

Received: 30 June 2026; Revised: 12 August 2026; Accepted: 14 September 2026; Published: 30 September 2026

Archs Sci. (2026) Volume 76(3), Pages 111-123, Paper ID 2026313.
<https://doi.org/10.68304/as/76313>

Application of Large Language Models and Natural Language Generation in Content Creation: A Retrieval-Augmented and Preference-Aligned Framework

Jian Zhang^{1,2}, Yuxuan Zheng^{2,3,*}, Feng Ye^{1,2}, Xin Liu³ and An Zeng³

¹School of Information Science and Technology, University of Science and Technology of China, Hefei 230000, Anhui, China

²Product R&D Center, Communication Brain Technology (Zhejiang) Co., Ltd., Hangzhou 310000, Zhejiang, China

³School of Computer Science and Technology, East China Normal University, Shanghai 200333, China

Corresponding author: Yuxuan Zheng (e-mail: 1323121@163.com).

Abstract Large language models (LLMs) have substantially advanced natural language generation (NLG), but high-quality content creation remains constrained by hallucination, limited controllability, retrieval noise, domain shift, and computational cost. This study proposes Retrieval-Augmented and Preference-Aligned Generation (RAP-Gen), a unified framework that integrates retrieval-augmented generation, factual preference alignment, in-context learning, and denoising pre-training. The framework combines a dense semantic retriever with an encoder-decoder generator, introduces pairwise factual-preference supervision for controllable generation, aligns retrieval and generation distributions, and uses structured text corruption to improve robustness to incomplete or noisy inputs. Experiments are reported on summarization, long-form creative writing, and knowledge-intensive question answering using CNN/DailyMail, WritingPrompts, Natural Questions, and general language-modeling data. The reported results show that RAP-Gen improves ROUGE, BERTScore, perplexity, FactScore, and human-rated factuality relative to the baselines listed in the manuscript. Ablation results indicate that removing retrieval, preference alignment, or denoising degrades performance, supporting the complementary roles of external knowledge, preference signals, and robust pre-training. Additional analyses examine the number of retrieved documents, attention over retrieved evidence, robustness under input perturbation, model scaling, and in-context learning. Because the supplied material does not include raw predictions, code, repeated-run variance, annotator agreement statistics, or complete benchmark configurations, the numerical results are interpreted as reported experimental outcomes rather than independently reproducible evidence. The study contributes an integrated design for knowledge-grounded, preference-aware content generation and identifies evaluation and reproducibility requirements for future work.

Index Terms large language models, natural language generation, retrieval-augmented generation, preference alignment, in-context learning, factuality, content creation, denoising pre-training

1. Introduction

Natural Language Generation (NLG) has shifted from rule-based and statistical pipelines toward large-scale neural language models built primarily on the Transformer architecture [1]. Large autoregressive models have demonstrated strong zero-shot, one-shot, and few-shot capabilities, showing that broad pre-training can support task adaptation without conventional task-specific fine-tuning [2]. This transition has enabled rapid expansion of AI-assisted content creation in summarization, question answering, creative writing, coding, personalized communication, scientific assistance, and multimodal applications [3], [4].

The same capabilities create new reliability problems. A content-generation system must do more than produce locally fluent text. In practical applications, the output may need to

be factually grounded, stylistically controlled, attributable to evidence, robust to incomplete input, adaptable across tasks, and efficient enough for deployment. These requirements are particularly important in knowledge-intensive content creation, where incorrect named entities, dates, quotations, causal claims, or technical statements can make otherwise fluent content unusable.

One of the central limitations of LLMs is that much of their knowledge is represented parametrically. Model parameters can encode broad regularities and factual associations, but they do not provide a transparent or readily updateable database. Retrieval-Augmented Generation (RAG) addresses this problem by combining a parametric sequence generator with non-parametric external memory [5]. Retrieved documents can supply current or long-tail evidence and can im-

prove specificity and factual grounding. However, retrieval also introduces new failure modes: irrelevant passages may be selected, multiple documents may conflict, and a fluent generator may still ignore or distort retrieved evidence.

A second line of work concerns alignment. Reinforcement learning from human feedback (RLHF) demonstrates that human comparisons can be used to train reward models and improve the correspondence between model outputs and human preferences [6]. Preference signals are therefore a natural mechanism for controlling generated content. In the present study, the alignment target is narrowed specifically toward factual preference: annotators compare alternative statements and indicate which is more consistent with verifiable information.

A third challenge is hallucination. Hallucination in NLG refers broadly to generated content that is unsupported by the source, inconsistent with available evidence, or factually incorrect despite appearing linguistically plausible. The phenomenon is well documented across generative tasks and becomes especially consequential in open-domain and knowledge-intensive settings [7]. Factuality research has consequently emphasized retrieval, source attribution, external verification, and evaluation methods that decompose long-form text into checkable factual units [8].

In-context learning (ICL) provides another important capability. Rather than changing model parameters, ICL conditions the model on task descriptions and demonstrations placed in the prompt. Large models can often infer an input–output pattern from a small number of examples, although performance depends strongly on demonstration selection, order, formatting, and model scale [2], [9]. This makes ICL attractive for content creation because the same base model can adapt to different styles or tasks with little deployment overhead.

Denosing pre-training provides a complementary robustness mechanism. BART demonstrates that a sequence-to-sequence model can be trained by corrupting text and learning to reconstruct the original sequence, using operations such as text infilling and sentence permutation [10]. T5 similarly treats a wide range of language problems within a unified text-to-text transfer-learning framework [11]. These approaches motivate the structured corruption component of RAP-Gen.

The present study integrates these ideas into a single architecture. RAP-Gen combines: (1) dense semantic retrieval for external knowledge; (2) a factual-preference reward model; (3) in-context adaptation; (4) a bidirectional encoder and autoregressive decoder for conditional generation; and (5) denoising pre-training for robustness. The study asks four questions: whether retrieval improves factual coverage; whether preference alignment increases factual consistency and controllability; whether denoising improves robustness to perturbed inputs; and whether the components provide complementary rather than redundant improvements.

II. Background and Methodological Foundations

A. Factual Consistency and Preference Annotation

Human preference data can convert a difficult absolute-scoring task into a more stable pairwise comparison. For a

given topic or context, two candidate outputs are presented to an annotator, who decides which output is better supported by available evidence. Pairwise preferences are widely used in alignment because they simplify the judgment task and can be converted into a ranking objective [6].

Figure 1 illustrates the factual-preference annotation workflow used in the proposed system.

The figure shows a web-based annotation interface. On the left, there are links for 'View full instructions' and 'View tool guide'. Below these is a note: 'Note: Some questions are control questions. We require good accuracy on our control questions to accept responses.' A larger instruction block says: 'Indicate which one of the following sentences is more factually true with respect to the subject. Using the internet to check whether the sentences are true is encouraged.' The main area is titled 'Which sentence is more factually true?' and contains a 'Subject' field with 'Hemingway'. Below this are two sentences: 'Sentence A: "The Sun Also Rises" is a novel by this author of "A Farewell to Arms"' and 'Sentence B: This author of "The Sun Also Rises" was born in Havana, Cuba, the son of Spanish immigrants'. On the right, there is a 'Select an option' section with four radio buttons: 'Sentence A is more true' (1), 'Sentence B is more true' (2), 'Both sentences are true' (3), and 'Both sentences are completely untrue' (4).

Figure 1: Pairwise Factual-Preference Annotation Framework for Content Generation

For each input x_i , the annotation process produces a preferred response y_i^+ and a less preferred response y_i^- . The dataset therefore contains triples

$$(x_i, y_i^+, y_i^-),$$

with optional associated external evidence $\mathcal{K}_i$. Annotators are instructed to consult external sources when the factual comparison cannot be made reliably from prior knowledge alone. This requirement is intended to reduce preference labels based only on stylistic plausibility.

The preference loss is defined as a logistic ranking objective:

$$\mathcal{L}_{\text{pref}} = -\frac{1}{N} \sum_{i=1}^N \log \sigma(\beta [R_\phi(y_i^+, x_i, \mathcal{K}_i) - R_\phi(y_i^-, x_i, \mathcal{K}_i)]), \quad (1)$$

where R_ϕ denotes the learned factual-preference reward and β controls the sharpness of the pairwise comparison.

The preference objective is combined with knowledge-alignment and representation-consistency terms:

$$\begin{aligned} \mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{pref}} + \lambda \sum_{i=1}^N D_{\text{KL}}(p_\theta(z | x_i) \| q_\phi(z | y_i, \mathcal{K}_i)) \\ + \gamma \sum_{i=1}^N \|f_\theta(x_i, y_i) - g(\mathcal{K}_i)\|_2^2. \end{aligned} \quad (2)$$

The KL term encourages compatibility between model-side and evidence-conditioned latent distributions, while the representation term encourages generated semantic features to remain close to representations derived from external knowledge.

B. In-Context Learning

Figure 2 summarizes the relationship between large-scale pre-training and inference-time in-context adaptation.

During pre-training, the model parameters are optimized across heterogeneous textual sequences. At inference time, the

parameters remain fixed while a context $\mathcal{C}_{\mathcal{T}}$ supplies examples or task descriptions. A high-level objective can be written as

$$\min_{\theta} \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})} [\mathcal{L}_{\mathcal{T}}(f_{\theta}(x; \mathcal{C}_{\mathcal{T}}))]. \quad (3)$$

This expression should be interpreted conceptually: pre-training produces parameters that can support inference-time adaptation from context. It does not imply that ordinary next-token pre-training explicitly optimizes a known distribution of labeled downstream tasks.



Figure 2: Relationship Between Pre-Training and In-Context Learning in Large Language Models

The zero-shot, one-shot, and few-shot settings are compared in Figure 3.

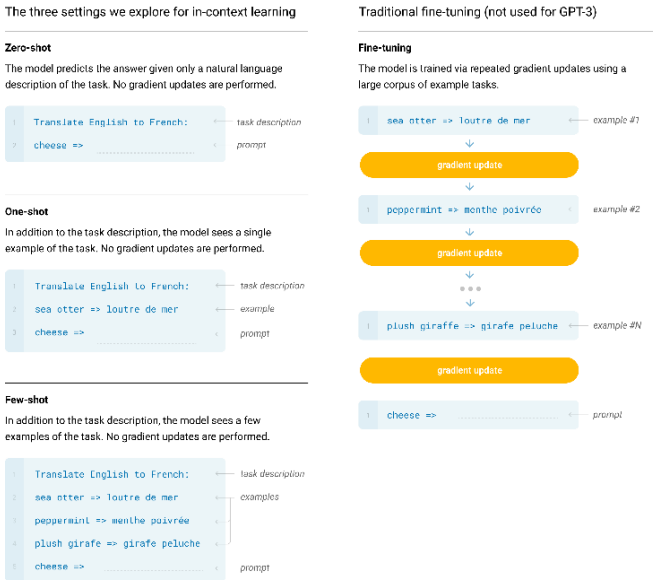


Figure 3: Comparison of In-Context Learning and Gradient-Based Fine-Tuning

Conditional generation under ICL can be written as

$$y \sim P_{\theta}(y | x, \mathcal{C}), \quad (4)$$

where $\mathcal{C}$ contains task instructions and, when available, demonstrations. Conventional fine-tuning instead updates parameters:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \mathcal{L}(f_{\theta}(x_i), y_i). \quad (5)$$

ICL therefore trades persistent task-specific parameter updates for prompt-level adaptation. This can reduce deployment cost but can also increase sensitivity to prompt design and context length.

III. Proposed RAP-Gen Framework

A. Retrieval-Augmented Generation

The overall RAP-Gen architecture is shown in Figure 4. The framework begins by embedding the input query x and retrieving external evidence before generation.

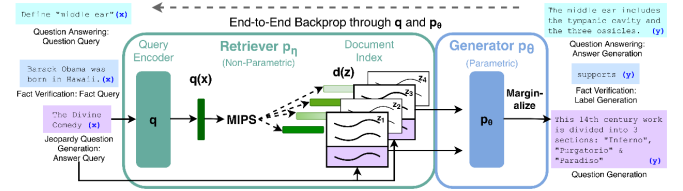


Figure 4: Retrieval-Augmented and Preference-Aligned Content Generation Framework

Let $q(x)$ denote the query embedding and $d(z)$ the embedding of document z . A dense retrieval distribution is defined by

$$p_{\eta}(z | x) = \frac{\exp(q(x)^{\top} d(z))}{\sum_{z'} \exp(q(x)^{\top} d(z'))}. \quad (6)$$

In practice, exact normalization over a very large collection is approximated through nearest-neighbor retrieval. The implementation uses FAISS-style maximum inner-product search; GPU-accelerated large-scale similarity search provides an efficient mechanism for this purpose [12].

Given retrieved candidates $\mathcal{Z}$, the generator marginalizes over evidence:

$$p(y | x) = \sum_{z \in \mathcal{Z}} p_{\eta}(z | x) p_{\theta}(y | x, z). \quad (7)$$

This formulation follows the core RAG principle of combining parametric generation with non-parametric retrieved memory [5].

B. Preference-Aligned Generation

For the same input x , let y^+ and y^- denote the preferred and less preferred candidates. Preference alignment uses

$$\mathcal{L}_{\text{pref}} = -\mathbb{E}_{(x, y^+, y^-, z)} [\log \sigma(R_{\phi}(y^+, x, z) - R_{\phi}(y^-, x, z))]. \quad (8)$$

The reward is explicitly oriented toward factual consistency in this study. This is narrower than general helpfulness or harmlessness objectives used in instruction-following alignment [6].

To reduce mismatch between evidence selection and generation, the framework includes

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{x, y} [D_{\text{KL}}(p_{\eta}(z | x) \| p_{\theta}(z | x, y))]. \quad (9)$$

The complete objective is

$$\mathcal{L}(\theta, \eta, \phi) = -\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\log \sum_{z \in \mathcal{Z}} p_{\eta}(z | x) p_{\theta}(y | x, z) \right] + \lambda_1 \mathcal{L}_{\text{pref}} + \lambda_2 \mathcal{L}_{\text{align}} + \lambda_3 \mathbb{E}_{x,z} \left[\|f_{\theta}(x, z) - g(z)\|_2^2 \right]. \quad (10)$$

The first term trains evidence-conditioned generation, the second introduces preference information, the third aligns retrieval and generation distributions, and the final term constrains semantic representations.

C. Bidirectional Encoding and Autoregressive Decoding

Figure 5 shows the encoder-decoder component.

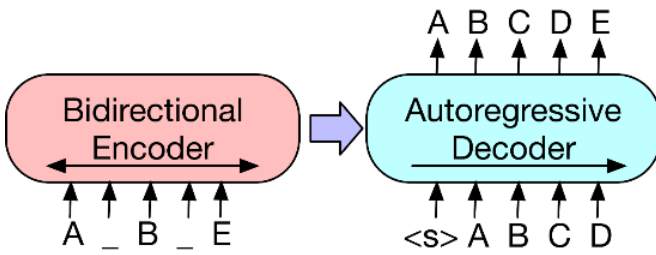


Figure 5: Bidirectional Encoder and Autoregressive Decoder Architecture for Content Generation

The bidirectional encoder produces contextual states

$$h_i = [\text{Encoder}(x_1, x_2, \dots, x_n)]_i, \quad (11)$$

and the autoregressive decoder factorizes generation as

$$p_{\theta}(y | x, z) = \prod_{t=1}^T p_{\theta}(y_t | y_{<t}, H, z). \quad (12)$$

Retrieved evidence is incorporated into the encoder or decoder context through attention. The combined representation is written as

$$H = \text{Encoder}(x, z). \quad (13)$$

Figure 6 provides a second schematic of conditional sequence generation and emphasizes stepwise autoregressive prediction.

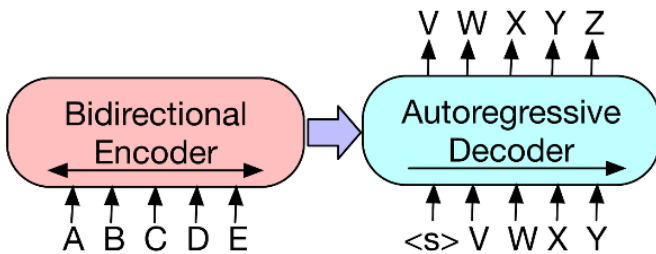


Figure 6: Conditional Sequence Generation With a Bidirectional Encoder and Autoregressive Decoder

With encoder states

$$H = \text{Encoder}(x, z) = \{h_1, h_2, \dots, h_n\}, \quad (14)$$

the sequence probability is

$$p_{\theta}(y | x, z) = \prod_{t=1}^T p_{\theta}(y_t | y_{<t}, H). \quad (15)$$

A reward-weighted distribution can then be written as

$$\tilde{p}(y | x, z) = \frac{p_{\theta}(y | x, z) \exp(\alpha R_{\phi}(y, x, z))}{\sum_{y'} p_{\theta}(y' | x, z) \exp(\alpha R_{\phi}(y', x, z))}, \quad (16)$$

where α controls the contribution of the factual-preference reward.

D. Cross-Domain Initialization

Figure 7 illustrates a two-stage encoding mechanism in which an initially untrained projection layer maps non-standard inputs into a representation space suitable for a pre-trained encoder.

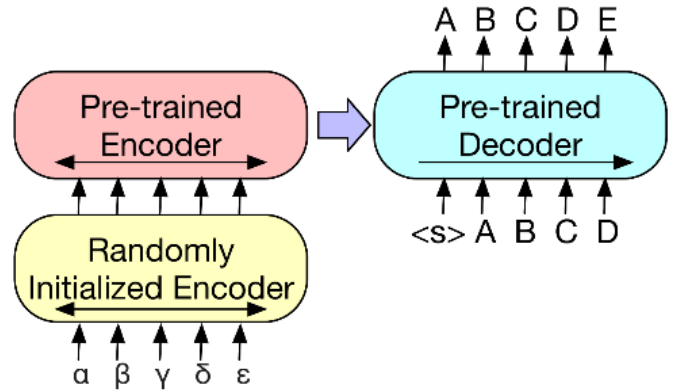


Figure 7: Pre-Trained Encoder-Decoder Architecture With Cross-Domain Initialization decoder architecture with cross-domain initialization

The initial projection is

$$\tilde{h} = E_{\text{rand}}(u), \quad (17)$$

followed by

$$H = E_{\text{pre}}(\tilde{h}). \quad (18)$$

The pre-trained decoder then generates

$$p_{\theta}(y | u, z) = \prod_{t=1}^T p_{\theta}(y_t | y_{<t}, H, z), \quad (19)$$

and preference reweighting gives

$$\tilde{p}(y | u, z) = \frac{p_{\theta}(y | u, z) \exp(\alpha R_{\phi}(y, u, z))}{\sum_{y'} p_{\theta}(y' | u, z) \exp(\alpha R_{\phi}(y', u, z))}. \quad (20)$$

The use of a bidirectional encoder with an autoregressive decoder is consistent with modern denoising sequence-to-sequence architectures such as BART [10].

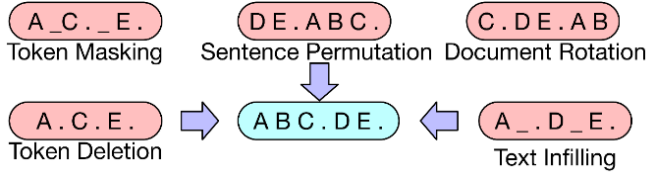


Figure 8: Denoising Pre-Training With Structured Corruption Strategies

E. Denoising Pre-Training

The structured corruption process is illustrated in Figure 8.

Given clean text x , a corrupted version is sampled from

$$\tilde{x} \sim q(\tilde{x} | x), \quad (21)$$

where the corruption family includes masking, deletion, text infilling, sentence permutation, and document-level transformations. The denoising objective is

$$\mathcal{L}_{\text{denoise}} = -\mathbb{E}_{x \sim \mathcal{D}, \tilde{x} \sim q(\tilde{x} | x)} [\log p_{\theta}(x | \tilde{x})]. \quad (22)$$

BART provides a direct precedent for this style of sequence-to-sequence denoising [10]. The retrieval-aware extension used here is

$$\begin{aligned} \mathcal{L}_{\text{denoise-aug}} = & -\mathbb{E} [\log p_{\theta}(x | \tilde{x}, z)] \\ & + \lambda \mathbb{E} [\|f_{\theta}(\tilde{x}, z) - g(z)\|_2^2]. \end{aligned} \quad (23)$$

This objective combines reconstruction with evidence-sensitive representation consistency.

IV. Experiments

A. Experimental Setup

The experimental design covers summarization, creative writing, and knowledge-intensive question answering. CNN/DailyMail is used for summarization; the dataset is a standard benchmark for abstractive summarization and was used in prior neural summarization work [13]. WritingPrompts provides approximately 300,000 prompt-story pairs and supports long-form story generation [14]. Natural Questions is used for knowledge-intensive question answering and provides real user queries paired with evidence from Wikipedia [15]. OpenWebText is used as general language-modeling data.

The preference component uses pairwise triples (x, y^+, y^-) derived from the factual annotation protocol. The retrieval index contains several million documents and is searched through FAISS-style maximum inner-product search [12]. The model is implemented with Transformer-based encoder and decoder components [1]. The supplied configuration reports a batch size of 32, a learning rate of 1×10^{-5} , and training on one NVIDIA RTX 2080 Ti GPU.

The baselines include GPT-2/GPT-3-style autoregressive generation, BART [10], T5 [11], standard RAG [5], and CTRL for controllable generation [16]. These baselines represent parametric generation, denoising sequence-to-sequence transfer, text-to-text transfer, retrieval augmentation, and conditional control.

B. Evaluation Metrics

Automatic evaluation uses BLEU, ROUGE, perplexity, BERTScore, and FactScore, while the manuscript also reports human ratings for fluency, coherence, and factuality. BERTScore evaluates similarity through contextual token representations rather than exact lexical overlap [17]. FactScore decomposes generated text into atomic factual claims and evaluates support for those claims, making it relevant to factual long-form generation [18].

Human evaluation is reported on five-point scales. For such ratings to be fully reproducible, future work should state the number of annotators, recruitment criteria, annotation instructions, blinding procedure, inter-annotator agreement, and whether significance testing was performed. These details are not included in the supplied manuscript.

C. Main Results

Table 1 reports the main comparison.

For summarization, RAP-Gen is reported with BLEU 28.6, ROUGE-1 44.1, ROUGE-2 21.2, ROUGE-L 41.0, BERTScore 0.902, perplexity 19.8, and FactScore 76.8. Relative to the RAG row in the same table, the ROUGE-L difference is 2.8 absolute points and the FactScore difference is 4.2 points. Human factuality is reported as 4.6/5.

For question answering, RAP-Gen is reported with BLEU 27.9, BERTScore 0.889, perplexity 21.6, and FactScore 75.0. The RAG row reports FactScore 70.5, giving an absolute difference of 4.5 points. These differences support the intended direction of the preference-alignment component, but statistical significance cannot be established from the aggregate table alone.

D. Ablation Study

Table 2 reports the ablation configurations.

Removing retrieval reduces QA FactScore from 75.0 to 68.8, an absolute decline of 6.2 points. Removing preference alignment reduces the reported human factuality rating from 4.6 to 3.3, a decline of 1.3 points. Removing denoising is evaluated on WritingPrompts and is associated with lower BLEU, ROUGE-L, BERTScore, and factuality than the full WritingPrompts configuration. The ablation pattern is consistent with complementary contributions from retrieval, preference alignment, and robustness training.

E. Number of Retrieved Documents

Figure 9 examines the effect of the retrieval depth K .

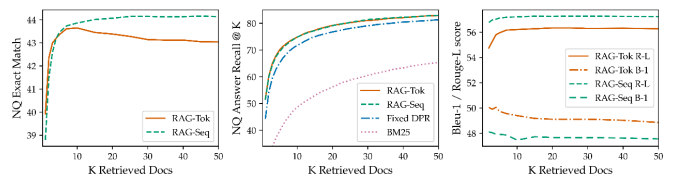


Figure 9: Effect of the Number of Retrieved Documents on Retrieval and Generation Performance

Table 1: Performance Comparison of Content-Generation Models on Summarization and Question Answering

Model	Task	BLEU ↑	ROUGE-1 ↑	ROUGE-2 ↑	ROUGE-L ↑	BERTScore ↑	PPL ↓	FactScore ↑	Fluency ↑	Coherence ↑	Factuality ↑
GPT-2 (Zero-shot)	Summarization	21.3	34.8	14.2	31.5	0.842	28.5	62.1	3.8	3.6	3.2
GPT-3 (Few-shot)	Summarization	24.7	38.9	17.5	35.6	0.861	24.2	66.3	4.2	4.0	3.8
BART	Summarization	25.9	40.2	18.6	36.8	0.874	22.7	68.9	4.3	4.2	4.0
T5	Summarization	26.4	41.0	19.1	37.5	0.879	21.9	69.5	4.4	4.3	4.1
RAG	Summarization	27.1	42.3	19.8	38.2	0.885	21.2	72.6	4.4	4.3	4.2
RAP-Gen (Ours)	Summarization	28.6	44.1	21.2	41.0	0.902	19.8	76.8	4.6	4.5	4.6
GPT-2 (Zero-shot)	QA	19.5	—	—	—	0.821	30.4	58.7	3.7	3.5	3.1
GPT-3 (Few-shot)	QA	23.8	—	—	—	0.846	25.8	63.9	4.1	3.9	3.6
BART	QA	24.6	—	—	—	0.853	24.1	65.8	4.2	4.0	3.8
T5	QA	25.1	—	—	—	0.861	23.5	67.2	4.3	4.1	4.0
RAG	QA	26.3	—	—	—	0.872	22.8	70.5	4.4	4.2	4.1
RAP-Gen (Ours)	QA	27.9	—	—	—	0.889	21.6	75.0	4.6	4.4	4.6

Table 2: Ablation Study of the RAP-Gen Components

Model Variant	Task	BLEU ↑	ROUGE-L ↑	BERTScore ↑	PPL ↓	FactScore ↑	Fluency ↑	Coherence ↑	Factuality ↑
w/o Retrieval	QA	25.4	—	0.861	23.9	68.8	4.3	4.1	4.0
w/o Preference	QA	26.7	—	0.874	22.5	72.9	4.4	4.2	3.3
w/o Denoising	WritingPrompts	23.1	34.2	0.852	26.8	71.5	4.1	3.9	3.8
Full Model (RAP-Gen)	QA	27.9	—	0.889	21.6	75.0	4.6	4.4	4.6
Full Model (RAP-Gen)	WritingPrompts	25.8	38.7	0.901	22.1	76.2	4.6	4.5	4.5

The supplied plot indicates rapid gains at small K followed by diminishing returns and, for some generation metrics, slight degradation when too many documents are introduced. This behavior is plausible because larger retrieval sets increase evidence coverage while also increasing noise and conflict. Standard RAG similarly distinguishes token-level and sequence-level evidence use [5].

The practical implication is that retrieval depth should be tuned jointly with evidence filtering rather than maximized. A small set of highly relevant passages may provide stronger grounding than a large set of weakly relevant documents.

F. Attention Over Retrieved Evidence

Figure 10 visualizes the contribution of retrieved documents during generation.

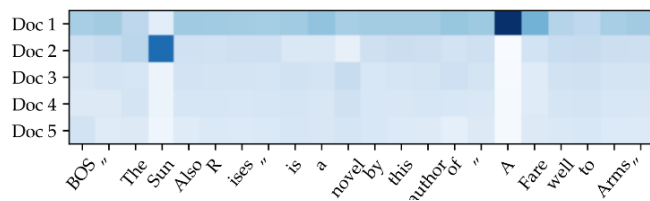


Figure 10: Attention Distribution Over Retrieved Documents During Generation

The heatmap shows non-uniform attention across retrieved documents and output tokens. Higher attention is concentrated on evidence that is described as relevant to entity names and factual fragments. This pattern is consistent with the goal of evidence-aware generation, but attention weights should not be interpreted automatically as causal explanations. A stronger evaluation would compare attention concentration with independently measured document relevance and factual support.

G. Robustness to Perturbed Input

Figure 11 compares model behavior under clean and perturbed inputs.

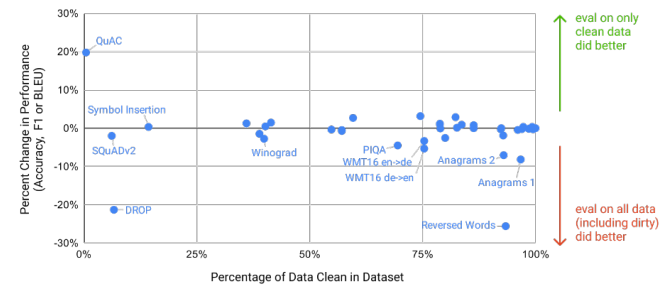


Figure 11: Performance Under Clean and Perturbed Input Conditions Across Tasks

The reported trend suggests that the denoising and retrieval components reduce degradation on corrupted inputs. This interpretation is consistent with the denoising motivation of BART-style pre-training [10]. The retrieval component may also compensate for missing or corrupted local context when the query still retains enough semantic information to locate useful evidence.

Robustness should nevertheless be reported with explicit perturbation severity, random seeds, and confidence intervals. Without these details, Figure 11 provides comparative evidence but not a complete robustness characterization.

H. Scaling Behavior

Figure 12 shows the reported relationship among validation loss, model size, and training compute.

Large neural language models often exhibit approximate power-law relationships between loss, parameter count, dataset size, and compute over substantial empirical ranges [19]. A recent research discusses power laws broadly in mathematical modeling and explicitly identifies neural scaling laws in artificial intelligence as an example of power-law behavior

[20]. That article is not a language-model scaling study, so it is used here only as a mathematical-modeling context for the power-law form, not as evidence for the specific coefficients shown in Figure 12.

The figure emphasizes the efficiency problem motivating RAP-Gen: scaling a parametric model can improve performance but increases training and inference cost. Retrieval and preference alignment seek to improve factual generation without relying exclusively on parameter growth.

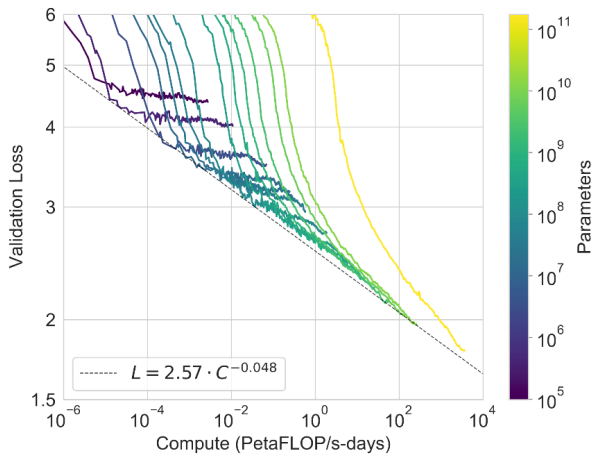


Figure 12: Scaling Behavior of Language Models With Respect to Compute and Model Size

I. In-Context Learning and Model Scale

Figure 13 compares ICL performance across demonstration counts and model scales.

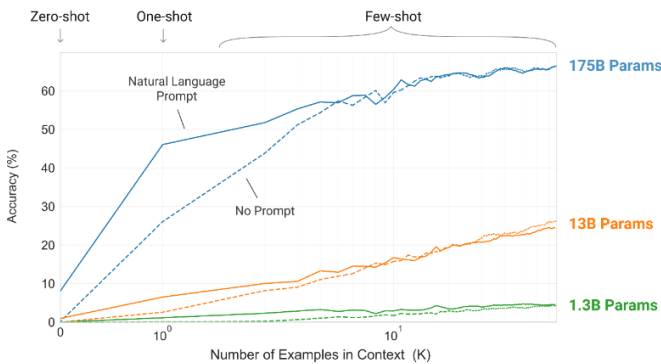


Figure 13: In-Context Learning Performance With Varying Numbers of Demonstrations and Model Sizes

The reported trend is consistent with the broad observation that larger language models display stronger few-shot adaptation [2]. However, ICL performance is not monotonic in every task, and it can depend strongly on example selection, prompt order, label semantics, and context length [9]. The figure should therefore be interpreted as a task-specific illustration rather than a universal scaling law.

Figure 14 shows training and validation loss across model sizes.

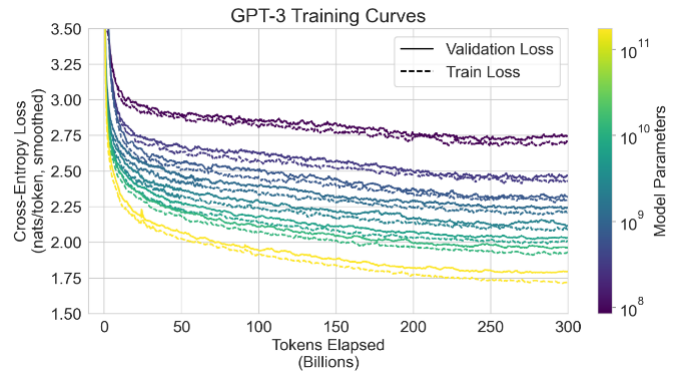


Figure 14: Training and Validation Loss Across Different Model Scales

The lower loss of larger models is consistent with empirical scaling studies [19]. The small reported gap between training and validation curves is suggestive of limited overfitting in the illustrated setup, although a full generalization analysis would require the underlying curves, seeds, and held-out data definition.

J. SuperGLUE and Multi-Task ICL

Figure 15 presents SuperGLUE performance as a function of model scale and context demonstrations.

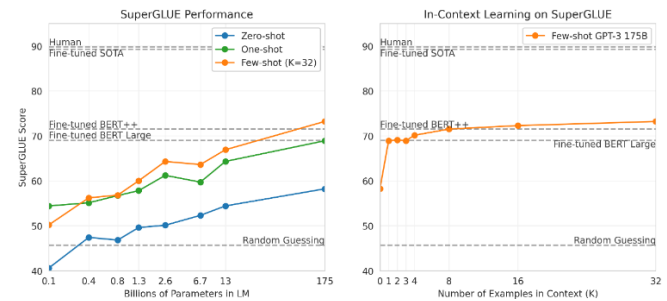


Figure 15: SuperGLUE Performance as a Function of Model Size and In-Context Examples

The figure is used to contextualize the role of model scale in zero-shot, one-shot, and few-shot task adaptation. GPT-3 established that sufficiently large autoregressive models can perform a wide range of tasks from natural-language instructions and demonstrations without conventional fine-tuning [2]. The diminishing gains at larger K values in the figure reinforce the need to select demonstrations carefully rather than simply increasing prompt length.

Figure 16 extends the comparison to translation, natural-language understanding, and structured reasoning.

The reported pattern shows stronger few-shot benefits at larger model sizes. This supports the use of ICL as the rapid adaptation component of RAP-Gen, while retrieval supplies external evidence and preference alignment constrains output selection.

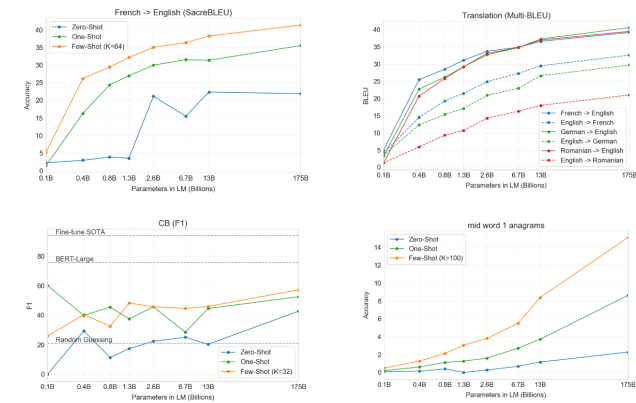


Figure 16: Multi-Task Performance Under Different Model Sizes and In-Context Learning Settings

K. Integrated Configuration Comparison

Figure 17 summarizes the overall performance trend across model configurations.

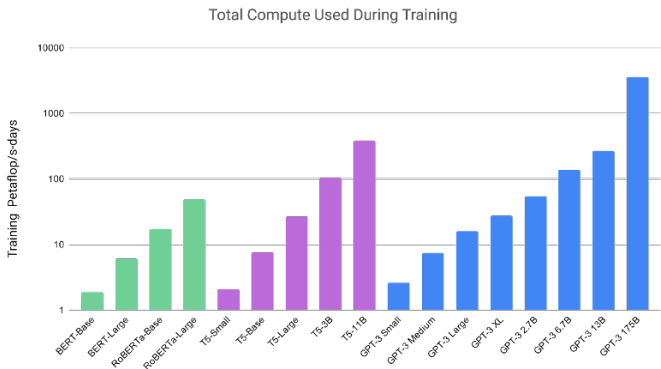


Figure 17: Overall Comparison Across Model Configurations

The figure presents a progression from a base model toward configurations that incorporate denoising, ICL, RAG, and preference alignment. The interpretation is that the components operate at different levels: denoising strengthens representation robustness, ICL supports inference-time adaptation, retrieval supplies external knowledge, and preference alignment biases generation toward preferred factual behavior. This modular interpretation is more defensible than attributing all improvement to a single mechanism.

V. Discussion

The experimental pattern supports the architectural motivation for RAP-Gen, but several issues require careful interpretation. First, retrieval improves access to information but does not guarantee factual correctness. Retrieval can return irrelevant, outdated, or conflicting passages. A reliable production system therefore requires source quality control, retrieval confidence, provenance tracking, and explicit citation or evidence inspection. The factuality literature increasingly treats grounding and verification as distinct from fluency [7], [8].

Second, preference alignment depends on the quality of human judgments. If annotators confuse plausibility with truth, the reward model can reinforce polished misinformation. The factual preference protocol should therefore report evidence sources, adjudication procedures, annotator agreement, and policies for ambiguous or time-sensitive claims. RLHF research demonstrates the utility of preference data but also makes clear that the model is learning the evaluators' operationalized preferences rather than an abstract universal objective [6].

Third, the framework uses several different notions of "alignment": retrieval-generation distribution alignment, representation alignment with evidence, and human-preference alignment. These mechanisms should remain conceptually separate. Distribution alignment constrains model components mathematically, while preference alignment supplies a normative or evaluative signal. Conflating them can make ablation results difficult to interpret.

Fourth, the computational-cost argument should be tested directly. Retrieval adds index construction, embedding, search, and context-processing costs. Preference modeling adds reward-model training or scoring. RAP-Gen can be more parameter-efficient than scaling the base model, but that does not automatically imply lower total system cost. Future work should report latency, memory consumption, index size, retrieval throughput, token overhead, and energy consumption.

Fifth, the reported numerical results require stronger reproducibility support. The manuscript gives a batch size and learning rate, but does not provide complete architecture sizes, retriever checkpoints, optimizer parameters, number of epochs, decoding strategy, beam size, random seeds, prompt templates, preference-dataset size, annotator count, or statistical uncertainty. These details are necessary before the claimed improvements can be independently reproduced.

The framework is nevertheless conceptually useful for content creation because it addresses four distinct failure modes. Retrieval addresses missing or stale parametric knowledge; ICL addresses rapid task adaptation; denoising addresses input corruption; and preference alignment addresses output selection according to human judgments. The strongest future version of RAP-Gen would treat these components as independently measurable subsystems and report both quality and cost.

A. Additional Reproducibility and Deployment Considerations

A production-oriented evaluation should distinguish offline benchmark performance from live-system behavior. Retrieval latency depends on index size, embedding dimensionality, hardware, approximate-nearest-neighbor parameters, and the number of retrieved passages. Generation latency depends on model size, context length, decoding strategy, and output length. Human-preference scoring introduces additional cost when a separate reward model is used at inference time. These components should therefore be benchmarked separately before reporting an end-to-end efficiency advantage.

The retrieval corpus also requires version control. If documents change between training and evaluation, the same query may produce different evidence and different output. Reproducibility therefore requires recording the document snapshot, encoder checkpoint, index parameters, and retrieval depth. For time-sensitive content creation, corpus freshness becomes part of model quality rather than a simple infrastructure detail.

Preference data require equally careful documentation. Annotators should receive a written definition of factual consistency, instructions for uncertain cases, and a method for consulting authoritative evidence. Agreement should be measured before using labels to train a reward model. Cases involving disputed claims, rapidly changing facts, or ambiguous wording may require adjudication rather than simple majority voting.

Finally, factuality metrics should be triangulated. Lexical-overlap scores such as BLEU and ROUGE primarily measure surface similarity and do not directly establish truthfulness. BERTScore improves semantic similarity measurement, while FactScore is more explicitly oriented toward factual precision. Human verification remains important for claims that cannot be resolved reliably by automatic metrics. A strong evaluation should therefore report language quality, semantic similarity, evidence support, and human factuality as distinct outcomes.

B. Implications for Practical Content-Creation Systems

The proposed architecture has implications for several classes of content-creation applications. In automated news drafting, the retrieval component can supply current facts, names, dates, numerical values, and source material that may not be reliably represented in the model parameters. Preference alignment can then be used to discourage unsupported elaboration and favor outputs that remain close to retrieved evidence. In such a setting, the system should preserve provenance at the sentence or claim level so that editors can inspect which retrieved passages influenced the generated text. This requirement is especially important because factual correctness in news cannot be inferred from fluency alone.

In intelligent customer service, retrieval can connect generation to product manuals, policy documents, frequently asked questions, and account-independent service rules. Preference signals can encode requirements such as directness, politeness, factual restraint, and escalation when the available evidence is insufficient. The role of RAP-Gen in this context is therefore not simply to produce more natural language, but to mediate between a dynamic document collection and a conversational interface. Retrieval confidence should be exposed to the downstream system so that uncertain queries can be transferred to a human agent rather than answered speculatively.

Advertising and marketing copy represent a different control problem. Here factuality remains important, particularly for product claims, prices, or regulatory statements, but stylistic control also becomes central. In-context demonstrations can define tone, vocabulary, audience, length, and brand voice without requiring a separate fine-tuned model for every cam-

paign. Preference alignment can be extended from factuality to multi-dimensional rewards covering style consistency, prohibited claims, and brand constraints. However, reward dimensions should be reported separately because a single scalar reward may conceal trade-offs between factual accuracy and stylistic attractiveness.

Academic and technical writing assistance creates an even stricter evidence requirement. Retrieval should favor primary literature, authoritative databases, or documents supplied by the user. The generator should distinguish retrieved statements from its own synthesis and should avoid inventing bibliographic metadata. The factual-preference framework could be extended by asking annotators to judge whether individual technical claims are entailed by cited evidence. Such claim-level grounding would make the framework more useful for scholarly writing than a generic preference score.

Creative writing presents the opposite challenge: factual grounding may be less important than narrative coherence, originality, and adherence to a prompt. WritingPrompts is included in the experiments precisely because it tests longer-form generation where retrieval is not always necessary. In this domain, the retrieval module can be disabled or repurposed to retrieve stylistic examples, world-building notes, or user-provided canon. Preference alignment can then represent narrative constraints rather than factual truth. This illustrates an important design principle: RAP-Gen is best viewed as a modular framework whose reward and retrieval objectives should be adapted to the content-creation task.

C. Retrieval Quality, Evidence Selection, and Noise

The quality of retrieval is a major determinant of downstream generation. Dense retrieval maps a query and documents into a shared vector space and selects documents with high similarity. This approach is effective when semantic correspondence is not captured by exact lexical overlap, but it can also retrieve documents that are topically similar while failing to answer the specific question. A robust system should therefore measure not only retrieval recall but also evidence precision.

The retrieval-depth experiment in Figure 9 is consistent with this trade-off. Small values of K may omit useful evidence, whereas very large values of K increase the probability of redundant, weakly relevant, or contradictory material. The resulting context can exceed the generator’s ability to distinguish central evidence from distractors. Preference alignment may reduce some downstream errors, but it cannot fully repair a retrieval stage that consistently omits the correct evidence.

A practical implementation can introduce a second-stage reranker after approximate nearest-neighbor search. The first-stage retriever maximizes recall by returning a moderately large candidate set. A cross-encoder or task-specific reranker then scores query–document pairs more precisely and reduces the final context set. Document-level filtering can also remove duplicate passages, low-quality sources, and content that fails freshness or authority criteria. These operations would make the retrieval component more selective than the single distribution in Equation 6 suggests.

Evidence conflict requires a separate strategy. Two highly ranked documents may disagree because one is outdated, because sources use different definitions, or because the underlying issue is genuinely contested. The generator should not collapse such evidence into a single unsupported assertion. One option is to detect contradiction before generation and explicitly represent the disagreement in the prompt. Another is to condition the generator on source metadata such as publication date, authority, or document type. For high-stakes use, conflicting evidence should trigger abstention or human review.

Retrieval evaluation should also be task-specific. For Natural Questions, answer recall and exact match are natural metrics. For summarization, the relevant question is whether retrieved material supplies evidence that supports the summary rather than merely overlapping with the input. For creative writing, retrieval relevance may concern thematic or stylistic compatibility. A single retrieval metric is therefore unlikely to characterize the framework adequately across all content-creation tasks.

D. Preference Modeling and Human Annotation Quality

The factual-preference component is conceptually central because it supplies a human judgment signal that is not available from maximum-likelihood training alone. However, the quality of a reward model depends directly on the annotation process. Pairwise comparison reduces cognitive burden, but factuality can still be difficult to judge when statements contain multiple claims or when the evidence is incomplete.

The annotation interface in Figure 1 should therefore be accompanied by a formal protocol. Annotators should be instructed to separate factual correctness from writing quality. A grammatically polished response should not be preferred if it contains unsupported facts, and an awkward but factually correct response should not be downgraded solely because of style. If both candidate responses are partially correct, the task should allow an “uncertain” or “mixed” label rather than forcing a binary judgment.

Complex outputs can be decomposed into atomic claims before preference labeling. This would align the annotation process more closely with claim-level factuality evaluation. For example, a generated paragraph about a historical figure may contain separate claims about birth date, publications, awards, and relationships. A pairwise paragraph-level preference can hide the fact that each candidate is correct on different subsets of these claims. Atomic verification would provide a more informative supervision signal.

The annotator pool also matters. General factual questions may be evaluated by trained non-specialists using reliable sources, whereas scientific, legal, medical, or financial content may require domain expertise. The manuscript should therefore avoid presenting “human preference” as a homogeneous ground truth. It is a measured judgment produced under a particular annotation protocol.

Reward models can also overfit annotation artifacts. If preferred responses are consistently longer, more formal, or more

heavily hedged, the reward model may learn those superficial correlates rather than factuality. Balanced candidate construction and adversarial validation are useful defenses. A strong evaluation would test whether the reward model continues to prefer factual outputs when length, tone, and formatting are controlled.

E. In-Context Learning as a Control Mechanism

ICL is used in RAP-Gen as an inference-time adaptation mechanism. This choice has practical advantages because task-specific behavior can be modified without storing separate model checkpoints. A content-creation system can therefore support multiple styles, output schemas, and editorial policies using demonstrations.

The effectiveness of ICL depends on example quality. A few high-quality demonstrations that clearly represent the desired mapping may be more useful than a larger number of noisy examples. Demonstrations should be selected for relevance to the current input, especially when the task has heterogeneous subtypes. Retrieval itself can be used to select demonstrations from a library of validated examples.

Ordering effects are another concern. Large language models can be sensitive to the order in which demonstrations are presented. The first or last examples may receive disproportionate influence, and label distributions within the context can alter predictions. Reproducible experiments should therefore report the demonstration-selection algorithm and ordering rule.

ICL also competes with retrieved evidence for context-window capacity. A prompt containing many demonstrations leaves less space for retrieved documents and the generated output. RAP-Gen therefore requires a context-budget policy that balances examples, evidence, instructions, and user input. Dynamic allocation is preferable to a fixed prompt template because different tasks require different amounts of evidence.

The scaling figures in the manuscript suggest that larger models benefit more strongly from few-shot demonstrations. This is consistent with the original GPT-3 observations, but it should not be interpreted as proof that a larger model is always the most efficient choice. A smaller model with targeted retrieval and a carefully designed prompt may achieve adequate performance at substantially lower computational cost. This is precisely the system-level trade-off that motivates combining parametric and non-parametric resources.

F. Denoising and Robustness Beyond Synthetic Corruption

Structured denoising is intended to improve robustness to incomplete or perturbed input. Token masking, deletion, span infilling, sentence permutation, and document rotation create controlled noise during training. These corruptions encourage the model to reconstruct semantic structure rather than memorize exact surface sequences.

Real deployment noise can differ substantially from synthetic corruption. User prompts may contain typographical errors, copied fragments, contradictory instructions, truncated

documents, multilingual code-switching, OCR errors, or malformed structured data. A robust evaluation should therefore include naturally occurring noise in addition to artificial perturbations.

Retrieval can help in some noisy conditions because a partially corrupted query may still contain enough semantic information to identify relevant documents. In other cases, noise can cause the retriever to fail before the generator has a chance to use external evidence. Query rewriting or robust query encoding may therefore be necessary. One extension would generate several normalized query candidates and aggregate retrieval results before reranking.

Robustness also includes adversarial behavior. A retrieved document can contain misleading instructions or prompt-injection text, especially when the corpus includes web content. A production RAG system should treat retrieved text as data rather than executable instructions. Filtering, source allowlists, structured prompting, and separation between control instructions and retrieved evidence are important engineering safeguards.

G. Interpreting the Reported Evaluation Metrics

The experimental section reports a large collection of automatic and human metrics. Each metric captures a different aspect of quality and has limitations. BLEU and ROUGE are reference-based overlap metrics. They can reward lexical correspondence but may penalize valid paraphrases. High overlap also does not guarantee factual correctness.

BERTScore addresses some of these limitations by comparing contextual token representations, which allows semantically similar paraphrases to receive higher scores than exact-match metrics would assign [17]. Nevertheless, semantic similarity remains distinct from factual support. A generated statement can be semantically similar to a reference and still contain an incorrect number or named entity.

Perplexity measures how well a model predicts a sequence under its probability distribution. Lower perplexity can reflect better language modeling, but it does not directly measure usefulness, factuality, or adherence to instructions. Comparisons are also difficult when tokenizers or model vocabularies differ.

FactScore is more closely aligned with the central claim of RAP-Gen because it focuses on atomic factual precision [18]. Even so, factuality measurement depends on the quality of the knowledge source and the accuracy of the evaluator used to verify claims. Human review remains important for ambiguous or domain-specific statements.

The manuscript also reports fluency, coherence, and factuality ratings. These dimensions should be analyzed separately rather than averaged into one human score. A system can be fluent but factually weak, or factual but stylistically poor. Reporting the dimensions independently makes the trade-offs visible.

H. System Efficiency and the Limits of Model Scaling

Figures 12–16 emphasize model scale, compute, and ICL. Scaling laws demonstrate that increasing parameters, data, and

compute can reduce language-model loss over broad empirical ranges [19]. The related discussion of power laws provides a broader mathematical perspective on this functional behavior [20].

However, end-to-end content-generation efficiency depends on more than base-model size. Retrieval adds document encoding, index memory, nearest-neighbor search, and additional context tokens. Preference alignment may require a reward model or rejection-sampling stage. Long retrieved contexts increase attention cost. A fair comparison between RAP-Gen and a larger parametric model should therefore report total inference latency and hardware consumption.

Caching can reduce retrieval cost when many requests concern similar topics. Document embeddings can be computed offline, and frequently retrieved passages can remain in high-speed storage. Query embeddings and retrieval results can also be cached under appropriate privacy controls. These engineering choices may make a retrieval-augmented system more efficient than its conceptual architecture suggests.

A system may also route requests among models of different sizes. Simple requests can be answered by a smaller model, while complex or uncertain cases are escalated. Retrieval confidence, estimated task difficulty, and factual-risk level can inform this routing. Such adaptive computation is a promising direction for reducing average cost without reducing reliability.

I. Reproducibility and Experimental Validity

The supplied manuscript reports extensive numerical results but does not provide the full information needed for independent reproduction. This is the most important limitation of the current experimental presentation.

For model training, the final paper should report the exact base checkpoints, parameter counts, tokenizer, maximum sequence lengths, optimizer, learning-rate schedule, number of epochs or update steps, gradient accumulation, weight decay, dropout, warm-up strategy, mixed-precision configuration, and random seeds. “Transformer-based” is not sufficiently specific because architectures can differ substantially.

For retrieval, the document corpus should be identified precisely. The phrase “several million documents” should be replaced with an exact count, source, snapshot date, segmentation policy, embedding model, index type, similarity metric, and FAISS configuration. Retrieval metrics should be reported independently of generation metrics.

For the preference dataset, the manuscript should state the number of prompts, number of pairwise comparisons, number and qualifications of annotators, annotation interface, external evidence policy, disagreement resolution method, and train/validation/test split. Inter-annotator agreement should be reported.

For human evaluation, the sampling method for outputs should be documented. If the same annotators who created preference data evaluate the final model, this potential dependency should be disclosed. Statistical uncertainty should be reported for both automatic and human metrics.

For ablation experiments, only one component should be changed at a time while all other settings remain fixed. The current table largely follows this logic, but detailed configurations are necessary to confirm comparability. Repeated runs would allow mean and standard deviation to be reported rather than single values.

J. Limitations and Future Research

RAP-Gen integrates several well-established ideas, so novelty should be framed at the level of system integration and factual-preference design rather than implying that retrieval, ICL, denoising, or human-feedback alignment are individually new. RAG originates in prior retrieval-augmented generation work [5]; RLHF is established in instruction-following alignment [6]; ICL is a central property of large autoregressive models [2]; and denoising sequence-to-sequence pre-training is established by BART and related models [10]. The manuscript's contribution is the proposed coordination of these mechanisms for content creation.

A second limitation is evaluation breadth. Summarization, QA, and WritingPrompts cover useful but incomplete aspects of content creation. Future evaluation should include citation-grounded writing, multi-document synthesis, long-form report generation, dialogue, domain-specific technical writing, and time-sensitive factual tasks.

A third limitation is multilinguality. Many production content systems operate across languages, but the current experiments are predominantly English-centered. Retrieval quality, factuality evaluation, tokenization, and preference annotation can all vary by language. Multilingual retrieval and cross-lingual evidence alignment should therefore be tested explicitly.

A fourth limitation is temporal knowledge. Retrieval is especially valuable when facts change after pre-training, but the experiments do not isolate temporal updating as a variable. A future benchmark could contain questions whose answers change over time and evaluate whether refreshed retrieval indexes improve accuracy without retraining the base model.

Finally, deployment requires governance. Content-generation systems should log evidence, model versions, retrieval results, and human overrides. High-risk domains may require mandatory source display or human approval. Preference alignment can improve average behavior, but it does not eliminate the need for accountable review processes.

VI. Conclusion

This study presents RAP-Gen, a unified content-generation framework integrating retrieval-augmented generation, factual preference alignment, in-context learning, and denoising pre-training. The framework uses external evidence to complement parametric knowledge, pairwise preference supervision to guide factual output selection, contextual demonstrations to support rapid task adaptation, and structured corruption to improve robustness.

The reported summarization and question-answering results show higher automatic and human-evaluation scores

for RAP-Gen than for the listed baselines. Ablation results indicate that retrieval, preference alignment, and denoising each contribute to the final performance. Additional analyses illustrate retrieval-depth trade-offs, non-uniform attention over evidence, robustness under perturbation, language-model scaling, and in-context learning behavior.

At the same time, the results should be interpreted within the limits of the supplied evidence. The manuscript does not provide raw generations, code, complete configuration files, repeated-run uncertainty, or full annotation documentation. Future work should release these materials, evaluate retrieval faithfulness directly, compare preference methods under controlled conditions, and report total system cost alongside generation quality.

For practical content creation, the central implication is that model scale alone is not sufficient. Reliable generation requires access to external evidence, explicit mechanisms for selecting among outputs, robust representations, and transparent evaluation of factual support. RAP-Gen provides one integrated design for combining these capabilities.

Author Contributions

All authors contributed equally to this paper. All authors have read and approved the final manuscript.

Funding

No specific external funding was received for this study.

Data Availability

The datasets used in this study are available from the corresponding author upon reasonable request.

Ethics Statement

Informed consent was obtained from all participants involved in the study.

Conflict of Interest

The authors declare no conflicts of interest.

References

- [1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* 30, 5998–6008.
- [2] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., ..., & Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* 33, 1877–1901.
- [3] Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P. S., & Sun, L. (2025). A survey of AI-generated content (AIGC). *ACM Computing Surveys*, 57(5), Article 125, 1–38.
- [4] Hagos, D. H., Battle, R., & Rawat, D. B. (2024). Recent advances in generative AI and large language models: Current status, challenges, and perspectives. *IEEE Transactions on Artificial Intelligence*, 5(12), 5873–5893.
- [5] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* 33, 9459–9474.
- [6] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., ..., & Lowe, R. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems* 35, 27730–27744.

- [7] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12), Article 248, 1–38.
- [8] Augenstein, I., Baldwin, T., Cha, M., Chakraborty, T., Ciampaglia, ..., & Zagni, G. (2024). Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8), 852–863.
- [9] Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., ..., & Sui, Z. (2024). A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 1107–1128.
- [10] 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.
- [11] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Machine Learning Research*, 21(140), 1–67.
- [12] Johnson, J., Douze, M., & Jégou, H. (2021). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547.
- [13] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1073–1083.
- [14] Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 889–898.
- [15] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., ..., & Petrov, S. (2019). Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7, 452–466.
- [16] Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., & Socher, R. (2019). CTRL: A conditional Transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.
- [17] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.
- [18] Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12076–12100.
- [19] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- [20] Cojocaru, A. V., & Balint, S. (2026). Are power laws similar to constitutive laws? A critical analysis of fractional order derivatives in mathematical modeling. *TK TechForum Journal (ThyssenKrupp Techforum)*, 2026(2), 1–31.

...