ConTReGen: Context-driven Tree-structured Retrieval for Open-domain Long-form Text Generation

Kashob Kumar Roy[1] Pritom Saha Akash[1] Kevin Chen-Chuan Chang[1] Lucian Popa[2]

[1]University of Illinois at Urbana-Champaign, USA   [2]IBM Research, USA

{kkroy2, pakash2, kcchang}@illinois.edu,   lpopa@us.ibm.com

Abstract

Open-domain long-form text generation requires generating coherent, comprehensive responses that address complex queries with both breadth and depth. This task is challenging due to the need to accurately capture diverse facets of input queries. Existing iterative retrieval-augmented generation (RAG) approaches often struggle to delve deeply into each facet of complex queries and integrate knowledge from various sources effectively. This paper introduces ConTReGen, a novel framework that employs a context-driven, tree-structured retrieval approach to enhance the depth and relevance of retrieved content. ConTReGen integrates a hierarchical, top-down in-depth exploration of query facets with a systematic bottom-up synthesis, ensuring comprehensive coverage and coherent integration of multifaceted information. Extensive experiments on multiple datasets, including LFQA and ODSUM, alongside a newly introduced dataset, ODSUM-WikiHow, demonstrate that ConTReGen outperforms existing state-of-the-art RAG models. [1]

1 Introduction

Large Language Models (LLMs) have transformed various domains through their remarkable performance across a spectrum of tasks. However, LLMs often struggle with generating hallucinated or factually incorrect content, particularly when addressing knowledge-intensive tasks in open-domain settings (Asai et al., 2023a; Gao et al., 2023). These limitations typically arise from either the lack of long-tail relevant knowledge or reliance on outdated information embedded within their parameters. To address these challenges, Retrieval-augmented Generation (RAG), therefore, has emerged as a promising solution (Lewis et al., 2020; Petroni et al., 2020; Izacard et al., 2023).

RAG enhances LLMs by incorporating external knowledge from a corpus, effectively reducing hallucinations and factual errors in knowledge-driven tasks such as question answering.
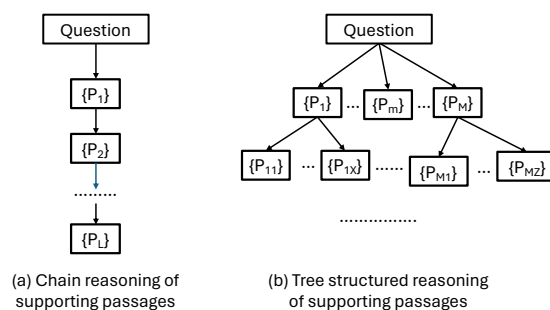

Figure 1: Schematic illustration of retrieval reasoning

Recently, significant progress has been made in open-domain question answering, particularly in the realm of short-form answers (Shao et al., 2023; Press et al., 2023; Trivedi et al., 2022; Xu et al., 2024). Rather than adhering to the traditional one-step retrieve-then-generation approach, recent works increasingly adopt iterative retrieval processes (Gao et al., 2023). These methods utilize either previously generated responses (Shao et al., 2023; Trivedi et al., 2022; Jiang et al., 2023; Asai et al., 2023b), or employ a sequence of follow-up questions or interlinked queries (Press et al., 2023; Xu et al., 2024; Qi et al., 2019). Such iterative, chain-like strategies have proven particularly effective in short-form factoid question-answering tasks where questions often demand direct, specific pieces of information (Qi et al., 2019; Press et al., 2023) and their supporting passages commonly form chain-reasoning of facts (Xiong et al., 2020; Trivedi et al., 2022; Xu et al., 2021) as in Figure 1(a).

Moving beyond short-form question answering, real-world queries often entail greater complexity and necessitate more comprehensive, detailed responses that encompass multiple facets

---

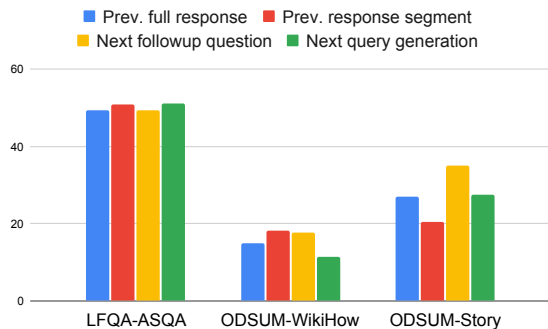[1]Code and data are available at https://github.com/kkroy2/ConTReGen

Figure 2: Retrieval Recall Performance. Prev. full response (Shao et al., 2023), Prev. response segment (Asai et al., 2023b), Next followup question (Press et al., 2023; Xu et al., 2024), Next query generation (Khattab et al., 2022).
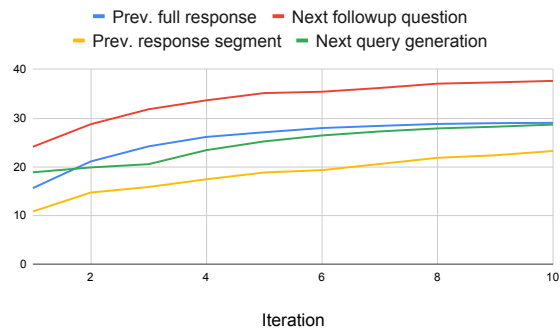

Figure 3: Retrieval Recall Performance per iteration on ODSUM-Story. Prev. full response (Shao et al., 2023), Prev. response segment (Asai et al., 2023b), Next followup question (Press et al., 2023; Xu et al., 2024), Next query generation (Khattab et al., 2022)

of the queries. For these multifaceted queries, it is essential to retrieve supporting knowledge from diverse sources, integrating this information into coherent and comprehensive long-form responses. We refer to this scenario as *Open-domain Long-form Text Generation*. This perspective unifies the commonalities between existing long-form question-answering (LFQA) and open-domain multi-document summarization (ODSUM) tasks, addressing both under a single framework. LFQA (Krishna et al., 2021; Fan et al., 2019) requires not only retrieval of relevant facts from diverse knowledge sources but also integrating them into a coherent paragraph-length answer. Similarly, ODSUM (Giorgi et al., 2023) involves the aggregation of information from diverse sources into a unified, coherent summary.

In our experiments with chain-like iterative approaches on LFQA and ODSUM tasks, we identified several limitations. One notable issue is the low retrieval recall of performance across the datasets in Figure 2, especially evident in ODSUM datasets where the need for multi-faceted information is more implicit. Secondly, we observed that these iterative approaches quickly reach a plateau in terms of retrieval recall as demonstrated in Figure 3. In other words, they fail to retrieve new relevant passages during subsequent iterations.

One of the key reasons behind these limitations is the simplistic modeling of the reasoning of retrieving passages as a chain of semantic aspects as shown in Figure 1(a). However, when the retrieval of information spans multiple facets and originates from diverse sources, it naturally forms a complex retrieval reasoning structure as illustrated in Fig-

ure 1(b). For example, a query like '*How to Detect Hidden Cameras and Microphones*' may require information from multiple facets such as '*Conducting a Physical Search*', '*Searching for Electrical Signals*', and so on, where each facet may need information from its multiple subfacets. For instance, this '*Conducting a Physical Search*' facet needs information about its subfacets such as '*Investigating your smoke detectors and other electronics.*', '*Using to check for two-way mirrors*', etc. Consequently, this yields a tree structure of facets that organizes the need for information from the broadest to the most specific ones.

Furthermore, the method of formulating subsequent retrieval queries plays a crucial role in thoroughly exploring all facets of an input query. Current iterative approaches often generate the next query based on segments of the previous response or all previously retrieved information. We argue that this sequential approach to query formulation inherently restricts the retrieval search scope. This is because it tends to allow a few aspects of the information to dominate the retrieval process, potentially leading to a biased exploration where certain facets are disproportionately emphasized, while others may remain underexplored or unexplored.

To accommodate these findings, we introduce a novel tree-structured retrieval augmented generation framework (ConTReGen) that conceptualizes the retrieval process as a hierarchical exploration of an input question or query's various facets, organized in a tree structure. Each branch of this tree represents a specific facet, enabling a systematic and comprehensive exploration of the query. This method enables an in-depth exploration of

(a) Top-down Planning and Retrieval
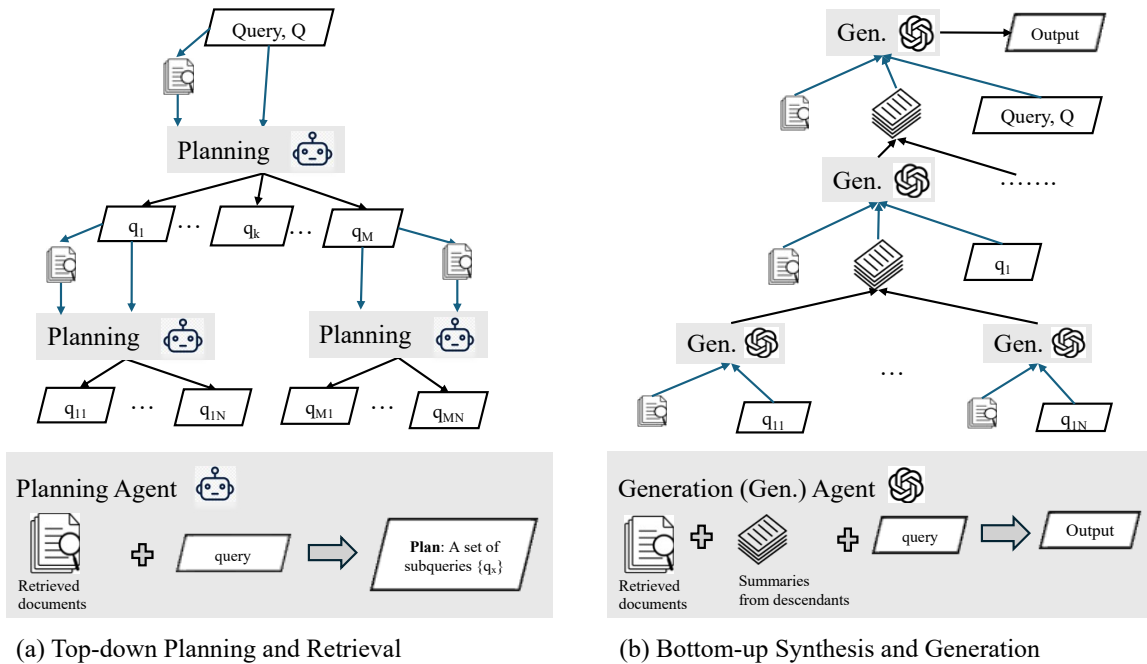
(b) Bottom-up Synthesis and Generation

Figure 4: ConTReGen Framework.

each query facet, from the broadest facet down to the finer specific through a top-down strategy. Additionally, to facilitate a more thorough aggregation and synthesis of information, we leverage a bottom-up generation technique that synthesizes information from the leaf nodes upwards, ensuring that all retrieved data contributes cohesively to the final output. This integrated approach significantly enhances the depth and relevance of the retrieved content, contributing to more coherent and contextually rich text generation.

To evaluate the efficacy of our framework on open-domain long-form text generation, we conducted experiments using both the LFQA and ODSUM datasets, including ASQA (Stelmakh et al., 2022) and ODSUM-Story (Zhou et al., 2023). Additionally, we introduced a new, large-scale open-domain summarization dataset named ODSUM-WikiHow. The experimental results demonstrate that ConTReGen significantly outperforms state-of-the-art RAG baselines across all three datasets.

2 Methodology

Our framework addresses the task of context-driven Long-Form Text Generation which aims to generate a comprehensive long text, $\mathbf{Y}$ for an input query or question, denoted as, $\mathbf{Q}$ by leveraging relevant passages from a given corpus $\mathbf{C}$.

Our proposed framework is structured into a two-

stage process designed to enhance the retrieval and generation of long-form text responses: 1) during the top-down planning and retrieval stage as shown in Figure 4(a), the process begins by retrieving passages directly related to the input query or question. Utilizing the insights gained from these initial passages, the planning agent generates a series of subquestions or subqueries, which serve as a plan for deeper exploration. Each of these subquestions is recursively used to generate successive plans to continually expand the search scope of relevant passages. This recursive planning and retrieval results in a comprehensive tree structure, where each node represents a query and its associated retrieved passages; 2) in Figure 4(b), the bottom-up synthesis and generation stage of the framework begins at the leaf nodes, summarizes the relevant information from retrieved documents. The model then works its way upwards, synthesizing the retrieved documents and the summarized information from descendant nodes to generate text responses for each intermediate node. This ascending integration continues until it reaches the root of the tree, producing a final response that comprehensively addresses the original input question or queries. This methodical approach ensures that every facet of the query is thoroughly explored, allowing for a detailed and well-structured response generation.

2.1 Top-down Planning and Retrieval

The concept of top-down planning and retrieval stems from observing how humans approach complex information-seeking tasks (Kuhlthau, 1991; Marchionini, 1995). When faced with a multi-faceted query, individuals naturally decompose the complex query into smaller, specific facet-focused manageable sub-queries (Wei et al., 2013). Our framework aims to mimic this human-like approach to information retrieval by structuring the process as a hierarchical exploration of various facets of a complex query or question. This top-down planning and retrieval stage is designed to systematically break down the query into subquestions/subqueries from multiple facets that guide the retrieval process towards a more thorough and nuanced aggregation of relevant information.

Planning. The process begins with analyzing the input query or question and its initial retrieved passages. We employ LLM as the planning agent in Figure 4(a) to generate a series of sub-questions as a plan, ensuring a broad and diverse exploration of the query or question. Each sub-question targets a specific aspect/facet of the query.
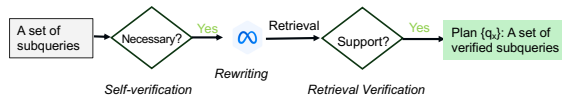

Figure 5: Two-step Verification.

Verification. Ideally, each sub-question should be essential for addressing the original question or query and should be contextually well-informed to ensure it can retrieve passages relevant to the original query. We apply a two-step verification process as shown in Figure 5: i) we leverage LLM to predict whether a sub-question is required to address the main query/question. If the sub-question is identified as necessary, the LLM is used to rewrite it as a self-sufficient and contextually-rich search query. This step ensures that each sub-question is both relevant and comprehensive, making it capable of independently guiding the retrieval process; ii) Each sub-question is then used individually as a query to retrieve passages. We evaluate whether these retrieved passages are relevant to the original query. For this relevancy assessment, we again use LLMs to ensure that the retrieved passages effectively contribute to addressing the original query. This verification process ensures that the decom-

Dataset	ODSUM-Story	ODSUM-WikiHow
Corpus Size	1138	506295
Number of Queries	635	25896
Avg # docs per Query	8.78	15.26
Avg Doc Length	621.93	81.29
Avg Query Length	10.18	6.83
Avg Summary Length	274.65	124.24

Table 1: Statistics of ODSUM-Story and ODSUM-WikiHow: Avg length in terms of words.

position of the original query into sub-questions is not only comprehensive and contextually accurate but also effective in retrieving relevant information.

Recursive Exploration. Once obtained verified set of subquestions for the original input, we recursively repeat the same planning-then-verification strategy to generate successive plans for each subquestion, continuously expanding the retrieval scope. This recursive process ensures an in-depth exploration of each query dimension, from the broadest facets to the finer ones. It continues until no further plans are needed or the predefined maximum depth is reached. This stage results in a detailed and well-structured tree where each node represents a specific sub-question and its corresponding relevant passages and the root node corresponds to the original query.

2.2 Bottom-up Synthesis and Generation

This stage, as shown in Figure 4(b), focuses on synthesizing the retrieved information to produce a coherent and comprehensive long-form response that addresses the original query. Key advantages of this bottom-up strategy include: i) utilizing a large volume of retrieved passages by summarizing information at each descendant level without exceeding the input token length limit, and ii) effectively filtering out irrelevant or less important details, which enhances the quality of the final response. The process begins at the leaf nodes of the tree, which contain the most specific subquestions and their retrieved passages. Each leaf subquestion represents a fine-grained aspect of the original query. We employ LLM to summarize the key information from the retrieved passages that are relevant to the sub-question. answer to complex, open-domain questions.

From the leaf nodes, the synthesis and generation process works its way upwards through the tree. Similar to the leaf node, each intermediate node contains a subquestion and its retrieved passages. Each intermediate node integrates the summarized

information from its child nodes with its retrieved passages. This involves merging the key insights and details from the descendant queries to form a coherent and contextually rich response for each sub-question represented by the node. This upward process continues through each level of the tree. At each step, the synthesized response from a lower level is combined with the retrieved passages at the current level. The process stops at the root node, which represents the original query or question. At this point, the final comprehensive response is generated to address the original query, effectively utilizing the information gathered throughout the process.

3 WikiHow: A new ODSUM dataset

Open-domain summarization (ODSUM) is to generate a comprehensive summary for a given query by gathering information from a large index of documents. One of the key challenges in open-domain summarization (ODSUM) is the scarcity of datasets that encompass real-world queries alongside a well-curated document corpus (Giorgi et al., 2023). While a recent work (Zhou et al., 2023) constructed a dataset ODSUM-Story from SQuAL-ITY (Wang et al., 2022), question-based abstraction summarization for short stories. However, the corpus size and the diversity of queries (only from the story domain) are limited as shown in Table 1. To address this gap, we introduce a large-scale open-domain summarization derived from WikiHow[2] articles. Existing datasets sourced from WikiHow, such as (Cohen et al., 2021; Koupaee and Wang, 2018; Boni et al., 2021), are constructed for traditional text summarization tasks. Moreover, A recent WikiHowQA dataset (Bolotova-Baranova et al., 2023) focuses on multi-document long-form question answering with supporting documents sourced from article reference links. Unfortunately, its corpus of documents is not publicly available, which limits its utility in LFQA. In addition, it utilizes only 11,746 queries from a potential 25,896. On the other hand, we have developed the ODSUM-WikiHow dataset, which includes 25,896 queries, each paired with human-written, coherent summaries and supported by a substantial corpus of 506,295 documents.

Dataset Construction   We process the article titles as input queries and use author-provided summaries as the ground-truth summaries. WikiHow

articles are categorized into two types: one type describes tasks using a single method detailed in successive steps, while the other type outlines multiple methods, with each method comprising several steps. To prepare the corpus, we exploit this structured format where each step is discussed in a separate paragraph. We treat each paragraph as an individual document, which makes this dataset particularly retrieval-intensive. As a result, on average, each query is supported by 15 documents, enhancing the complexity and depth of the retrieval challenge. The detailed stats are reported in Table 1.

4 Experimental Setup

4.1 Datasets and Evaluation Metrics

Open-domain Summarization   In addition to ODSUM-WikiHow, we use the recent ODSUM-Story (Zhou et al., 2023) dataset in our experiments. In this dataset, inputs are queries and a corpus of documents, the task is to generate a summary by retrieving multiple relevant documents from the corpus. For evaluation, we use commonly used Rouge-L (R-L), BertScore (BS), NLI-based Entailment (Ent.) and Contradiction (Con.) scores (Liu et al., 2019). Additionally, we use UNIEVAL (Zhong et al., 2022) to assess the coherence, consistency, and relevance of the generated summaries.

Long-form QA   In this work, we use the ASQA (Stelmakh et al., 2022) dataset where inputs are ambiguous questions with multiple interpretations, and outputs should cover correct answers for all of them. To evaluate the performance on this dataset, we use a set of questions that requires 5 or more evidence passages. disambiguation metrics defined as a good long-form answer to an ambiguous question should contain short answers to all disambiguated questions as well as the context necessary to understand the source of ambiguity and the relationship between the short answers in (Stelmakh et al., 2022). EM (string Exact Match) is the fraction of disambiguations for which the corresponding short answer is present in the long answer. DA-F1 (Disambig-F1) is the fraction of disambiguated questions that can be answered from the predicted long answers. DR is the geometric mean of DA-F1 and ROUGE-L.

4.2 Baselines

RetGen retrieves passages a single time using the input query itself and then utilizes these passages

---

	ODSUM-Story							ODSUM-WikiHow						
	RL	BS	NLI		UniEval			RL	BS	NLI		UniEval		
			Ent.	Con.	Rel.	Coh.	Cons.			Ent.	Con.	Rel.	Coh.	Cons.
RetGen	15.19	48.19	10.75	44.52	64.02	63.41	64.91	27.51	60.17	18.64	11.67	81.71	81.11	80.55
IterRetGen	14.11	49.47	13.8	38.95	69.15	70.17	71.01	24.32	59.28	18.24	10.76	79.41	78.41	78.24
Self-Ask	17.29	50.88	12.99	24.68	85.5	85.64	86.23	28.99	60.66	22.79	7.98	82.97	82.13	82.97
SearChain	12.9	46.23	4.41	51.47	76.79	72.65	75.14	26.52	55.59	14.25	23.68	82.01	80.5	80.94
DSP	17.09	51.52	14.15	16.34	88.2	87.9	89	25.22	58.21	17.3	11.5	85.15	85.37	86.28
Self-RAG	17.81	50.16	11.73	16.26	55.81	52.6	54.81	19.77	56.24	21.35	7.76	83.95	84.07	83.78
ConTReGen	19.33	54.01	21.35	16.98	89.28	89.27	89.72	34.88	62.21	24.3	7.89	86.83	86.76	87.19

Table 2: ODSUM Performance Comparison

to generate the response.

IterRetGen (Shao et al., 2023) iteratively retrieves passages by using the previously generated response as the next query to retrieve passages in subsequent iterations and use them to update and refine the previously generated response.

Self-Ask (Press et al., 2023) utilizes an elicitive prompting technique that involves generating follow-up questions based on the previous retrieved knowledge. These follow-up questions are then used to fetch additional relevant passages and use them to generate intermediate responses.

SearChain (Xu et al., 2024) utilizes a structured, iterative interaction between LLMs and retriever to enhance reasoning in complex tasks. It dynamically generates and refines a chain of query-answer pairs that are verified through IR. Unlike our ConTReGen approach, which focuses on constructing tree-structured reasoning by exploring each query facet in-depth, it refines the generated CoQ at each round. However, the linear chain-of-query approach often fails to fully explore all relevant facets of a query, leading to potential gaps in the information retrieved.

DSP (Khattab et al., 2022) methodically handles complex queries by first demonstrating the desired query processing behavior, then searching relevant information through iterative decompositions of the query into simpler sub-queries, and finally generating the response by synthesizing all information.

Self-RAG (Asai et al., 2023b) leverages adaptive retrieval and self-reflection into the generation process. This approach allows the LLM to dynamically retrieve information when needed and use reflection tokens to assess and improve the relevance and factual accuracy of its outputs.

4.3 Implementation Details

In our experiments, we utilize LLAMA3-8B (AI@Meta, 2024) as the Large Language Model (LLM) for ConTReGen and all baselines, except

	ODSUM-Story	WikiHow-ODSUM	LFQA-ASQA
RetGen	16.14	13.99	36.56
IterRetGen	26.98	18.01	49.42
Self-Ask	36.79	22.3	51.5
DSP	27.61	14.94	53.39
Self-RAG	21.14	19.37	46.27
ConTReGen	57.69	42.44	57.38

Table 3: Retrieval Recall Performance

for Self-RAG. For Self-RAG, which requires fine-tuning of the LLM with additional reflection tokens, we use the provided trained LLAMA2-7B model. Across all methods, we use the pre-trained Contriever (Gautier et al., 2022) as the dense retriever, setting the number of retrieved passages to topk = 5 for ASQA and ODSUM-WikiHow datasets, and topk = 3 for ODSUM-Story. For iterative approaches, the maximum number of iterations is set to {5, 10}, while for ConTReGen, the maximum depth of the tree is set to 2. These settings are used as the default unless explicitly mentioned otherwise. We have used 1-shot prompting in both modules.

5 Experimental Results

5.1 Retrieval Performance

Table 3 presents a comparison of the retrieval recall performance between our proposed ConTReGen approach and existing baseline methods. RetGen performs a single round of context passage retrieval from the corpus, while the other baseline methods utilize iterative retrieval techniques. For these baselines, we evaluate performance after five iterations. The results demonstrate that ConTReGen significantly outperforms all baseline approaches. Notably, ConTReGen achieves substantial improvements in recall across various datasets: it shows a 20.9 points increase on ODSUM-Story, a 20.14 points increase on WikiHow-ODSUM, and a 3.99 points increase on ASQA, compared to the second-best methods.

	EM	DA-F1	DR	NLI	
				Ent.	Con.
RetGen	28.91	20.64	18.95	24.58	29.27
IterRetGen	35.52	24.56	25.82	33.82	19.85
Self-Ask	37.02	25.08	26.89	31.99	22.22
SearChain	28.26	13.47	18.99	7.79	53.89
DSP	36.22	24.59	27.03	29.91	**16.17**
Self-RAG	22.41	12.09	18.27	21.63	38.33
ConTReGen	**41.16**	**30.23**	**30.31**	**41.51**	21.4

Table 4: LFQA Performance on ASQA dataset

	R-L	BERTScore	NLI Entailment
RetGen	15.19	48.19	10.75
ConTRe w/o BSG	18.62	53.45	19.23
ConTReGen	**19.33**	**54.01**	**21.35**

Table 5: Ablation Study

	ODSUM-Story		ODSUM-WikiHow		LFQA-ASQA	
	ReP	NReP	ReP	NReP	ReP	NReP
IterRetGen	53.8	8.46	41.43	6.18	72.35	16.96
Self-Ask	61.47	21.74	50.11	8.28	77.49	14.72
DSP	47.49	13.8	36.3	4.17	76.36	13.69
Self-RAG	40.02	8.01	47.63	9.25	72.75	8.86
ConTReGen	**81.11**	**45.62**	**76.67**	**25.18**	**82.46**	**21.88**

Table 6: Retrieval Reasoning Analysis

5.2 ODSUM Performance

Table 2 illustrates the performance of ConTReGen compared to several baseline methods on open-domain summarization tasks: ODSUM-Story and ODSUM-WikiHow datasets. Our approach, ConTReGen, consistently outperforms other methods across multiple metrics, demonstrating its effectiveness in generating high-quality summaries.

ConTReGen achieves the highest scores in most metrics. It records a Rouge-L (RL) score of 19.33, BertScore (BS) of 54.01 on the ODSUM-Story dataset, and On ODSUM-WikiHow, RL score of 34.88, BS of 62.21 surpassing the second-best methods by 1.52, 2.49, 5.89 and 1.55 points, respectively. This improvement highlights ConTReGen's ability to accurately capture and reproduce more relevant information.

Going beyond token-level similarity-based metrics, we use NLI-based Entailment and Contradict scores to evaluate how generated summary sentences are logically aligned to the reference summary. ConTReGen excels with an entailment (Ent.) score of 21.35 on ODSUM-Story and 24.3 on ODSUM-WikiHow, outperforming all other baselines. It achieves competitive contradiction scores to Self-RAG. As Self-RAG utilizes the fine-tuned LLMs with reflection tokens to not only filter out the relevant passages but also to select best-generated text segment, it helps to improve generation quality with less contradictory information.

Furthermore, we use pre-trained QA-based UNIEVAL (Zhong et al., 2022) metric to evaluate the generated summary on multiple dimensions: Relevance (Rel.), Coherence (Coh.), and Consistency (Cons.). On both datasets, ConTReGen shows superior performance across all dimensions, indicating its effectiveness not only in retrieving relevant information but also in integrating it cohesively and consistently into the generated summaries.

5.3 LFQA Performance

As shown in Table 4, ConTReGen achieves the highest Exact Match (EM) score of 41.16 and DA-F1 score of 30.23, indicating its superior ability to recall short answers for more disambiguated questions. Furthermore, the larger DR score of 30.31 highlights the effectiveness of ConTReGen in terms of factual accuracy and text quality as this score combines both Rouge-L and DA-F1 scores. Additionally, ConTReGen achieves NLI entailment (Ent.) scores of 41.51, outperforming baselines by a large margin. However, DSP gets lower contradiction scores (Con.) than ConTReGen, because of DSP's advanced Chain-of-prompting compared to ConTReGen's one-shot prompting.

5.4 Ablation Study

Impact of Retrieval enhancements: To further analyze the contribution of retrieval in downstream tasks, we conducted an experiment on the ODSUM-Story dataset where we replaced our Bottom-up Synthesis and Generation (BSG) module with a simple generation module. This simple generation module takes the top K=3 passages retrieved by our retrieval system and generates the response once. We compared this performance with RetGen, which directly retrieves the top K=3 passages from the corpus index and generates a response. Table 5 shows that ConTRe w/o BSG outperforms the RetGen approach.

Impact of Bottom-up Synthesis and Generation: The structure of our retrieval systems encourages to leverage the BSG module in the generation process. In Table 5, we can see that BSG module improves the overall performance across all metrics.

RetGen	IterRetGen	Self-Ask	DSP	Self-RAG	ConTReGen
46.2	52.87	61.31	43.4	51.6	**80.65**

Table 7: Facet Coverage Analysis

5.5 Analysis

5.5.1 Retrieval Reasoning Analysis

We analyze the retrieval reasoning of iterative approaches and ConTReGen by examining the structural retrieval relationships among evidence passages. For each query and its associated evidence passages, we construct a relationship graph. In this graph, a directed edge from passage A to passage B (A→B) indicates that passage A can retrieve passage B from the corpus. The query itself acts as the root node, with edges leading from the query to an evidence passage (Q→A) if the query can directly retrieve Passage A. Based on this graph structure, we divide evidence passages into two categories: i) Reachable Passages (ReP), which are reachable from the root query through the graph, and ii) Non-Reachable Passages (NReP), which are not reachable. We assess and compare the recall performance for each category as detailed in Table 6. This analysis shows that ReP recall scores are significantly higher than those for NReP, illustrating that iterative approaches excel when there is direct retrieval connectivity among passages. However, these methods often fail to capture passages that lack explicit retrieval connectivity. Conversely, ConTReGen shows superior capability of retrieving both types of passage. This is because hierarchical planning not only ensures the effective utilization of retrieved knowledge to exploit explicit retrieval connectivity among passages but also establishes new implicit connections of passages.

5.5.2 Facet Coverage Analysis

To assess the facet coverage of input queries by each approach, we analyze a set of queries from the ODSUM-WikiHow dataset that require information from multiple methods, treating each method as a distinct facet. We considered a facet as covered if an approach successfully retrieves at least one passage corresponding to that facet. According to the results presented in Table 7, iterative approaches generally manage to retrieve passages from facets directly accessible by the query itself. In contrast, ConTReGen not only retrieves passages from aspects directly accessible by the query but also effectively covers additional aspects not directly retrievable by the query.

	EM	DA-F1	DR	NLI	
				Ent.	Con.
DSP	36.22	24.59	27.03	29.91	16.17
DSP w/ ConTRe	**50.34**	**28.57**	**32.09**	**33.81**	**14.75**
Self-RAG	22.41	12.09	18.27	21.63	38.33
Self-RAG w/ ConTRe	**37.58**	**17.8**	**23.25**	**26.63**	**34.73**

Table 8: Versatility of ConTRe (Retrieval only)

RetGen	IterRetGen	SelfAsk	SearChain	DSP	SelfRAG	ConTReGen
1	5	6	19.56	12	22	44.52

Table 9: Number of LLM call

5.5.3 Versatility of ConTRe

Our retrieval-only module, ConTRe, is designed to be compatible with any sophisticated generation module. Among the baseline approaches evaluated, only the DSP and Self-RAG frameworks employ distinct generation strategies different from ConTReGen. Specifically, DSP integrates Chain-of-Thought prompting into its generation process, while Self-RAG enhances its fine-tuned large language model (LLM) with additional reflection tokens to improve text generation quality. We integrated ConTRe with both the DSP and Self-RAG generation modules and reported results in Table 8 on ASQA. This integration of ConTRe significantly enhances performance in both cases, surpassing their respective default retrieval strategies.

5.6 Computation Overhead

Since the LLM is the most computationally intensive component, we measured the average number of LLM calls made by each method on the ODSUM-Story dataset, as shown in Table 9. Although ConTReGen requires more LLM or retrieval calls, it uses a smaller LLM (Phi-3, 3.8B) and still significantly outperforms all baselines with larger models (LLAMA3, 8B), achieving a +16.72 improvement in recall, as detailed in Table 11. Exploring the trade-offs between computational cost and task performance could be a promising direction for future work.

6 Related Works

Recently, considerable research has focused on Retrieval-Augmented Generation (RAG) across various NLP tasks (Gao et al., 2023). Existing approaches to single-time retrieval augmented generation typically involved retrieving knowledge directly from the input itself (Izacard and Grave, 2020; Lewis et al., 2020; Petroni et al., 2020), using expanded queries (Chuang et al., 2023), or rewrit-

ten queries (Ma et al., 2023). However, as these methods often struggle to retrieve all relevant passages in one go, recent developments have shifted towards iterative retrieval approaches. These involve LLMs actively interacting with the retrieval process by formulating contextually rich subsequent queries, which can include utilizing full or partial responses (Shao et al., 2023; Trivedi et al., 2022; Asai et al., 2023b; Jiang et al., 2023), or generating new queries (Press et al., 2023; Xu et al., 2024; Yao et al., 2022). Additionally, adaptive retrieval-augmented approaches have emerged, which decide when to retrieve based on token probability (Jiang et al., 2023), by generating explicit retrieval token (Asai et al., 2023a), or by leveraging the LLM's self-knowledge. No iterative approaches leverage hierarchical exploration of diverse facets of input queries in-depth. A recent paper MEMWALKER (Chen et al., 2023) constructs a tree structure from a given long text irrespective of queries. The model traverses the structure upon receiving an input query and generates the response, whereas our ConTReGen constructs a query-focused tree by retrieving the relevant information from a corpus. It focuses on understanding long text while ConTReGen focuses on both retrieving relevant contexts and leveraging them to generate the response. Similar to MEMWALKER (Chen et al., 2023), another recent paper (Sarthi et al., 2024) utilizes a bottom-up method to build a hierarchical tree by clustering and summarizing text chunks, transforming a flat corpus into a multi-layered one for varied-level information retrieval. However, this approach may struggle with very large retrieval corpora and highly diversified queries because the clustering and summarization processes are conducted independently of the queries. In contrast, ConTReGen employs a top-down strategy, systematically branching out from the main query into sub-queries that explore different facets and uses query-focused bottom-up summarization, enhancing relevance and effectiveness in information summarization without altering the retrieval corpus.

7 Conclusion

In this paper, we introduced ConTReGen, an innovative framework designed to enhance the capabilities of open-domain long-form text generation through a novel context-driven, tree-structured retrieval approach. ConTReGen organizes the re-

trieval process hierarchically, allowing for an in-depth exploration of various facets of input queries, and integrates a systematic bottom-up synthesis. This approach addresses the prevalent challenges in the field by ensuring comprehensive coverage and coherent integration of multi-faceted information. Additionally, our retrieval-only ConTRe module is model-agnostic and can be seamlessly adapted to various advanced generation techniques. Our extensive experiments across multiple datasets have demonstrated that ConTReGen significantly outperforms existing state-of-the-art retrieval-augmented generation models. These results not only validate the effectiveness of our approach but also underline its potential to be a versatile tool in enhancing the generation of contextually rich, accurate long-form content. Moreover, the introduction of the ODSUM-WikiHow dataset is particularly significant for furthering research in open-domain summarization.

Acknowledgments

Limitation

While ConTReGen has shown promising results in open-domain long-form text generation, its effectiveness is contingent upon the quality of the information retrieved. The model is susceptible to incorporating noisy or irrelevant documents, which can degrade the quality of the generated content. NLI contradiction scores of ConTReGen also highlight the need for improvement in managing irrelevant contexts. To mitigate this issue, explicitly trained generation models, such as recent works (Yoran et al., 2023; Asai et al., 2023b) could be employed

to enhance the robustness of the generation process against irrelevant contexts. Developing advanced generation techniques tailored for ConTReGen represents a promising direction for future research.

Moreover, the planning agent in ConTReGen, which is crucial for generating and organizing sub-queries, could benefit from explicit training. This training would enable the agent to adaptively generate plans and autonomously determine the necessary depth of exploration for each aspect of a query. Currently, ConTReGen may generate vague or non-informative sub-queries, and its two-step verification process relies heavily on the LLM's reasoning and existing knowledge base. Despite its impressive retrieval capabilities, there is significant potential for enhancing the system by implementing a more sophisticated model. Such a model would not only verify but also generate contextually rich, retrieval-effective sub-queries, thereby substantially improving both the accuracy and relevance of the retrieval process.

Furthermore, as this work focuses on long-form text generation, it would be an interesting future direction to explore tree-structured retrieval in single factoid answer scenarios.

References

AI@Meta. 2024. Llama 3 model card.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023a. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023b. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Valeriia Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. 2023. Wikihowqa: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5291–5314.

Odellia Boni, Guy Feigenblat, Guy Lev, Michal Shmueli-Scheuer, Benjamin Sznajder, and David Konopnicki. 2021. Howsumm: a multi-document summarization dataset derived from wikihow articles. *arXiv preprint arXiv:2110.03179*.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Yung-Sung Chuang, Wei Fang, Shang-Wen Li, Wen-tau Yih, and James Glass. 2023. Expand, rerank, and retrieve: Query reranking for open-domain question answering. *arXiv preprint arXiv:2305.17080*.

Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. 2021. Wikisum: Coherent summarization dataset for efficient human-evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 212–219.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

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

Izacard Gautier, Caron Mathilde, Hosseini Lucas, Riedel Sebastian, Bojanowski Piotr, Joulin Armand, and Grave Edouard. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Wang, and Arman Cohan. 2023. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8177–8199, Singapore. Association for Computational Linguistics.

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Mahnaz Koupaee and William Yang Wang. 2018. Wiki-how: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.

Carol C Kuhlthau. 1991. Inside the search process: Information seeking from the user's perspective. *Journal of the American society for information science*, 42(5):361–371.

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.

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

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Gary Marchionini. 1995. *Information seeking in electronic environments*. 9. Cambridge university press.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D Manning. 2019. Answering complex open-domain questions through iterative query generation. *arXiv preprint arXiv:1910.07000*.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*.

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*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. 2013. A survey of faceted search. *Journal of Web engineering*, pages 041–064.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*.

Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In *Proceedings of the ACM on Web Conference 2024*, pages 1362–1373.

Weiwen Xu, Yang Deng, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Exploiting reasoning chains for multi-hop science question answering. *arXiv preprint arXiv:2109.02905*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

Yijie Zhou, Kejian Shi, Wencai Zhang, Yixin Liu, Yilun Zhao, and Arman Cohan. 2023. Odsum: New benchmarks for open domain multi-document summarization. *arXiv preprint arXiv:2309.08960*.

	Topk = 3, Iterations = 10 | LLAMA3-8B as LLM				Topk = 5, Iterations = 10 | Phi-3-Mini-128K as LLM			
	R-L	BERTScore	NLI Entailment	Retrieval Recall	R-L	BERTScore	NLI Entailment	Retrieval Recall
RetGen	15.19	48.19	10.75	16.14	15.39	51.16	10.82	21.69
IterRetGen	13.12	48.94	14.82	29.03	14.66	50.86	14.85	37.95
Self-Ask	16.86	51.13	15.14	37.65	16.46	51.32	15.73	39.13
DSP	17.82	52.06	12.92	28.67	16.85	50.36	10.57	32.93
Self-RAG	18.96	47.96	9.37	23.24	17.78	47.83	11.34	29.28
ConTReGen	**19.33**	**54.01**	**21.35**	**57.69**	**18.36**	**53.42**	**18.15**	**53.51**

Table 10: Performance comparison by varying the number of iterations and Top k on ODSUM-Story dataset

RetGen	IterRetGen	SelfAsk	SearChain	DSP	SelfRAG	ConTReGen (Phi-3)	ConTReGen (LLAMA-3)
16.14	26.98	36.79	15.24	27.61	21.14	53.51	57.59

Table 11: Retrieval Performance of ConTReGen with different LLMs

A Appendix

A.1 Experimental results by varying hyperparameters

We analyzed the performance of several iterative retrieval techniques per iteration on ODSUM-Story and reported the results in Figure 3, which highlights that they are likely to reach a plateau of retrieval recall very quickly. Additionally, Table 10 shows the performance comparison across baselines after 10 iterations, while keeping the setting of ConTReGen as the number of subquestions in a plan is 5, and the depth of tree is 2.

A.2 Retrieval Performance by varying LLMs

Considering resource-constrained scenarios, we conducted further experiments using the lightweight Phi-3-Mini-4K LLM in our retrieval module. The recall performance on the ODSUM-Story dataset demonstrates that our Phi-3 retrieval significantly outperforms baseline approaches that use the LLAMA-3-8B model as shown in Table 11.

A.3 Error Analysis

We have conducted a detailed error analysis to identify scenarios where the input query might not be sufficiently informative to retrieve relevant reference passages from the corpus. Specifically, we have found 10 instances within the ODSUM-Story dataset where the input query, when used as the retriever's query, did not yield any evidence passages in the Top 32 results. In these challenging cases:

- All other baselines: Failed to retrieve any evidence passages in every case.

- **ConTReGen**: Successfully retrieved at least one evidence passage in 3 out of the 10 cases.

To further illustrate the robustness and reliability of ConTReGen, consider the following case where the original query is too vague considering the corpus:

Original Query: "What is the storyline of PRISON PLANET?"

Result: All approaches including retriever (topk=32) and ConTReGen failed to retrieve any evidence passages from the corpus.

Upon modifying the original query to include the author's name: **Modified Query:** "What is the storyline of PRISON PLANET by BOB TUCKER?"

Result:

- All baselines failed, except for DSP and Retriever (topk=32) retrieved just 1 evidence passage out of 9.

- **ConTReGen**: Retrieved 6 evidence passages out of 9.

The identified error cases underline a crucial aspect of query informativeness. In scenarios where the input query lacks informativeness, all approaches including ConTReGen face challenges in retrieving relevant passages. However, ConTReGen shows a notable improvement over all baselines when the query is slightly modified to include additional context, such as the author's name. This suggests that while ConTReGen is more robust and reliable, its performance can still be influenced by the informativeness of the input query.