WIKIGENBENCH: Exploring Full-length Wikipedia Generation for New Events

Anonymous ACL submission

Abstract

Generating comprehensive and accurate Wikipedia articles for newly emerging real-world events presents significant challenges. Previous efforts have often fallen short by focusing only on short snippets, neglecting verifiability, or ignoring the impact of the pre-training corpus. In this paper, we simulate a real-world scenario where structured, full-length Wikipedia articles with citations are generated for new events using input documents from web sources. To minimize data leakage in Large Language Models (LLMs), we select recent events and construct a new benchmark, WIKIGENBENCH, consisting of 1320 events paired with their corresponding related web documents. We also design a comprehensive set of systematic metrics and LLM-based baseline methods to evaluate the capability of LLMs in generating factual, full-length Wikipedia articles. The data and code will be released upon acceptance.

1 Introduction

Wikipedia serves as an essential resource for in-depth and trustworthy summaries of a wide range of subjects (Lemmerich et al., 2019), supporting many knowledge-intensive NLP tasks like information retrieval (Lehmann et al., 2015; Sharma et al., 2024), question answering (Chen et al., 2017; Yang et al., 2018), and automatic summarization (Liu et al., 2018). Wikipedia's accuracy, relevance, credibility, and completeness are ensured by volunteer editors. However, the exponential growth of internet information (Raffel et al., 2019; Biderman et al., 2022) makes manual curation challenging, as it struggles to keep up with new events and dispersed sources[1]. Thus, efficient and reliable automatic generation of high-quality Wikipedia articles is crucial yet underexplored (Huschens et al., 2023).

Many efforts have focused on generating Wikipedia articles. Early work often treated this task as a multi-document summarization challenge, using web documents as a static source and employing traditional IE and ML techniques to create Wikipedia articles (Sauper and Barzilay, 2009; Liu et al., 2018; Perez-Beltrachini et al., 2019; Banerjee and Mitra, 2016). With the advent of pre-trained language models (PLMs), despite their limitations in generating long texts, Wikipedia generation is approaching real-world scenarios, emphasizing document structure and verifiability. Fan and Gardent (2022) designed a retrieval mechanism to identify relevant supporting information from the web and used BART to generate long-form biographies section by section, given predefined section headings. Qian et al. (2023) explored the reliability of generated articles by considering supporting evidence. LLMs offer new possibilities for long text generation, but their use in Wikipedia generation is still preliminary. For instance, Shao et al. (2024) utilized proprietary LLMs to generate Wikipedia articles on a relatively small dataset and mainly focuses on the pre-writing stage.

Despite advances in Wikipedia article generation, several critical research problems need further exploration. First, it is essential to define the full-length Wikipedia generation task in the LLM era for real-world scenarios, considering model pre-exposure and retrieval verifiability to mitigate their impact. Second, comparative studies are needed to evaluate the effectiveness of retrieval-augmented models with consideration of the influence of various factors including base models, reranking techniques, number of retrieved documents and so on. Finally, the complexity of the task—being long, knowledge-intensive, and open-ended—makes evaluation challenging, requiring a comprehensive set of metrics to effectively assess the quality of generated Wikipedia articles.

To address these problems, we formalize the Wikipedia generation task and create the benchmark WIKIGENBENCH, carefully considering

[1] https://en.wikipedia.org/wiki/Wikipedia:Statistics

practical factors such as the selection of Wikipedia entries, model pre-exposure, and the collection of related documents. We focus mainly on new events that occurred after the knowledge cutoff date of our main experimental model (Ouyang et al., 2022; Touvron et al., 2023), mitigating model pre-exposure effects and ensuring that generation relies primarily on related documents. Other types of entries like celebrities, which have extensive historical records seen by LLMs, are excluded to avoid skewed performance. Unlike previous datasets that consist of either large-scale short snippets (Qian et al., 2023; Liu et al., 2018) or small-scale full-length articles (Shao et al., 2024; Banerjee and Mitra, 2015), our benchmark aims to construct a medium-scale dataset for systematically evaluating real-world Wikipedia generation. We meticulously curate a collection of related documents for each entry using a search engine to minimize the influence of document variability and ensure an equal footing for generation. This approach allows us to focus on the generation capabilities of LLMs based on the same set of related documents, without relying on the retrieval capabilities of various search engines.

To investigate LLM capabilities in Wikipedia generation, we develop baseline models under the Retrieval-Augmented Generation (RAG) framework (Lewis et al., 2020; Izacard and Grave, 2020; Hu et al., 2023). Our goal is to use state-of-the-art RAG techniques to retrieve important information for Wikipedia generation (Gao et al., 2023c; Ma et al., 2023; Shao et al., 2023). With this aim, We develop three methods: a naive RAG approach called Retrieve-then-Read (**RR**), an advanced RAG method called Plan-Retrieve-Read (**PRR**), and a finetuned RR model (**TunedRR**). RR reranks related documents and reads the top ones for generation, while PRR uses LLMs' planning capabilities and a multi-stage reranking strategy to outline and generate articles section by section. TunedRR employs a fine-tuning strategy for Wikipedia generation. For evaluation, we compile a comprehensive set of metrics focusing on writing, informativeness, and verifiability to assess Wikipedia articles' quality. Our work provides the first systematic comparison of LLM-based methodologies for full-length Wikipedia generation, offering valuable insights and highlighting the potential of combining retrieval techniques with LLM-based generation models to improve Wikipedia generation quality.

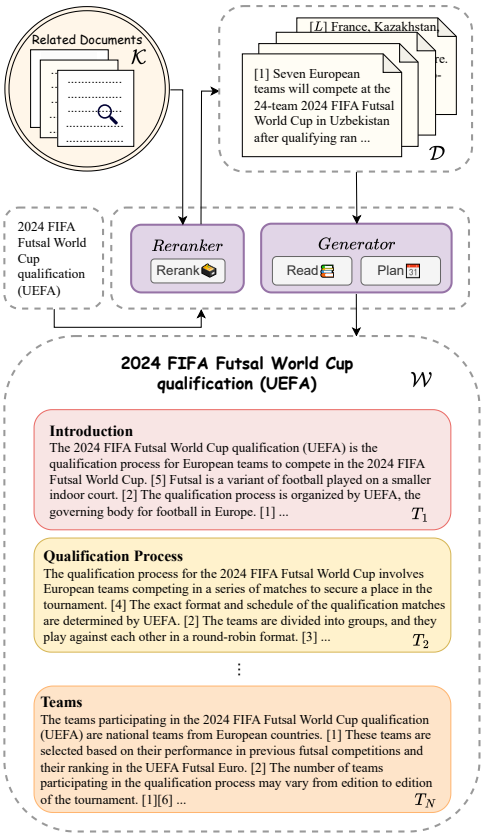

Figure 1: Illustration of the proposed Wikipedia generation task.

2 Full-length Wikipedia Generation Benchmark

2.1 Task Formalization

Formally, given an event and its related documents $\mathcal{K}$, pre-collected through search engine or manually extracted by human editors from a large external corpus (the internet) , this task aims to generate a full-length Wikipedia article $\mathcal{W}$ with N sections $\{T_1, T_2, ..., T_N\}$ and M sentences $\{S_1, S_2, ..., S_M\}$, as shown in Figure 1.

$$\mathcal{D} = \{D_1, ..., D_L\} = \textit{Reranker}(\mathcal{K}) \quad (1)$$

$$\mathcal{W} = T_1 \cup T_2 \cup \ldots \cup T_N = \textit{Generator}(\mathcal{D}) \quad (2)$$

$$T_i = \{< S_1, C_1 >, \ldots, < S_{M_i}, C_{M_i} >\}$$

$$|\mathcal{W}| = M = \sum_{i=1}^{N} M_i \quad (3)$$

T_i denotes the i section and is composed of M_i sentences. Each generated sentence S_j belongs to a specific section of the article (e.g., T_i), maintaining a clear structure. To ensure verifiability, every

sentence S_j is accompanied by its corresponding citation $C_j \subset \mathcal{K}_{\mathbf{E}}$. This definition ensures the generated Wikipedia article is coherent, well-organized, and substantiated with verifiable sources.

To generate a well-written and structured full-length Wikipedia article, the process usually involves two main components: a *reranker* and a *generator*. The LLM-based generator, either directly invoked or fine-tuned, reads the reranked set of documents $\mathcal{D}$ and generates the article. The reranker reorders the documents according to the generator's requirements, prioritizing the most relevant information.

2.2 Dataset Construction

To achieve our task, we collect Wikipedia entries, including articles, section outlines, and related document from human editors. During dataset construction, we select the newest possible Wikipedia entries to mitigate potential training data leakage. Additionally, we maintain entries with word counts between 1000 to 3000 to align with typical Wikipedia article lengths and filter out low-resource entries that may not serve as good evaluation cases. This approach helps ensure the dataset's relevance and quality.

Due to the high failure rate of Wikipedia's own links, we also utilize Google's search API to extract relevant web pages based on article titles, obtaining related documents from both human editors and search engines for comparison. We focus on Wikipedia entries about events, creating a dataset of 1,320 entries. Using the latest events minimizes the likelihood that the model has been trained on related Wikipedia data, crucial for evaluating the model's ability to present factual information. We select 309 entries as the test set and use the remaining entries for training (Appendix B.3).

2.3 Dataset Statistics

Table 1 outlines the statistics of the WIKIGENBENCH dataset, which totally consists of 107 million words across 1,320 Wikipedia entries and 53k related documents. Wikipedia articles in the dataset have an average of 1,665 words and 5.97 sections. Related documents curated by humans average 13k words, whereas documents retrieved via search engine average 24k words. These related documents substantially cover the information and knowledge about the events, making them suitable input for automatic generation of Wikipedia articles. More details about the dataset and the collection process can be found in Appendix B.

Source	Dataset Statistics	
Reference Wikipedia	Sections (avg.)	5.97
	Word count (avg.)	1665.51
Related documents by human editor	# Related docs (avg.)	17.49
	Word count (avg.)	13k
Related documents by search engine	# Related docs (avg.)	24.12
	Word count (avg.)	24k
Reference Wikipedia + Related documents	Events	1320
	# Related docs	55k
	Word count	107M

Table 1: Statistics of WIKIGENBENCH dataset. We report the scale of Wikipedia reference articles and related documents.

3 Evaluation Metrics

For the evaluation of Wikipedia articles, Wikipedia's assessment criteria[2] are significant but predominantly descriptive and lack quantitative measures. Manual evaluation of large-scale test data is costly and inherently subjective. Previous work has used automated metrics focusing on fluency, informativeness, and faithfulness. We adjust the dimensions to writing, informativeness, and verifiability, referencing Wikipedia's criteria. Recent studies (Sottana et al., 2023; Chiang and Lee, 2023; Lin and Chen, 2023a) have demonstrated the effectiveness of using LLMs as evaluators. In our evaluation, we design LLM-based metrics with appropriate prompts, specifically utilizing GPT4 and Prometheus2 (Kim et al., 2024), a 7B LLM-based evaluator. Metrics are rated on a 0-5 scale, with detailed prompts provided in Appendix D. In writing, we incorporate fluency and extend to measure outlining and organization. For informativeness, we measure content coverage and focus, and include n-gram-based metrics like ROUGE, METEOR. Verifiability, different from factualness, mainly measures the models' citation abilities and is a key feature in Wikipedia evaluation. Compared to previous work (Appendix A), we have developed a more comprehensive set of automatic evaluation metrics. Next, we present the metrics for each dimension.

Writing We design three metrics to evaluate Wikipedia articles: *Fluency*, *Organization*, and

[2]https://en.wikipedia.org/wiki/Wikipedia:Assessing_articles

Outline Scores. The *Fluency Score* assesses fluency and readability, the *Organization Score* evaluates structure and logical connections, and the *Outline Score* checks section heading quality. We utilize Prometheus2 to assess the *Organization Score* (Shao et al., 2024), and GPT4 to assesse the *Fluency Score* and *Outline Score*.

Informativeness We compile n-gram-based metrics including METEOR (Banerjee and Lavie, 2005), ROUGE-L (Lin, 2004), and BLEU (Papineni et al., 2002), which are widely used in text generation. Higher values in these metrics indicate greater similarity between the generated and reference texts. These metrics are computed using the NLG-eval package (Sharma et al., 2017). Since the Wikipedia reference article may not always be the "gold answer," we design the *Info Score* to evaluate the overall richness of the generated content. The *Focus Score* examines whether the article remains on topic and maintains a clear focus, while the *Coverage Score* determines if the article provides an in-depth exploration of the topic. We use Prometheus2 to evaluate both *Focus Score* and *Coverage Score* (Shao et al., 2024).

Verifiability Verifiability measures whether the information in a Wikipedia article comes from a reliable source[3]. To assess our model's verifiability, we develop three metrics: Citation Recall, Citation Precision (Liu et al., 2023), and Citation Rate. To implement the measures, we utilize the NLI model TRUE (Honovich et al., 2022). We define $\phi(C_{i,j}, S_i) = 1$ if the citation $C_{i,j}$ entails the sentence S_i. For sentence S_i and its corresponding citations $C_i = \{C_{i,1}, ..., C_{i,O_i}\}$:

$$\text{Citation Recall} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}\left(\max_j \phi(C_{i,j}, S_i) = 1\right) \quad (4)$$

$$\text{Citation Precision} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{O_i}\phi(C_{i,j}, S_i)}{O_i}\right) \quad (5)$$

$$\text{Citation Rate} = \frac{\sum_{i=1}^{N}(\#\text{words}(S_i)\cdot \text{Citation Recall}_i)}{\#\text{words}(\mathcal{W})} \quad (6)$$

As shown in the equations, Citation Recall is the proportion of sentences with at least one valid citation, where $\mathbb{I}$ denotes the indicator function, returning 1 if a condition is true and 0 if false. Citation Precision is the average proportion of valid citations per sentence. To rectify the influence of sentence length, Citation Rate is the weighted average of each sentence's Citation Recall, with the weights being the number of words in the sentences.

[3]https://en.wikipedia.org/wiki/Wikipedia:Verifiability

4 Baseline Methods

According to task description in Section 2.1, we design the following three types of generation frameworks:

RR (Retrieve-then-Read) RR is a naive Retrieval-Augmented Generation (RAG) framework. We follow the "Retrieve-then-Read" method from Ma et al. (2023) and adapt it for Wikipedia article generation. In this framework, a reranker orders reference documents based on their relevance to the event keyword and provides the top L documents to the generator. The generator, a frozen LLM, reads these documents and directly generates the Wikipedia article.

PRR (Plan-Retrieve-Read) PRR is an advanced RAG framework inspired by hierarchical generation techniques in long story and dialogue generation (Fan et al., 2018; Bansal et al., 2022). PRR first uses a frozen LLM to plan the overall structure and generate section headings based on the reference documents. For each section, PRR employs the "Retrieve-then-Read" strategy to rerank related documents according to section headings and event keywords, and then generates the content for each section. We then aggregate the content of each section together to form the final Wikipedia article.

TunedRR The method is inspired by the idea that a small amount of data can teach a model to follow instructions, as shown in LIMA (Zhou et al., 2023). TunedRR aims to finetune the generator based on the RR model. This requires a training dataset of input related documents and corresponding output Wikipedia articles. While the related documents from our evaluation data can be used, the associated Wikipedia articles cannot due to the high failure rate of citation links (Liu et al., 2018). To address this, we leverage the robust performance of GPT4. We feed the related documents into GPT4 and use the same prompt as RR to generate synthesized Wikipedia articles. This process produced 1,011 data samples, which we used to train LLama2 and Vicuna models.

5 Experiments

5.1 Experimental Settings

Our baseline methods include proprietary ChatGPT variants (GPT3.5-turbo and GPT3.5-turbo-16k) and open-source LLMs (Llama2-7b-chat, Llama2-13b-chat (Touvron et al., 2023), Vicuna-7b-v1.5, and Vicuna-13b-v1.5 (Chiang et al.,

2023)). FastChat is employed to enhance inference efficiency in open-source LLMs. All models use the same prompt (details in Appendix C) to ensure evaluation fairness. For rerankers, we use traditional sparse word-based techniques like TF-IDF (Ramos et al., 2003) and BM25 (Robertson et al., 2004), as well as advanced dense vector-based retrievers, including DPR (Karpukhin et al., 2020) and GTR (Ni et al., 2022).

We segment related documents into 256-word chunks (Borgeaud et al., 2022) before feeding them into the reranker module to reduce the computational burden of verifiability evaluation (Gao et al., 2023a). The reranker uses DPR by default and selects the top 5 chunks. For TunedRR, we utilize FastChat to employ full parameter fine-tuning with default hyperparameters. Additionally, we evaluate human-authored Wikipedia articles, as shown in the first row of Table 2. We do not assess the quality of their citations due to the numerous physical documents and links that are difficult to crawl.

5.2 Main Results

In our main experiments, we evaluate three types of frameworks combined with different base LLMs and the default DPR reranker on writing, informativeness, and verifiability. The results are shown in Table 2.

Writing We can see that all methods perform consistently high in terms of fluency. Among the base models we adopt, GPT3.5 stands out as the top performer, achieving impressive *Fluency* and *Organization Scores* that are close to those of human-authored Wikipedia articles. This demonstrates that current LLMs are exceptionally adept at generating organized and readable text, resembling the natural flow and grammatical correctness of human writing. It is also noted that the hierarchical generation methods (PRR) tend to have lower writing performance due to the separate generation of each section, compared to the corresponding RR methods. Benefiting from fine-tuning the output results of GPT4, TunedRR demonstrates improved fluency and coherence. Regarding the outlining ability, we can see that PRR consistently exhibits stable section content planning, regardless of the base model size, compared to RR methods. However, the TunedRR methods perform the worst in outlining, as these models are unable to generate titles in the correct format. Despite this, it does not influence the overall organization score.

Informativeness It is evident that the overall information in reference Wikipedia is very rich, as reflected in the high *Info Score*. Among the three methods, weaker base models benefit from the fine-tuning process of TunedRR and exhibit stable performance across different metrics. The overall informativeness of Wikipedia generated using PRR methods tends to be higher than that of RR methods. It is worth noting that even when PRR produces longer Wikipedia articles than reference Wikipedia, there can still be a significant disparity in the richness of information. This may be caused by the excessive amount of content unrelated to the main topic in each section, as revealed by the relatively low *Focus Score* of PRR. Consequently, this may explain why PRR often scores lower than RR in *Coverage Score*. It is also noted that n-gram based metrics like ROUGE-L and METEOR are heavily influenced by length, as pointed out by Krishna et al. (2021). Thus, using multiple metrics is helpful to analyze the performance of the models.

Verifiability The base generation model plays a critical role in determining citation capability. In the RR and PRR methods, GPT3.5-based methods outperform others significantly. In contrast, open-source models exhibit much lower citation abilities, with Citation Rate not exceeding 20%, aligning with Gao et al. (2023b). The TunedRR methods demonstrate competitive citation capability. Simple fine-tuning can enhance Citation Recall and Citation Precision by nearly 20% compared to RR methods. Nevertheless, the upper limit of this fine-tuning is still suboptimal compared to the capabilities of GPT4. In the future, exploring high-quality data for fine-tuning will be crucial to improving verifiability.

Article Length Reference Wikipedia articles focusing on recent events have around 1,600 words, while RR methods produce shorter articles, typically around 500 words. Hierarchical PRR methods can generate much longer articles, even over 5,000 words. However, the informativeness of generated articles is not necessarily positively correlated with length but depends on the model's capabilities. For example, GPT3.5 achieves a higher *Info Score*, while the 7B weaker models generate excessively long articles with low *Info Score*.

5.3 Analysis of Retrieval Setting

To conduct an in-depth analysis, we explore the retrieval setting, including different reranker tech-

Models	Writing			Informativeness					Verifiability			Length
	Fluency Score	*Org. Score*	*Outline Score*	MET	R-L	*Focus Score*	*Cover. Score*	*Info Score*	Cit. Rate	Cit. Recall	Cit. Prec.	
Reference Wikipedia	4.45	3.61	2.64	-	-	4.02	4.10	4.83	-	-	-	1639.0
RR (Retrieve-then-Read)												
GPT3.5-turbo-0613	**4.31**	4.05	**2.86**	**10.73**	**17.81**	4.26	**3.94**	**3.49**	**42.09**	**38.78**	**36.70**	579.1
GPT3.5-turbo-1106-16k	4.29	4.02	2.84	10.29	17.42	4.22	3.89	3.39	38.38	33.98	32.68	541.3
Llama2-7b	3.87	3.64	1.43	10.21	16.05	3.77	3.27	2.94	10.16	15.85	15.83	625.7
Llama2-13b	3.97	**4.16**	2.39	9.74	15.89	**4.38**	3.91	3.03	7.91	8.91	8.91	552.9
Vicuna-7b	4.06	3.46	1.61	10.18	17.34	3.69	3.39	3.27	6.40	4.41	4.38	535.2
Vicuna-13b	4.18	3.72	2.27	9.80	17.33	3.98	3.63	3.39	16.88	11.03	10.70	491.8
PRR (Plan-Retrieval-Read)												
GPT3.5-turbo-0613	**4.02**	3.36	2.76	22.29	22.26	3.69	**3.51**	**3.78**	**50.96**	53.47	52.43	1991.2
GPT3.5-turbo-1106-16k	**4.02**	3.38	2.76	22.24	22.27	**3.70**	**3.51**	3.76	50.6	**53.57**	**52.53**	1988.9
Llama2-7b	2.83	2.74	2.52	24.14	21.16	2.87	2.69	2.27	13.08	27.02	26.89	4210.4
Llama2-13b	3.70	**3.41**	2.70	24.18	20.94	3.68	3.40	3.29	12.77	14.30	14.30	3789.5
Vicuna-7b	2.89	2.40	2.71	23.43	21.87	2.61	2.68	2.61	14.89	22.40	22.08	5146.4
Vicuna-13b	3.65	3.06	**3.01**	**24.50**	**23.29**	3.20	3.17	3.37	18.96	24.93	23.37	4182.9
TunedRR (Retrieve-then-Read on Fine-tuned Models)												
Llama2-7b-SFT	4.06	3.78	0.47	12.03	17.19	3.91	3.67	3.34	32.29	24.67	21.23	740.9
Llama2-13b-SFT	**4.22**	**3.88**	0.17	11.39	17.19	**4.01**	**3.69**	3.32	**38.08**	27.62	24.85	633.5
Vicuna-7b-SFT	3.93	3.54	**0.66**	**13.45**	17.08	3.57	3.03	3.21	24.74	25.75	22.62	1109.6
Vicuna-13b-SFT	4.07	3.68	0.65	12.80	**17.26**	3.91	3.67	**3.40**	34.68	**29.58**	**26.73**	944.6

Table 2: Wikipedia generation results for different combinations of LLMs and generation methods. Cit. stands for citation, MET for METEOR, R-L for ROUGE-L, Org. for Organization, and Cover. for Coverage. The LLM-based scores are *italicized* and range from 0 to 5, while the other metrics range from 0 to 100. The **best** results for each method are in **bold**, the second best results are underlined.

#Docs	*Fluency Score*	*Org. Score*	R-L	Cit. Recall	Cit. Precision	Length
0	4.62	4.32	16.22	-	-	574.7
5	4.29	4.02	17.42	33.98	32.68	541.3
10	4.30	3.99	17.80	**34.75**	**32.80**	559.9
15	4.29	4.08	18.09	32.41	30.22	583.2
20	4.30	4.10	**18.44**	32.85	30.85	584.9

Table 3: Impact of the number of related documents.

niques, the number of related documents, and citation sources. In these experiments, we use the simple RR method with GPT3.5 as the generator. For the experiment in Table 3, we use GPT3.5-16k to allow more related documents as input, ensuring a sufficiently long context window.

Number of Related Documents We experiment with a number of related documents ranging from 0 to 20, as shown in Table 3. From this table, we see that the length of generated article ranges between 500-600 words and is insensitive to the number of input documents. Without any input, the model can produce a fluent and well-organized article, but none of the generated sentences can be verified.

Overall, the *Fluency* and *Organization Scores* of the model are not significantly affected by the number of related documents. As the number of documents increases, the amount of included information also grows, enhancing the informativeness (ROUGE-L and *Info Score*) of the generated content. At the same time, the verifiability of the model deteriorates with more related documents. The citation quality peaks with around 10 retrieved documents and gradually declines thereafter, indicating that the model may struggle to effectively handle an excessive number of input documents. Therefore, expanding the context window of LLMs may not fully address the challenges of generating full-length Wikipedia articles and could necessitate the integration of more advanced retrieval or reranking methods. The complete experimental results can be found in Table 7.

Sparse vs Dense Reranker To rerank related documents for generation, we compare widely used sparse rankers (TF-IDF, BM25) and dense rankers (DPR, GTR), as shown in Table 4. We used a random selection method as the baseline to set clear benchmarks for the worst possible outcomes. From Table 4, it is evident that articles produced with

Retrieval Method	*Fluency Score*	*Org. Score*	R-L	*Info Score*	Cit. Recall	Cit. Precision
Random	4.17	4.02	16.66	3.23	30.31	26.16
TF-IDF	4.37	4.16	**18.10**	3.59	**49.31**	**48.34**
BM25	**4.39**	4.19	17.38	**3.61**	46.50	44.10
DPR	4.31	**4.31**	17.81	3.49	42.09	36.70
GTR	4.33	4.10	17.89	3.54	44.80	40.21

Table 4: Performance of different rerankers with the top 5 documents are used for generation.

Document Source	*Fluency Score*	*Org. Score*	R-L	*Info Score*	Cit. Recall	Cit. Precision
Search Engine	**4.31**	**4.16**	**4.33**	**3.51**	35.96	34.31
Human Editor	4.31	4.05	4.28	3.43	35.71	33.76
Mixed	4.31	4.05	4.26	3.49	**38.78**	**36.7**

Table 5: Impact of different related document source.

random reranking performed significantly worse across all metrics. Among all reranking techniques, term-matching sparse rerankers (TF-IDF, BM25) outperform dense retrievers. This aligns with Sciavolino et al. (2022), who found that dense retrievers often struggle to identify rare entities not encountered during training, which is a significant issue for Wikipedia. Since sparse rerankers struggle with complex semantic queries and dense rerankers show competitive performance, we choose the commonly adopted DPR method as the default reranker. However, as stated above, LLMs still struggle to effectively utilize all the content within the context length, despite the increasing context length of models. Therefore, reranking techniques are crucial to final performance, and improvements in reranking would benefit the Wikipedia generation task.

Citation Source The related documents come from two sources: search engines and human editors. We analyze how the source of related documents influences Wikipedia article generation. Table 5 presents the generation results using different sources. While it is commonly believed that documents provided by human editors are of higher quality (Liu et al., 2018; Qian et al., 2023), our findings suggest that for new events, search engines can offer more informative and verifiable references. Regarding writing and informativeness, using search engine sources alone performs better than using human editor sources alone or a mix of both. In terms of verifiability, search engine and human editor sources perform similarly, with a mixture performing slightly better than a single source. This indicates that search engines can cover most information provided by human editors. This finding paves the way for the automatic generation of Wikipedia articles for new events, as search engines can provide access to a wide array of up-to-date and extensive news sources, ensuring a breadth and depth of information that rivals or exceeds what human editors can compile.

5.4 Analysis of Supervised Finetuning

This subsection explores how to enhance performance during tuning. We selected one representative metric from each of the three dimensions and plotted their performance trends with training epochs in Figure 2. In the Writing dimension, initial tuning rounds show a decline in fluency, but subsequent epochs reveal improved writing techniques, resulting in progressively higher *Fluency Scores*. After ten epochs, most models exceed their original performance, except for Vicuna-13b, likely due to its initially strong writing abilities. For the Informativeness dimension, a similar trend is observed: information richness initially declines but then recovers and surpasses initial performance levels. Llama2-13b even surpasses GPT3.5 in terms of *Info Score*. However, Vicuna-7b shows good performance by the fifth epoch but overfits after ten epochs, leading to a decline. In the Verifiability dimension, even one training epoch significantly enhances citation abilities. Further training improves citation accuracy, though the improvement rate slows after five epochs. Despite supervised training, open-source models still lag behind proprietary models in performance.

6 Related Work

6.1 Automated Wikipedia Generation

Automatically generating Wikipedia documents has been widely studied for over a decade. Most of the early works primarily focusing on information extraction rather than text generation. Sauper and Barzilay (2009) proposed a structure-aware method to produce Wikipedia documents from relevant articles. WikiWrite (Banerjee and Mitra, 2016) classifies retrieved information by capturing the relationship between referenced and target entities. With the advent of pretrained language models, Liu et al. (2018) addressed Wikipedia generation using a multi-document summarization approach with a decoder-only transformer model. Fan and Gardent (2022) employed BART to generate long-form biographies section by section using supporting doc-

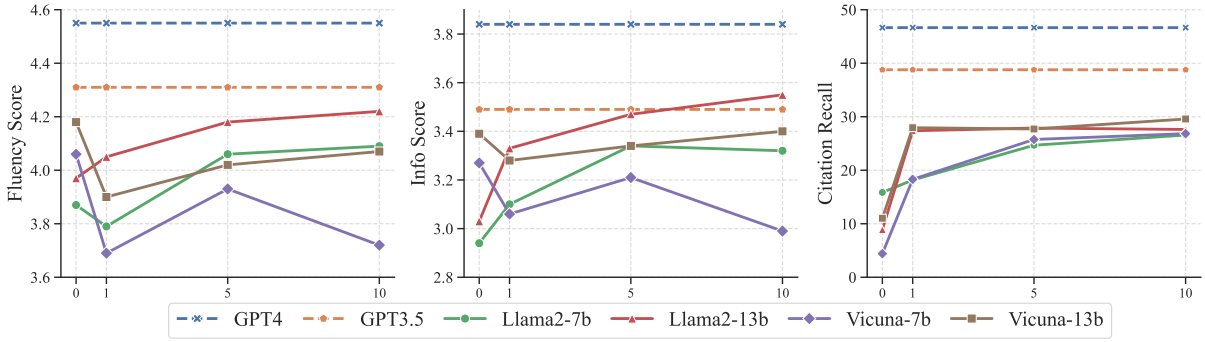

Figure 2: We fine-tuned models of various scales and families, evaluating checkpoints at 1, 5, and 10 epochs. We selected one primary metric from each of the three dimensions and displayed their performance trends with training epochs.

uments. WebBrain (Qian et al., 2023) leverages a complex system to produce short factual articles based on web corpus. The most recent work by Shao et al. (2024) applied GPT-4 for text generation on a small dataset, without exploring open-source models. However, their main focus is on the pre-writing stage and lacks systematic evaluation across different settings of Wikipedia generation.

6.2 Retrieval-Augmented Text Generation

Enhancing LLMs with retrieval during inference has become a common practice for generative tasks. By retrieving relevant document excerpts from external knowledge bases, retrieval-augmented generation models can ground references and generate informative and faithful text (Li et al., 2022; Gao et al., 2023c; Guu et al., 2020). In the era of LLMs, RAG has proven to be an effective and universal paradigm across various NLP tasks (Weston et al., 2018; Jiang et al., 2023). Studies have shown that retrieval can fetch in-context examples (Brown et al., 2020), thereby enhancing certain capabilities of LLMs (Huang et al., 2023; Ram et al., 2023; Chen et al., 2023). Lewis et al. (2020) first introduced RAG models, which combine a pre-trained seq2seq model with a non-parametric dense vector index of Wikipedia, setting new state-of-the-art benchmarks for open domain QA tasks.

To mitigate LLMs' tendency to produce hallucinations, researchers (Nakano et al., 2021) have proposed integrating language models with result pages from search engines to compose the final output. Depending on the complexity of the retrieval strategy, the system may perform multiple retrieval processes during generation and combine them with techniques like Chain of Thought (Trivedi et al., 2022; Feng et al., 2024). Our task emphasizes Wikipedia's verifiability, meaning its content should be determined by previously published information (Liu et al., 2023). Therefore, the RAG approach (Lewis et al., 2020), which involves retrieving relevant document excerpts before inference, is crucial for our task as it grounds citations and facilitates the generation of informative and faithful text (Li et al., 2022; Gao et al., 2023c; Guu et al., 2020).

7 Conclusion

In this paper, we address the challenge of generating full-length Wikipedia articles for newly emerging events. We introduce WIKIGENBENCH, a benchmark for evaluating LLM performance. Our experiments with the RAG framework demonstrate the potential of LLMs to generate coherent and informative Wikipedia articles. We explore various retrieval settings and examine the impact of different citation sources, finding that search engines provide more informative and verifiable references than human editors for new events. We also highlight the importance of high-quality data for fine-tuning to improve article verifiability. Overall, this work compares LLM-based methodologies for full-length Wikipedia generation, providing insights and guiding future research. The evaluation metrics we develop assist in assessing generated Wikipedia articles, leading to more reliable and informative automatic content generation.

Limitations

Our research encounters limitations, notably in our section-by-section generation approach, which may lead to redundancy and necessitate a rewriting strategy to ensure article cohesion. Further limitations

include the challenge of direct citation by LLMs, a bottleneck that might necessitate the exploration of post-citation methods such as employing NLI for improvement. Additionally, the information from related documents does not fully cover the content of original Wikipedia articles, making the n-gram metric comparison between the generated text and the original articles a weak reference rather than a definitive standard.

Ethics Statement

This work adheres to the ACL Ethics Policy. We assert that, to the best of our knowledge, our work does not present any ethical issues. We have conducted a thorough review of potential ethical implications in our research and found none.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Siddhartha Banerjee and Prasenjit Mitra. 2015. Wikikreator: Improving wikipedia stubs automatically. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 867–877.

Siddhartha Banerjee and Prasenjit Mitra. 2016. Wikiwrite: Generating wikipedia articles automatically. In *IJCAI*, pages 2740–2746.

Srijan Bansal, Suraj Tripathi, Sumit Agarwal, Sireesh Gururaja, Aditya Srikanth Veerubhotla, Ritam Dutt, Teruko Mitamura, and Eric Nyberg. 2022. R3 : Refined retriever-reader pipeline for multidoc2dial. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 148–154, Dublin, Ireland. Association for Computational Linguistics.

Stella Biderman, Kieran Bicheno, and Leo Gao. 2022. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Qinyu Chen, Wenhao Wu, and Sujian Li. 2023. Exploring in-context learning for knowledge grounded dialog generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10071–10081.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Angela Fan and Claire Gardent. 2022. Generating full length wikipedia biographies: The impact of gender bias on the retrieval-based generation of women biographies. *arXiv preprint arXiv:2204.05879*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Zhangyin Feng, Xiaocheng Feng, Dezhi Zhao, Maojin Yang, and Bing Qin. 2024. Retrieval-generation synergy augmented large language models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11661–11665.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023c. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, and Juanzi Li. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

Jie Huang, Wei Ping, Peng Xu, Mohammad Shoeybi, Kevin Chen-Chuan Chang, and Bryan Catanzaro. 2023. Raven: In-context learning with retrieval augmented encoder-decoder language models. *arXiv preprint arXiv:2308.07922*.

Martin Huschens, Martin Briesch, Dominik Sobania, and Franz Rothlauf. 2023. Do you trust chatgpt?–perceived credibility of human and ai-generated content. *arXiv preprint arXiv:2309.02524*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957, Online. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Florian Lemmerich, Diego Sáez-Trumper, Robert West, and Leila Zia. 2019. Why the world reads wikipedia: Beyond english speakers. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 618–626.

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

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

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

Yen-Ting Lin and Yun-Nung Chen. 2023a. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models.

Yen-Ting Lin and Yun-Nung Chen. 2023b. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 47–58, Toronto, Canada. Association for Computational Linguistics.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*.

Robert L Logan IV, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2021. Fruit: Faithfully reflecting updated information in text. *arXiv preprint arXiv:2112.08634*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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

Laura Perez-Beltrachini, Yang Liu, and Mirella Lapata. 2019. Generating summaries with topic templates and structured convolutional decoders. *arXiv preprint arXiv:1906.04687*.

Hongjing Qian, Yutao Zhu, Zhicheng Dou, Haoqi Gu, Xinyu Zhang, Zheng Liu, Ruofei Lai, Zhao Cao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Webbrain: Learning to generate factually correct articles for queries by grounding on large web corpus. *arXiv preprint arXiv:2304.04358*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2022. Simple entity-centric questions challenge dense retrievers.

Yijia Shao, Yucheng Jiang, Theodore A Kanell, Peter Xu, Omar Khattab, and Monica S Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *arXiv preprint arXiv:2402.14207*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation.

Vijay Sharma, Namita Mittal, Ankit Vidyarthi, and Deepak Gupta. 2024. Exploring web-based translation resources applied to hindi-english cross-lingual information retrieval. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(1):1–19.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of gpt-4: reliably evaluating large language models on sequence to sequence tasks. *arXiv preprint arXiv:2310.13800*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Evaluation Metrics Comparison

Relevant Work	Writing	Informativeness	Faithfulness/Verifiability	Full Length
(Banerjee and Mitra, 2016)	✗	ROUGE	✗	✔
(Liu et al., 2018)	✍	ROUGE log-perplexity	✗	✔
(Perez-Beltrachini et al., 2019)	✍	ROUGE Unigram F-measure	✗	✗
(Liu and Lapata, 2019)	✍	ROUGE	✗	✗
(Logan IV et al., 2021)	✗	UpdateROUGE	Entity Prec. & Recall	✗
(Fan and Gardent, 2022)	✗	ROUGE NER Coverage	NLI	✔
(Qian et al., 2023)	✍	BLEU METEOR ROUGE CIDEr	QAGS TripleScore	✗
(Shao et al., 2024)	Organization Score by Prometheus Focus Score by Prometheus Outline Recall by Prometheus	ROUGE Entity Recall Coverage by Prometheus	NLI(Citation quality)	✔
Ours	Fluency Score by GPT-4 Outline Score by GPT-4 Organization Score by Prometheus Relevance Score by Prometheus	BLEU METEOR ROUGE Info Score by GPT-4 Coverage Score by Prometheus Focus Score by Prometheus	NLI(Citation quality)	✔

Table 6: Comparison of Evaluation Metrics Across Different Works. ✍ indicates human evaluation, ✗ indicates no evaluation or the work is about short Wikipedia snippet, and ✔ signifies the work is about full-length Wikipedia generation. Metrics include writing quality, informativeness, and verifiability.

B WikiGenBen Datasest

B.1 Wikipedia Reference

This section provides a detailed example of a data entry for the "2023 USFL season," illustrating the dataset's structure and information richness. Our dataset includes detailed information of Wikipedia reference articles, acquired using the Python library MediaWiki.

- **ID:** 71284256
- **Keyword:** 2023 USFL season
- **URL:** https://en.wikipedia.org/wiki/2023%20USFL%20season
- **Summary:** The 2023 USFL season was the second season of the United States Football League. The regular season started on April 15 and ended on June 18. The postseason began on June 24 and ended with the 2023 USFL Championship Game on July 1. The league expanded the locations their teams play to four total stadiums, adding Ford Field in Detroit, Michigan, and Simmons Bank Liberty Stadium in Memphis, Tennessee.
- **Sections:** Offseason, Locations, Teams, Players, ...
- **Content for each section:**
  1. During the 2022 season, . . .
  2. The league stated its . . .
  3. On November 15, 2022, in conjunction. . .
  4. For the 2023 season, each USFL team. . .
  5. ...
- **Infobox:**

Key	Value
League	United States Football League
Sport	American football
Duration	Regular season: April 15 – June 18 Playoffs: June 24 – July 1
...	...

B.2 Related Documents

Beyond the core Wikipedia entries, our dataset includes related documents categorized as 'Human' and 'Search'. 'Human' documents come from the Wikipedia External Links Section, offering human-curated, credible information. 'Search' documents are obtained through Google searches, providing diverse perspectives and additional context.

- **Doc ID:** 1
- **Title:** Johnson, Roy S. (2022-11-14). ÜSFL reveals season 2 details for Birmingham, al. Retrieved 2022-12-13.
- **URL:** https://www.al.com/news/2022/11/usfl-reveals-season-2-details-for-birmingham.html
- **Content:** usfl reveals season 2 details for...
- **Source:** Human / Search

B.3 Training Data

Wikipedia text is typically used directly for training. However, due to the high failure rate of Wikipedia's related document links, we use an alternative method. We employ DPR reranking to fetch the top-5 documents for each entry and then use the RR prompt to generate text with GPT-4. This approach aims to teach the target model GPT-4's citation generation capabilities.

C Prompt for Generating

Our method involves creating a base template for the prompt, which is then supplemented with relevant documents until reaching a maximum input length. Specifically, for 4k models, the maximum input length is strictly capped at 2048 tokens.

C.1 Retrieve-then-Read

The **RR** approach involves a straightforward, one-stage process for directly generating an article.

Article Generation Prompt

Input:
I have a topic "{**keyword**}" that contains the following documents:
Document 1: {**doc1**}
Document 2: {**doc2**}
...
Based on the above information, you are assigned to write a Wikipedia article on the topic.
Organize the content of your article by sections. Before writing each section, always starts with "==SECTION NAME==".
You must cite the most relevant document for every sentence you write, in the format of "This is an example sentence.[k]", where k denotes Document k.

C.2 Plan-Retrieve-Read

In contrast, the **PRR** method necessitates a more structured approach to article generation. Initially, it requires the planning of an article outline. Once the outline is established, each section name generated during the planning phase serves as a guide for the subsequent retrieval and writing phases.

Outline Generation Prompt

Input:
I have a topic "{**keyword**}" that contains the following documents:
Document 1: {**doc1**}
Document 2: {**doc2**}
...
Based on the above information, you are assigned to write an outline for a Wikipedia article about this topic.
Your outline should only include the names of the sections, without any further details.
Do not use document name as your outline.
The format of your outline should be as follows:
1. Introduction
2. <Section Name 1>
...
n. <Section Name n>

Section Generation Prompt

Input:
I have a topic "{**keyword**}" and a section "{**section**}" that contains the following documents:
Document 1: {**doc1**}
Document 2: {**doc2**}
...
Based on the above information, you are assigned to write a Wikipedia article on the topic.
You must cite the most relevant document for every sentence you write, in the format of "This is an example sentence.[k]", where k denotes Document k.

D Prompt for Evaluating

We employed the GPT-4-1106-preview model by OpenAI for scoring, setting the temperature to 0 and keeping other parameters at their defaults. Regular expressions were used to match the corresponding scores. As LLM-EVAL(Lin and Chen, 2023b) shows that a single prompt can obtain multi-dimensional scores correlating well with human preferences, we called the GPT-4 API only once to get the Fluency and Informativeness Scores. This approach significantly reduces costs by eliminating the need for multiple prompts.

Evaluation Prompt for Fluent Score and Informativeness Score

Input:
Evaluate an encyclopedia text of a keyword on three metrics: fluency, informativeness, and faithfulness.
Give a score from 0-5 for each metric.
- Fluency: Assess the text for grammatical correctness, coherence of ideas, and overall readability. Look for smooth transitions between sentences and paragraphs, as well as clear organization of information.
- Informativeness: Evaluate the depth and breadth of information provided about the keyword. Check if the text covers various aspects of the topic, including its definition, background, significance, related concepts, and any relevant examples or applications.
- Faithfulness: Verify the accuracy of the information presented in the text by cross-referencing with credible sources or established knowledge. Assess whether the information aligns with accepted facts and evidence.

Only give three scores in the form of: Fluency: Score 1, Informativeness: Score 2, Faithfulness: Score 3. No need for explanation.

The GPT-4-1106-preview model is trained on data up to April 2023. Since nearly one-third of events in our benchmark occurred after April 2023, GPT-4-1106-preview cannot evaluate faithfulness accurately. Therefore, we do not report the faithfulness score.

Evaluation Prompt for Outline Score

Input:
Given a keyword and an outline about the Wikipedia of the keyword, assign a score ranging from 0 to 5 to evaluate the quality of the outline. Only give the score without explanation.

We adopt the same scoring rubrics as previous studies (Shao et al., 2024) to assess Organization, Focus and Coverage, with the exception of replacing Prometheus with Prometheus-2 for an extended context

length of 32k.

Criteria Description for Organization Score, Focus Score and Coverage Score

Coherence and Organization:
Is the article well-organized and logically structured?
- Score 1: Disorganized; lacks logical structure and coherence.
- Score 2: Fairly organized; a basic structure is present but not consistently followed.
- Score 3: Organized; a clear structure is mostly followed with some lapses in coherence.
- Score 4: Good organization; a clear structure with minor lapses in coherence.
- Score 5: Excellently organized; the article is logically structured with seamless transitions and a clear argument.

Relevance and Focus:
Does the article stay on topic and maintain a clear focus?
- Score 1: Off-topic; the content does not align with the headline or core subject.
- Score 2: Somewhat on topic but with several digressions; the core subject is evident but not consistently adhered to.
- Score 3: Generally on topic, despite a few unrelated details.
- Score 4: Mostly on topic and focused; the narrative has a consistent relevance to the core subject with infrequent digressions.
- Score 5: Exceptionally focused and entirely on topic; the article is tightly centered on the subject, with every piece of information contributing to a comprehensive understanding of the topic.

Broad Coverage:
Does the article provide an in-depth exploration of the topic and have good coverage?
- Score 1: Severely lacking; offers little to no coverage of the topic's primary aspects, resulting in a very narrow perspective.
- Score 2: Partial coverage; includes some of the topic's main aspects but misses others, resulting in an incomplete portrayal.
- Score 3: Acceptable breadth; covers most main aspects, though it may stray into minor unnecessary details or overlook some relevant points.
- Score 4: Good coverage; achieves broad coverage of the topic, hitting on all major points with minimal extraneous information.
- Score 5: Exemplary in breadth; delivers outstanding coverage, thoroughly detailing all crucial aspects of the topic without including irrelevant information.

E Generation Format

For generation, we require the model to append references in the format [1][2] after each claim, indicating support by chunk 1 and chunk 2. Additionally, we generate section titles using '==' as delimiters. We decompose the article accordingly and evaluate the quality of citations and section titles.

An example of model output

2022 EFL League One play-off final:

==Introduction==

The 2022 EFL League One play-off final is a highly anticipated football match that will determine the final promotion spot to the EFL Championship for the 2022-2023 season. The play-off final is the culmination of the EFL League One play-offs, which involve the teams that finish in the third to sixth positions in the league table. The winner of the final will join the top two teams in the league in gaining promotion to the higher division. This article will provide an overview of the 2022 EFL League One play-off final, including the date, venue, and teams involved.

==Date and Venue==

The 2022 EFL League One play-off final is scheduled to take place on Saturday, May 21, 2022.[2] The final will be held at Wembley Stadium in London, England, which has been the traditional venue for the EFL play-off finals.[2] Wembley Stadium is known for its iconic status in English football and has a seating capacity of over 90,000 spectators.[2]

==Teams==

The teams that will compete in the 2022 EFL League One play-off final will be determined through the League One play-off semi-finals. The semi-finals will involve the teams that finish in the third to sixth positions in the League One table.[2] The semi-final matches are scheduled to take place on Thursday, May 5, 2022, and Friday, May 6, 2022.[2] The winners of the semi-finals will advance to the final and compete for the promotion spot to the EFL Championship.

==Ticket Information==

Tickets for the 2022 EFL League One play-off final are expected to go on sale in May.[3] The English Football League (EFL) will provide more information on ticket sales closer to the date of the final.[3] In previous years, the play-off finals have attracted a significant number of spectators, with Wembley Stadium hosting capacity-limited crowds of just under 12,000.[3] However, this year, with the return to full capacity crowds, the atmosphere is expected to be even more electric.

==Broadcasting==

The 2022 EFL League One play-off final, along with the semi-finals, will be televised by Sky Sports, allowing fans to watch the matches from the comfort of their homes.[3] Additionally, highlights of the finals will be available on Quest, providing fans with a recap of the key moments from the matches.[3]

==Conclusion==

The 2022 EFL League One play-off final is set to be an exciting and highly anticipated event in English football. Taking place at Wembley Stadium, the final will determine the final promotion spot to the EFL Championship for the 2022-2023 season. With tickets expected to go on sale in May and the matches being broadcasted on Sky Sports, fans will have the opportunity to witness the drama and excitement of the play-off final.

F Impact of Retrieval Settings

Doc #	Writing			Informativeness					Verifiability			Length
	Fluency Score	Org. Score	Outline Score	MET	R-L	*Focus Score*	*Cover. Score*	*Info Score*	*Cit. Rate*	Cit. Recall	Cit. Prec.	
0	**4.62**	**4.32**	**2.91**	10.51	16.22	**4.61**	**4.29**	**3.70**	-	-	-	574.7
5	4.29	4.02	2.84	10.29	17.42	4.22	3.89	3.39	**38.38**	33.98	32.68	541.3
10	4.30	3.99	2.80	10.54	17.80	4.18	3.83	3.44	37.70	**34.75**	**32.80**	559.9
15	4.29	4.08	2.83	10.84	18.09	4.28	3.94	3.52	35.47	32.41	30.22	583.2
20	4.30	4.10	2.82	**10.99**	**18.44**	4.33	3.83	3.55	34.80	32.85	30.85	584.9

Table 7: Impact of the Number of Retrieved Documents

Rerank Method	Writing			Informativeness					Verifiability			Length
	Fluency Score	Org. Score	Outline Score	MET	R-L	*Focus Score*	*Cover. Score*	*Info Score*	*Cit. Rate*	Cit. Recall	Cit. Prec.	
Random	4.17	4.02	2.80	9.99	16.66	4.28	3.71	3.23	30.31	27.26	26.16	534.0
TF-IDF	4.37	4.16	**2.89**	10.67	**18.10**	**4.36**	**4.04**	3.59	**49.31**	**50.32**	**48.34**	576.9
BM25	**4.39**	4.19	2.83	10.17	17.38	**4.36**	3.95	**3.61**	46.50	46.74	44.10	546.8
DPR	4.31	**4.31**	2.86	**10.73**	17.81	3.49	3.57	3.49	42.09	38.78	36.70	577.2
GTR	4.33	4.10	2.86	10.46	17.89	4.31	3.97	3.54	44.80	41.97	40.21	560.0

Table 8: Performance of different rerankers.

Information Source	Writing			Informativeness					Verifiability			Length
	Fluency Score	Org. Score	Outline Score	MET	R-L	*Focus Score*	*Cover. Score*	*Info Score*	*Cit. Rate*	Cit. Recall	Cit. Prec.	
Search Engine	**4.31**	**4.16**	2.84	10.23	17.54	**4.33**	3.86	**3.51**	40.37	35.96	34.31	542.8
Human Editor	**4.31**	4.05	2.77	10.36	17.70	4.28	3.92	3.43	38.70	35.71	33.76	540.8
Mixed	**4.31**	4.05	**2.86**	**10.79**	**17.89**	4.26	**3.94**	3.49	**42.09**	**38.78**	**36.70**	579.1

Table 9: Influence of different related document source.

G Full Evaluation Results on Finetuned Models

Rerank Methods	Training Epochs	Writing			Informativeness					Verifiability			Length
		Fluency Score	*Org. Score*	*Outline Score*	*MET*	*R-L*	*Focus Score*	*Cover. Score*	*Info Score*	*Cit. Rate*	*Cit. Recall*	***Cit. Prec.***	
GPT4	**-**	4.55	4.40	2.88	10.74	18.08	4.57	4.29	3.84	54.27	46.65	40.84	569.0
Llama2-7B	**0**	3.87	3.64	1.43	10.21	16.05	3.77	3.27	2.94	10.16	15.85	15.83	625.7
	1	3.79	3.27	0.48	12.58	16.08	3.78	3.41	3.10	30.04	18.17	15.01	1113.1
	5	4.06	3.78	0.47	12.03	17.19	3.91	3.67	3.34	32.29	24.67	21.23	740.9
	10	4.09	3.76	0.39	11.73	17.23	3.93	3.74	3.32	33.38	26.58	23.20	699.4
Llama2-13B	0	3.97	4.16	2.39	9.74	15.89	4.38	3.91	3.03	7.91	8.91	8.91	552.9
	1	4.05	3.52	0.09	10.58	16.70	3.66	3.39	3.33	37.07	27.38	25.73	672.3
	5	4.18	3.80	0.17	11.30	17.26	3.98	3.67	3.47	38.68	27.88	24.29	643.3
	10	4.22	3.88	0.17	11.39	17.19	4.01	3.69	3.55	38.08	27.62	24.85	633.5
Vicuna-7B	0	4.06	3.46	1.61	10.18	17.34	3.69	3.39	3.27	6.40	4.41	4.38	535.2
	1	3.69	3.27	0.26	12.11	16.26	3.57	3.03	3.06	25.89	18.30	17.11	1006.0
	5	3.93	3.54	0.66	13.45	17.08	3.78	3.44	3.21	24.74	25.75	22.62	1109.6
	10	3.72	3.19	0.39	15.27	16.78	3.49	3.25	2.99	18.03	26.84	23.85	1444.9
Vicuna-13B	0	4.18	3.72	2.27	9.80	17.33	3.98	3.63	3.39	16.88	11.03	10.70	491.8
	1	3.90	3.49	0.16	11.33	16.51	3.74	3.46	3.28	36.96	27.95	26.29	814.0
	5	4.02	3.68	0.77	12.92	17.36	4.01	3.60	3.34	30.68	27.71	25.25	984.3
	10	4.07	3.68	0.65	12.80	17.26	3.91	3.67	3.40	34.68	29.58	26.73	944.6

Table 10: We selected different model checkpoints during training and evaluated their performance on testset.