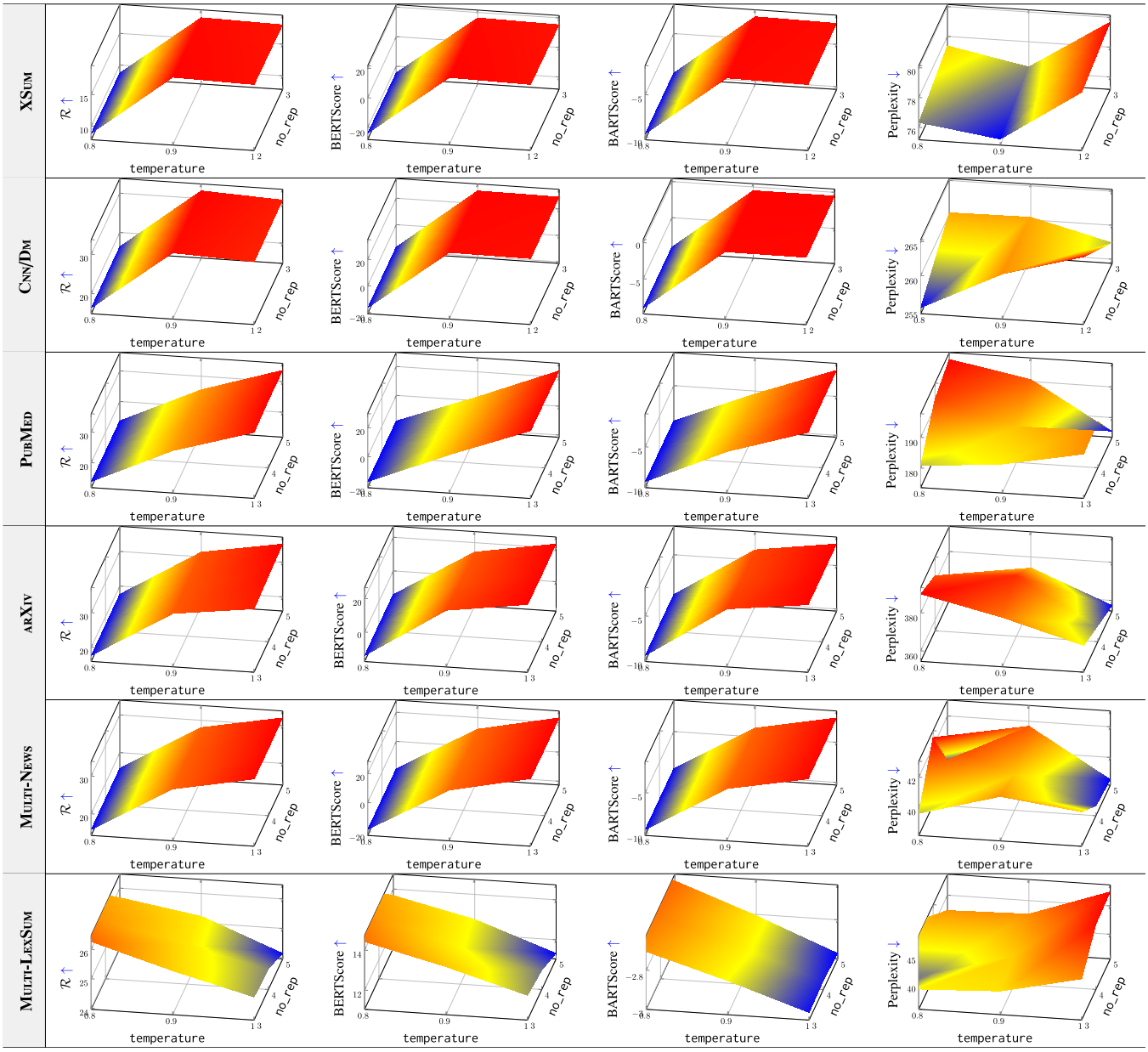


Table G.23
Hyperparameters effect grouped by datasets – Top-p Sampling.

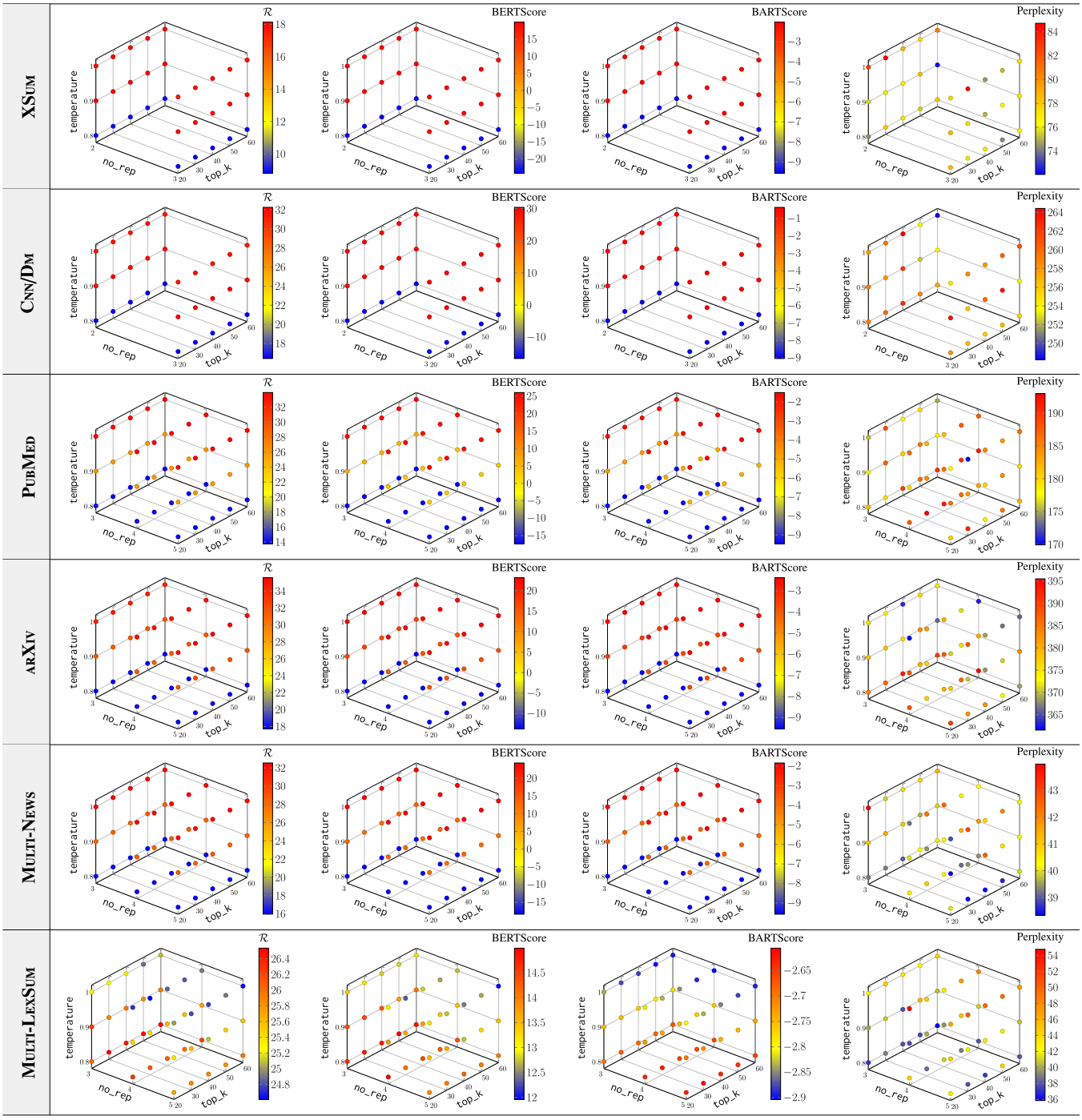


Table G.24
Hyperparameters effect grouped by datasets – η Sampling.

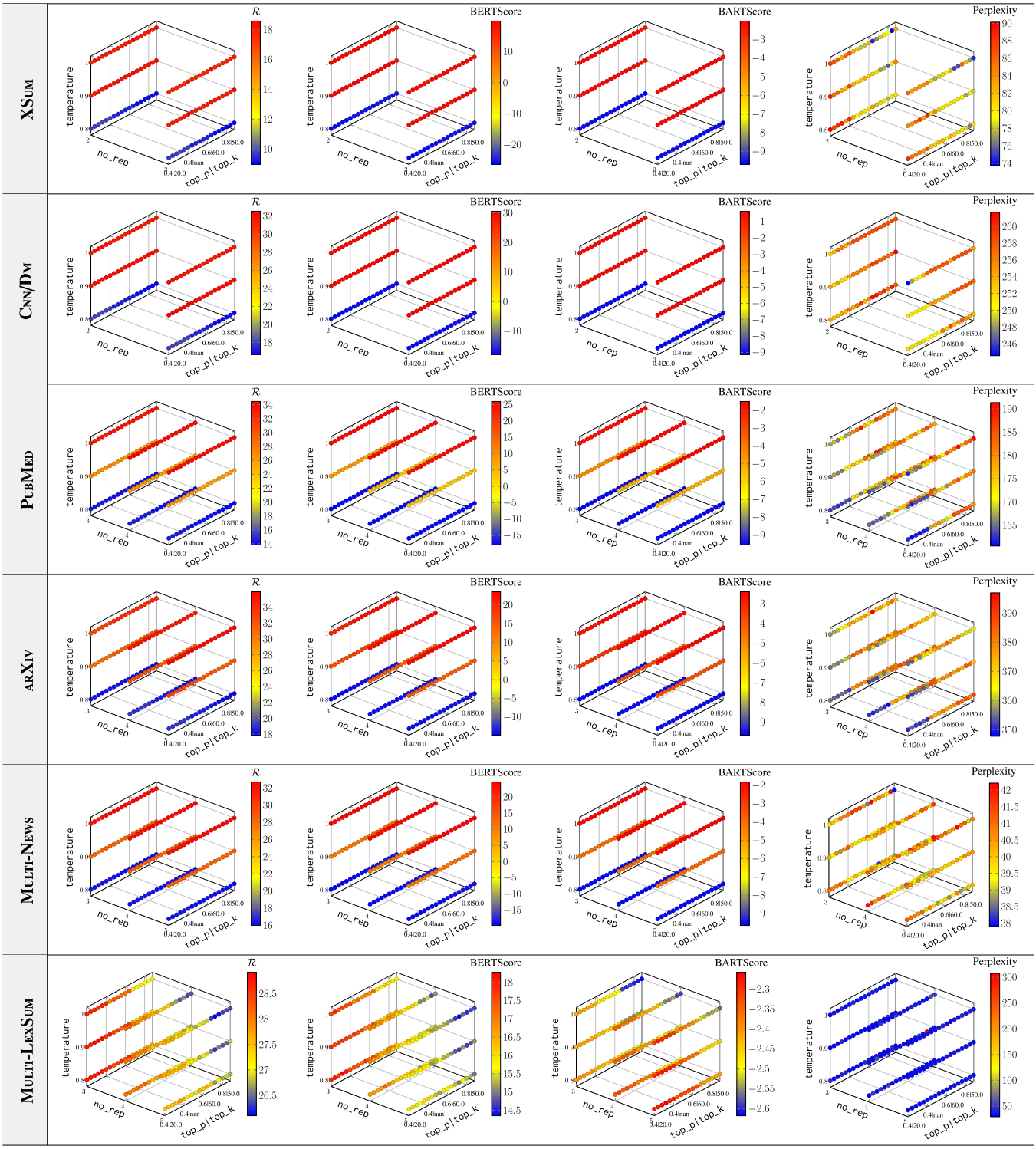
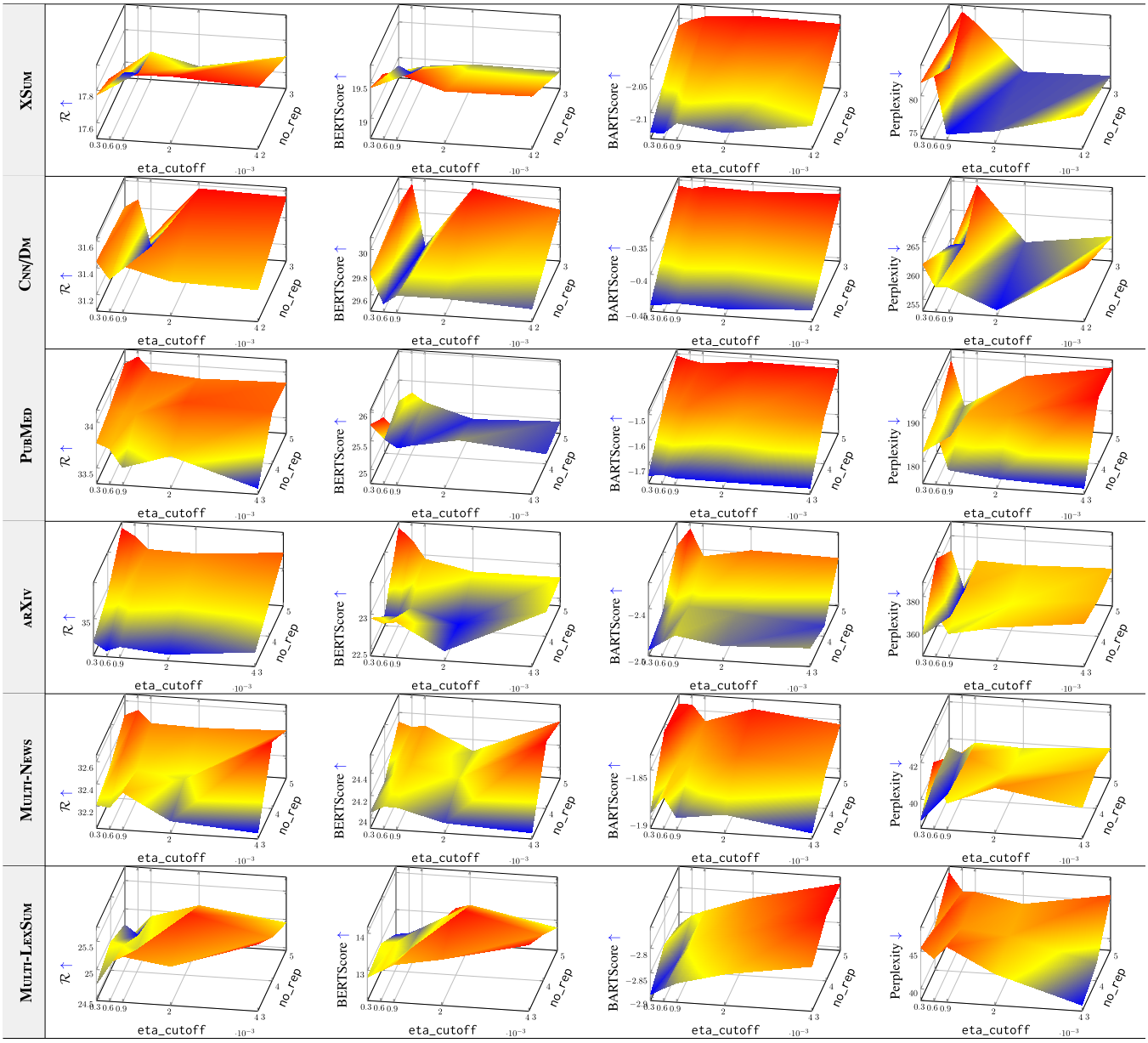


Table G.25
Hyperparameters effect grouped by datasets – Beam Sampling.



References

Aralikatte, R., Narayan, S., Maynez, J., Rothe, S., & McDonald, R. T. (2021). Focus attention: Promoting faithfulness and diversity in summarization. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, ACL/IJCNLP 2021, (volume 1: Long papers), virtual event, august 1–6, 2021* (pp. 6078–6095). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.ACL-LONG.474

Basu, S., Ramachandran, G. S., Keskar, N. S., & Varshney, L. R. (2021). Mirostat: A neural text decoding algorithm that directly controls perplexity. In *9Th international conference on learning representations, ICLR 2021, virtual event, austria, may 3–7, 2021*. OpenReview.net. https://openreview.net/forum?id=W1G1JZEIy5_.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv:2004.05150.

Bird, S. (2006). NLTK: The natural language toolkit. In N. Calzolari, C. Cardie, & P. Isabelle (Eds.), *ACL 2006, 21St international conference on computational linguistics and 44th annual meeting of the association for computational linguistics, proceedings of the conference, sydney, australia, 17–21 july 2006*. The Association for Computer Linguistics. https://aclanthology.org/P06-4018/. https://doi.org/10.3115/1225403.1225421

Caccia, M., Caccia, L., Fedus, W., Larochelle, H., Pineau, J., & Charlin, L. (2020). Language GANs falling short. In *8Th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26–30, 2020* https://dblp.org/rec/conf/iclr/CacciaCFLPC20.bib. https://openreview.net/forum?id=BJgza6VtPB.

Cao, S., & Wang, L. (2022). HIBRIDS: Attention with hierarchical biases for structure-aware long document summarization. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 786–807). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.58

Cao, Y., Bi, W., Fang, M., Shi, S., & Tao, D. (2022). A model-agnostic data manipulation method for persona-based dialogue generation. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 7984–8002). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.550

Chen, Y., Song, X., Lee, C., Wang, Z., Zhang, R., Dohan, D., Kawakami, K., Kochanski, G., Doucet, A., Ranzato, M., Perel, S., & de Freitas, N. (2022). Towards learning universal hyperparameter optimizers with transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems 35: Annual conference on neural information processing systems 2022, neurIPS 2022, new orleans, LA, USA, november 28, - december 9, 2022*. http://papers.nips.cc/paper_files/paper/2022/hash/cf6501108fced72ee5c47e2151c4e153-Abstract-Conference.html.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez,

J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., & Fiedel, N. (2023). PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research : JMLR*, *24*, 240:1–240:113. http://jmlr.org/papers/v24/22-1144.HTML.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In M. A. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT, new orleans, louisiana, USA, june 1–6, 2018, volume 2 (short papers)* (pp. 615–621). Association for Computational Linguistics. https://doi.org/10.18653/V1/N18-2097

Collobert, R., Hannun, A. Y., & Synnaeve, G. (2019). A fully differentiable beam search decoder. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 june 2019, long beach, california, USA* (pp. 1341–1350). PMLR (vol. *97*). Proceedings of Machine Learning Research. http://proceedings.mlr.press/v97/collobert19a.html.

Das, M., & Balke, W. (2022). Quantifying bias from decoding techniques in natural language generation. In N. Calzolari, C. Huang, H. Kim, J. Pustejovsky, L. Wanner, K. Choi, P. Ryu, H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, & S. Na (Eds.), *Proceedings of the 29th international conference on computational linguistics, COLING 2022, gyeongju, republic of korea, october 12–17, 2022* (pp. 1311–1323). International Committee on Computational Linguistics. https://aclanthology.org/2022.coling-1.112.

DeLucia, A., Mueller, A., Li, X. L., & Sedoc, J. (2020). Decoding methods for neural narrative generation. arXiv:2010.07375.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019, minneapolis, mn, usa, june 2–7, 2019, volume 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/V1/N19-1423

DeYoung, J., Beltagy, I., van Zuylen, M., Kuehl, B., & Wang, L. L. (2021). Ms^2: Multi-document summarization of medical studies. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event / punta cana, dominican republic, 7–11 november, 2021* (pp. 7494–7513). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.EMNLP-MAIN.594

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K. et al., (2024). The llama 3 herd of models. arXiv:2407.21783. https://doi.org/10.48550/ARXIV.2407.21783

Eikema, B., & Aziz, W. (2020). Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th international conference on computational linguistics, COLING 2020, barcelona, spain (online), december 8–13, 2020* (pp. 4506–4520). International Committee on Computational Linguistics. https://doi.org/10.18653/V1/2020.COLING-MAIN.398

Fabbri, A. R., Li, I., She, T., Li, S., & Radev, D. R. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, florence, italy, july 28- august 2, 2019, volume 1: Long papers* (pp. 1074–1084). Association for Computational Linguistics. https://doi.org/10.18653/V1/P19-1102

Fan, A., Lewis, M., & Dauphin, Y. N. (2018). Hierarchical neural story generation. In I. Gurevych, & Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, melbourne, australia, july 15–20, 2018, volume 1: Long papers* (pp. 889–898). Association for Computational Linguistics. https://aclanthology.org/P18-1082/. https://doi.org/10.18653/V1/P18-1082

Feng, X., Feng, X., & Qin, B. (2022). A survey on dialogue summarization: Recent advances and new frontiers. In L. De Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI 2022, vienna, austria, 23–29 july 2022* (pp. 5453–5460). ijcai.org. https://doi.org/10.24963/ijcai.2022/764

Frisoni, G., Italiani, P., Boschi, F., & Moro, G. (2022). Enhancing biomedical scientific reviews summarization with graph-based factual evidence extracted from papers. In A. Cuzzocrea, O. Gusikhin, W. M. P. van der Aalst, & S. Hammoudi (Eds.), *Proceedings of the 11th international conference on data science, technology and applications, DATA 2022, lisbon, portugal, july 11–13, 2022* (pp. 168–179). SCITEPRESS. https://doi.org/10.5220/0011354900003269

Frisoni, G., Italiani, P., Salvatori, S., & Moro, G. (2023). Cogito ergo summ: Abstractive summarization of biomedical papers via semantic parsing graphs and consistency rewards. In B. Williams, Y. Chen, & J. Neville (Eds.), *Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, washington, dc, usa, february 7–14, 2023* (pp. 12781–12789). AAAI Press. https://doi.org/10.1609/AAAI.V37I11.26503

Fu, Z., Lam, W., Yu, Q., So, A. M., Hu, S., Liu, Z., & Collier, N. (2023). Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder. arXiv:2304.04052. https://doi.org/10.48550/ARXIV.2304.04052

Gao, Y., Wang, W., Herold, C., Yang, Z., & Ney, H. (2020). Towards a better understanding of label smoothing in neural machine translation. In K. Wong, K. Knight, & H. Wu (Eds.), *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, AACL/IJCNLP 2020, suzhou, china, december 4–7, 2020* (pp. 212–223). Association for Computational Linguistics. https://aclanthology.org/2020.aacl-main.25/.

Giulianelli, M., Baan, J., Aziz, W., Fernández, R., & Plank, B. (2023). What comes next? evaluating uncertainty in neural text generators against human production variability. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 conference on empirical methods in natural language processing, EMNLP 2023, singapore, december 6–10, 2023* (pp. 14349–14371). Association for Computational Linguistics. https://aclanthology.org/2023.emnlp-main.887.

Gong, N., & Yao, N. (2023). A generalized decoding method for neural text generation. *Computer Speech & Language*, *81*, 101503. https://doi.org/10.1016/J.CSL.2023.101503

González, J. á., Louis, A., & Cheung, J. C. K. (2022). Source-summary entity aggregation in abstractive summarization. In N. Calzolari, C. Huang, H. Kim, J. Pustejovsky, L. Wanner, K. Choi, P. Ryu, H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, & S. Na (Eds.), *Proceedings of the 29th international conference on computational linguistics, COLING 2022, gyeongju, republic of korea, october 12–17, 2022* (pp. 6019–6034). International Committee on Computational Linguistics. https://aclanthology.org/2022.coling-1.526.

Goyal, T., Rajani, N., Liu, W., & Kryscinski, W. (2022). Hydrasum: Disentangling style features in text summarization with multi-decoder models. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022, abu dhabi, united arab emirates, december 7–11, 2022* (pp. 464–479). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.30

Graves, A., (2013). Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850.

Grusky, M., Naaman, M., & Artzi, Y., (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In M. A. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2018, new orleans, louisiana, USA, june 1–6, 2018, volume 1 (long papers)* 708–719. Association for Computational Linguistics https://dblp.org/rec/conf/naacl/GruskyNA18.bib. https://doi.org/10.18653/V1/N18-1065.

Grusky, M. (2023). Rogue scores. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1914–1934). Toronto, Canada: Association for Computational Linguistics. https://aclanthology.org/2023.acl-long.107. https://doi.org/10.18653/v1/2023.acl-long.107

Gu, N., Ash, E., & Hahnloser, R. H. R. (2022). Memsum: Extractive summarization of long documents using multi-step episodic markov decision processes. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 6507–6522). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.450

Guo, M., Ainslie, J., Uthus, D. C., Ontañón, S., Ni, J., Sung, Y., & Yang, Y. (2022). Longt5: Efficient text-to-text transformer for long sequences. In M. Carpuat, M. de Marneffe, & I. V. M. Ruíz (Eds.), *Findings of the association for computational linguistics: NAACL 2022, seattle, wa, united states, july 10–15, 2022* (pp. 724–736). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.FINDINGS-NAACL.55

Han, X.-W., Zheng, H.-T., Chen, J.-Y., & Zhao, C.-Z. (2019). Diverse decoding for abstractive document summarization. *Applied Sciences*, *9*(3). https://www.mdpi.com/2076-3417/9/3/386. https://doi.org/10.3390/app9030386

He, P., Peng, B., Wang, S., Liu, Y., Xu, R., Hassan, H., Shi, Y., Zhu, C., Xiong, W., Zeng, M., Gao, J., & Huang, X. (2023). Z-Code++: A pre-trained language model optimized for abstractive summarization. In A. Rogers, J. L. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2023, toronto, canada, july 9–14, 2023* (pp. 5095–5112). Association for Computational Linguistics. https://doi.org/10.18653/V1/2023.ACL-LONG.279

Hewitt, J., Manning, C. D., & Liang, P. (2022). Truncation sampling as language model desmoothing. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Findings of the association for computational linguistics: EMNLP 2022, abu dhabi, united arab emirates, december 7–11, 2022* (pp. 3414–3427). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.249

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. In *8Th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26–30, 2020*. OpenReview.net. https://openreview.net/forum?id=rygGQyrFvH.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *The tenth international conference on learning representations, ICLR 2022, virtual event, april 25–29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, K., Singh, S., Ma, X., Xiao, W., Nan, F., Dingwall, N., Wang, W. Y., & McKeown, K. R. (2023). SWING: Balancing coverage and faithfulness for dialogue summarization. In A. Vlachos, & I. Augenstein (Eds.), *Findings of the association for computational lin-*

guistics: EACL 2023, dubrovnik, croatia, may 2–6, 2023 (pp. 512–525). Association for Computational Linguistics. https://doi.org/10.18653/V1/2023.FINDINGS-EACL.37

Huang, L., Cao, S., Parulian, N. N., Ji, H., & Wang, L. (2021). Efficient attentions for long document summarization. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2021, online, june 6–11, 2021* (pp. 1419–1436). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.NAACL-MAIN.112

Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., & Callison-Burch, C. (2019). Comparison of diverse decoding methods from conditional language models. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, florence, italy, july 28- august 2, 2019, volume 1: Long papers* (pp. 3752–3762). Association for Computational Linguistics. https://doi.org/10.18653/V1/P19-1365

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with gumbel-softmax. In *5Th international conference on learning representations, ICLR 2017, toulon, france, april 24–26, 2017, conference track proceedings*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee.

Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K., (1977). Perplexity-a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, *62*(S1), S63. https://doi.org/10.3390/INFO12090355

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12), 248:1–248:38. https://doi.org/10.1145/3571730

Kalyan, K. S., Rajasekharan, A., & Sangeetha, S. (2021). AMMUS : A survey of transformer-based pretrained models in natural language processing. arXiv:2108.05542.

King, D., Shen, Z., Subramani, N., Weld, D. S., Beltagy, I., & Downey, D. (2022). Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search. arXiv:2203.08436. https://doi.org/10.48550/ARXIV.2203.08436

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. In M. Bansal, & H. Ji (Eds.), *Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, vancouver, canada, july 30, - august 4, system demonstrations* (pp. 67–72). Association for Computational Linguistics. https://doi.org/10.18653/V1/P17-4012

Koh, H. Y., Ju, J., Zhang, H., Liu, M., & Pan, S. (2022). How far are we from robust long abstractive summarization? In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022, abu dhabi, united arab emirates, december 7–11, 2022* (pp. 2682–2698). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.172

Kornilova, A., & Eidelman, V. (2019). Billsum: A corpus for automatic summarization of US legislation. arXiv:1910.00523.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., & Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In J. Flinn, M. I. Seltzer, P. Druschel, A. Kaufmann, & J. Mace (Eds.), *Proceedings of the 29th symposium on operating systems principles, SOSP 2023, koblenz, germany, october 23–26, 2023* (pp. 611–626). ACM. https://doi.org/10.1145/3600006.3613165

Laban, P., Schnabel, T., Bennett, P. N., & Hearst, M. A. (2022). Summac: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions on Machine Learning Research (TMLR)*, *10*, 163–177. https://doi.org/10.1162/TACL_A_00453

Leblond, R., Alayrac, J., Sifre, L., Pislar, M., Lespiau, J., Antonoglou, I., Simonyan, K., & Vinyals, O. (2021). Machine translation decoding beyond beam search. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event / punta cana, dominican republic, 7–11 november, 2021* (pp. 8410–8434). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.EMNLP-MAIN.662

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, & J. R. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, online, july 5–10, 2020* (pp. 7871–7880). Association for Computational Linguistics. https://doi.org/10.18653/V1/2020.ACL-MAIN.703

Lhoest, Q., del Moral, A. V., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Sasko, M., Chhablani, G., Malik, B., Brandeis, S., Scao, T. L., Sanh, V., Xu, C., Patry, N., McMillan-Major, A., Schmid, P., Gugger, S., Delangue, C., Matussière, T., Debut, L., Bekman, S., Cistac, P., Goehringer, T., Mustar, V., Lagunas, F., Rush, A. M., & Wolf, T. (2021). Datasets: A community library for natural language processing. In H. Adel, & S. Shi (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing: System demonstrations, EMNLP 2021, online and punta cana, dominican republic, 7–11 november, 2021* (pp. 175–184). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.EMNLP-DEMO.21

Li, S., & Xu, J. (2023). HierMDS: A hierarchical multi-document summarization model with global-local document dependencies. *Neural Computing & Applications*, *35*(25), 18553–18570. https://doi.org/10.1007/S00521-023-08680-0

Lin, C.-Y., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* Barcelona, Spain, Association for Computational Linguistics, 74–81. https://aclanthology.org/W04-1013.

Lowerre, B. T., (1976). The harpy speech recognition system. Carnegie Mellon University.

Massarelli, L., Petroni, F., Piktus, A., Ott, M., Rocktäschel, T., Plachouras, V., Silvestri, F., & Riedel, S. (2020). How decoding strategies affect the verifiability of generated text. In T. Cohn, Y. He, & Y. Liu (Eds.), *Findings of the association for computational linguistics: EMNLP 2020, online event, 16–20 november 2020* (pp. 223–235). Association for Computational Linguistics (vol. EMNLP *2020*). Findings of ACL. https://doi.org/10.18653/V1/2020.FINDINGS-EMNLP.22

Meister, C., Cotterell, R., & Vieira, T. (2020). Best-first beam search. *Transactions of the Association for Computational Linguistics*, *8*, 795–809. https://doi.org/10.1162/TACL_A_00346

Meister, C., Pimentel, T., Wiher, G., & Cotterell, R. (2023). Locally typical sampling. *Transactions of the Association for Computational Linguistics (TACL)*, *11*, 102–121. https://transacl.org/ojs/index.php/tacl/article/view/3993.

Meister, C., Wiher, G., & Cotterell, R. (2022). On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, *10*, 997–1012. https://transacl.org/ojs/index.php/tacl/article/view/3807.

Moro, G., Piscaglia, N., Ragazzi, L., & Italiani, P. (2023a). Multi-language transfer learning for low-resource legal case summarization. *Artificial Intelligence and Law*, (pp. 1–29). https://doi.org/10.1007/s10506-023-09373-8

Moro, G., & Ragazzi, L. (2022). Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *Thirty-sixth AAAI conference on artificial intelligence, AAAI 2022, thirty-fourth conference on innovative applications of artificial intelligence, IAAI 2022, the twelveth symposium on educational advances in artificial intelligence, EAAI 2022 virtual event, february 22, - march 1, 2022* (pp. 11085–11093). AAAI Press. https://doi.org/10.1609/AAAI.V36I10.21357

Moro, G., & Ragazzi, L. (2023). Align-then-abstract representation learning for low-resource summarization. *Neurocomputing*, *548*, 126356. https://doi.org/10.1016/J.NEUCOM.2023.126356

Moro, G., Ragazzi, L., & Valgimigli, L. (2023b). Carburacy: Summarization models tuning and comparison in eco-sustainable regimes with a novel carbon-aware accuracy. In B. Williams, Y. Chen, & J. Neville (Eds.), *Thirty-seventh AAAI conference on artificial intelligence, AAAI 2023, thirty-fifth conference on innovative applications of artificial intelligence, IAAI 2023, thirteenth symposium on educational advances in artificial intelligence, EAAI 2023, washington, dc, usa, february 7–14, 2023* (pp. 14417–14425). AAAI Press. https://doi.org/10.1609/AAAI.V37I12.26686

Moro, G., Ragazzi, L., & Valgimigli, L. (2023c). Graph-based abstractive summarization of extracted essential knowledge for low-resource scenarios. In K. Gal, A. Nowé, G. J. Nalepa, R. Fairstein, & R. Radulescu (Eds.), *ECAI 2023 - 26Th european conference on artificial intelligence, september 30, - october 4, 2023, kraków, poland - including 12th conference on prestigious applications of intelligent systems (PAIS 2023)* (pp. 1747–1754). IOS Press (*vol. 372*). Frontiers in Artificial Intelligence and Applications. https://doi.org/10.3233/FAIA230460

Moro, G., Ragazzi, L., Valgimigli, L., & Freddi, D. (2022). Discriminative marginalized probabilistic neural method for multi-document summarization of medical literature. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 180–189). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.15

Moro, G., Ragazzi, L., Valgimigli, L., Frisoni, G., Sartori, C., & Marfia, G. (2023d). Efficient memory-enhanced transformer for long-document summarization in low-resource regimes. *Sensors*, *23*(7), 3542. https://doi.org/10.3390/S23073542

Moro, G., Ragazzi, L., Valgimigli, L., & Molfetta, L. (2023e). Retrieve-and-rank end-to-end summarization of biomedical studies. In O. Pedreira, & V. Estivill-Castro (Eds.), *Similarity search and applications - 16th international conference, SISAP 2023, a coruña, spain, october 9–11, 2023, proceedings* (pp. 64–78). Springer (*vol. 14289*). Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-031-46994-7_6

Moro, G., Ragazzi, L., Valgimigli, L., Vincenzi, F., & Freddi, D. (2024). Revelio: Interpretable long-form question answering. In *The second tiny papers track at ICLR 2024, tiny papers @ ICLR 2024, vienna, austria, may 11, 2024*. OpenReview.net. https://openreview.net/forum?id=fyvEJXsaQf.

Nallapati, R., Zhou, B., dos Santos, C. N., Gülçehre, Ç., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Y. Goldberg, & S. Riezler (Eds.), *Proceedings of the 20th SIGNLL conference on computational natural language learning, conll 2016, berlin, germany, august 11–12, 2016* (pp. 280–290). ACL. https://doi.org/10.18653/V1/K16-1028

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don'T give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing, brussels, belgium, october 31, - november 4, 2018* (pp. 1797–1807). Association for Computational Linguistics. https://doi.org/10.18653/V1/D18-1206

OpenAI (2023). GPT-4 technical report. arXiv:2303.08774. https://doi.org/10.48550/ARXIV.2303.08774

Pasunuru, R., Liu, M., Bansal, M., Ravi, S., & Dreyer, M. (2021). Efficiently summarizing text and graph encodings of multi-document clusters. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2021, online, june 6–11, 2021* (pp. 4768–4779). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.NAACL-MAIN.380

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, neurIPS 2019, december 8–14, 2019, vancouver, BC, canada* (pp. 8024–8035). https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Paulus, R., Xiong, C., & Socher, R. (2018). A deep reinforced model for abstractive summarization. In *6Th international conference on learning representations, ICLR 2018, vancouver, bc, canada, april 30, - may 3, 2018, conference track proceedings*. OpenReview.net. https://openreview.net/forum?id=HkAClQgA-.

Phang, J., Zhao, Y., & Liu, P. J. (2023). Investigating efficiently extending transformers for long input summarization. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 conference on empirical methods in natural language processing, EMNLP 2023, singapore, december 6–10, 2023* (pp. 3946–3961). Association for Computational Linguistics. https://aclanthology.org/2023.emnlp-main.240.

Pillutla, K., Swayamdipta, S., Zellers, R., Thickstun, J., Welleck, S., Choi, Y., & Harchaoui, Z. (2021). MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, neurIPS 2021, december 6–14, 2021, virtual* (pp. 4816–4828). https://proceedings.neurips.cc/paper/2021/hash/260c2432a0eecc28ce03c10dadc078a4-Abstract.html.

van der Poel, L., Cotterell, R., & Meister, C. (2022). Mutual information alleviates hallucinations in abstractive summarization. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022, abu dhabi, united arab emirates, december 7–11, 2022* (pp. 5956–5965). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.399

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog, 1*(8), 9.

Ragazzi, L., Italiani, P., Moro, G., & Panni, M. (2024a). What are you token about? differentiable perturbed top-k token selection for scientific document summarization. In L. Ku, A. Martins, & V. Srikumar (Eds.), *Findings of the association for computational linguistics, ACL 2024, bangkok, thailand and virtual meeting, august 11–16, 2024* (pp. 9427–9440). Association for Computational Linguistics. https://doi.org/10.18653/V1/2024.FINDINGS-ACL.561

Ragazzi, L., Moro, G., Guidi, S., & Frisoni, G. (2024b). Lawsuit: A large expert-written summarization dataset of italian constitutional court verdicts. *Artificial Intelligence and Law*. https://api.semanticscholar.org/CorpusID:272530787.

Ragazzi, L., Moro, G., Valgimigli, L., & Fiorani, R. (2025). Cross-document distillation via graph-based summarization of extracted essential knowledge. *IEEE Transactions on Audio, Speech and Language Processing, 33*, 518–527. https://api.semanticscholar.org/CorpusID:273784877.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., & Choi, Y. (2023). Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *The eleventh international conference on learning representations, ICLR 2023, kigali, rwanda, may 1–5, 2023*. OpenReview.net. https://openreview.net/pdf?id=8aHzds2uUyB.

Scialom, T., Dray, P., Lamprier, S., Piwowarski, B., & Staiano, J. (2020). Discriminative adversarial search for abstractive summarization. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 july 2020, virtual event* (pp. 8555–8564). PMLR (*vol. 119*). Proceedings of Machine Learning Research. http://proceedings.mlr.press/v119/scialom20a.html.

Scialom, T., Dray, P., Lamprier, S., Piwowarski, B., Staiano, J., Wang, A., & Gallinari, P. (2021a). Questeval: Summarization asks for fact-based evaluation. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event / punta cana, dominican republic, 7–11 november, 2021* (pp. 6594–6604). Association for Computational Linguistics. https://doi.org/10.18653/V1/2021.EMNLP-MAIN.529

Scialom, T., Dray, P., Staiano, J., Lamprier, S., & Piwowarski, B. (2021b). To beam or not to beam: That is a question of cooperation for language GANs. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, neurIPS 2021, december 6–14, 2021, virtual* (pp. 26585–26597). https://proceedings.neurips.cc/paper/2021/hash/df9028fcb6b065e000ffe8a4f03eeb38-Abstract.html.

Sharma, G., & Sharma, D. (2023). Automatic text summarization methods: A comprehensive review. *SN Computer Science, 4*(1), 33. https://doi.org/10.1007/s42979-022-01446-w

Shen, Z., Lo, K., Yu, L., Dahlberg, N., Schlanger, M., & Downey, D. (2022). Multilexsum: Real-world summaries of civil rights lawsuits at multiple granularities. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems 35: Annual conference on neural information processing systems 2022, neurIPS 2022, new orleans, LA, USA, november 28, - december 9, 2022*. http://papers.nips.cc/paper_files/paper/2022/hash/552ef803bef9368c29e53c167de34b55-Abstract-Datasets_and_Benchmarks.html.

Su, Y., & Collier, N. (2023). Contrastive search is what you need for neural text generation. *Transactions on Machine Learning Research (TMLR), 2023*. https://openreview.net/forum?id=GbkWw3jwL9.

Su, Y., Lan, T., Wang, Y., Yogatama, D., Kong, L., & Collier, N. (2022). A contrastive framework for neural text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems 35: Annual conference on neural information processing systems 2022, neurIPS 2022, new orleans, LA, USA, november 28, - december 9, 2022*. http://papers.nips.cc/paper_files/paper/2022/hash/871cae8f599cb8bbfcb0f58fe1af95ad-Abstract-Conference.html.

Syed, A. A., Gaol, F. L., & Matsuo, T. (2021). A survey of the state-of-the-art models in neural abstractive text summarization. *IEEE Access, 9*, 13248–13265. https://doi.org/10.1109/ACCESS.2021.3052783

Vijayakumar, A. K., Cogswell, M., Selvaraju, R. R., Sun, Q., Lee, S., Crandall, D. J., & Batra, D. (2018). Diverse beam search for improved description of complex scenes. In S. A. McIlraith, & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, (aaai-18), the 30th innovative applications of artificial intelligence (iaai-18), and the 8th AAAI symposium on educational advances in artificial intelligence (eaai-18), new orleans, louisiana, USA, february 2–7, 2018* (pp. 7371–7379). AAAI Press. https://doi.org/10.1609/AAAI.V32I1.12340

Wan, D., & Bansal, M. (2022). FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization. In M. Carpuat, M. de Marneffe, & I. V. M. Ruíz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL 2022, seattle, wa, united states, july 10–15, 2022* (pp. 1010–1028). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.NAACL-MAIN.74

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019a). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, neurIPS 2019, december 8–14, 2019, vancouver, BC, canada* (pp. 3261–3275). https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019b). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7Th international conference on learning representations, ICLR 2019, new orleans, la, usa, may 6–9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJ4km2R5t7.

von Werra, L., Tunstall, L., Thakur, A., Luccioni, S., Thrush, T., Piktus, A., Marty, F., Rajani, N., Mustar, V., & Ngo, H. (2022). Evaluate & evaluation on the hub: Better best practices for data and model measurements. In W. Che, & E. Shutova (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022 - system demonstrations, abu dhabi, uae, december 7–11, 2022* (pp. 128–136). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.EMNLP-DEMOS.13

Weizhe, Y., Graham, N., & Pengfei, L., (2021). BARTScore: Evaluating Generated Text as Text Generation. In M. A. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang & J. W. Vaughan (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, neurips 2021, december 6–14, 2021, virtual* (pp. 27263–27277). https://proceedings.neurips.cc/paper/2021/hash/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Abstract.HTML

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv:1910.03771.

Wu, H., Zheng, H., & Yu, B. (2024). Parameter-efficient sparsity crafting from dense to mixture-of-experts for instruction tuning on general tasks. arXiv:2401.02731. https://doi.org/10.48550/ARXIV.2401.02731

Xiao, W., Beltagy, I., Carenini, G., & Cohan, A. (2022). PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 5245–5263). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.360

Xiao, W., & Carenini, G. (2020). Systematically exploring redundancy reduction in summarizing long documents. In K. Wong, K. Knight, & H. Wu (Eds.), *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, AACL/IJCNLP 2020, suzhou, china, december 4–7, 2020* (pp. 516–528). Association for Computational Linguistics. https://aclanthology.org/2020.aacl-main.51/.

Xu, J., Xiong, C., Savarese, S., & Zhou, Y. (2023). Best-k search algorithm for neural text generation. In A. Rogers, J. L. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2023, toronto, canada, july 9–14, 2023* (pp. 12385–12401). Association for Computational Linguistics. https://doi.org/10.18653/V1/2023.ACL-LONG.692

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., & Qiu, Z. (2024). Qwen2.5 technical report. arXiv:2412.15115. https://doi.org/10.48550/ARXIV.2412.15115

Zarrieß, S., Voigt, H., & Schüz, S. (2021). Decoding methods in neural language generation: A survey. *Information, 12*(9), 355. https://doi.org/10.3390/INFO12090355

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 july 2020, virtual event* (pp. 11328–11339). PMLR (*vol. 119*). Proceedings of Machine Learning Research. http://proceedings.mlr.press/v119/zhang20ae.html.

Zhang, M., Zhou, G., Yu, W., Huang, N., & Liu, W. (2022a). A comprehensive survey of abstractive text summarization based on deep learning. *Computational Intelligence and Neuroscience, 2022*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. In *8Th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, april 26–30, 2020* https://openreview.net/forum?id=SkeHuCVFDr. https://dblp.org/rec/conf/iclr/ZhangKWWA20.bib.

Zhang, Y., Ni, A., Mao, Z., Wu, C. H., Zhu, C., Deb, B., Awadallah, A. H., Radev, D. R., & Zhang, R. (2022b). Summ^n: A multi-stage summarization framework for long input dialogues and documents. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers), ACL 2022, dublin, ireland, may 22–27, 2022* (pp. 1592–1604). Association for Computational Linguistics. https://doi.org/10.18653/V1/2022.ACL-LONG.112