Chronological Passage Assembling in RAG framework for Temporal Question Answering

Byeongjeong Kim, Jeonghyun Park, Joonho Yang, Hwanhee Lee[*]
Department of Artificial Intelligence, Chung-Ang University
{michael97k, tom0365, plm3332, hwanheelee}@cau.ac.kr

Abstract

Long-context question answering over narrative tasks is challenging because correct answers often hinge on reconstructing a coherent timeline of events while preserving contextual flow in a limited context window. Retrieval-augmented generation (RAG) indexing methods aim to address this challenge by selectively retrieving only necessary document segments. However, narrative texts possess unique characteristics that limit the effectiveness of these existing approaches. Specifically, understanding narrative texts requires more than isolated segments, as the broader context and sequential relationships between segments are crucial for comprehension. To address these limitations, we propose ChronoRAG, a novel RAG framework specialized for narrative texts. This approach focuses on two essential aspects: refining dispersed document information into coherent and structured passages, and preserving narrative flow by explicitly capturing and maintaining the temporal order among retrieved passages. We empirically demonstrate the effectiveness of ChronoRAG through experiments on the NarrativeQA dataset, showing substantial improvements in tasks requiring both factual identification and comprehension of complex sequential relationships, underscoring that reasoning over temporal order is crucial in resolving narrative QA.

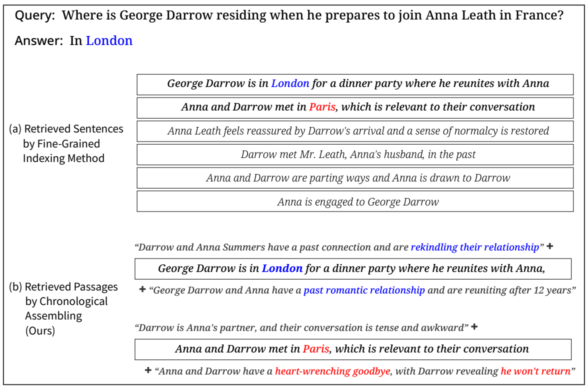

Query: Where is George Darrow residing when he prepares to join Anna Leath in France?
Answer: In London
(a) Retrieved Sentences by Fine-Grained Indexing Method
George Darrow is in London for a dinner party where he reunites with Anna
Anna and Darrow met in Paris, which is relevant to their conversation
Anna Leath feels reassured by Darrow's arrival and a sense of normalcy is restored
Darrow met Mr. Leath, Anna's husband, in the past
Anna and Darrow are parting ways and Anna is drawn to Darrow
Anna is engaged to George Darrow
(b) Retrieved Passages by Chronological Assembling (Ours)
"Darrow and Anna Summers have a past connection and are rekindling their relationship" +
George Darrow is in London for a dinner party where he reunites with Anna,
+ "George Darrow and Anna have a past romantic relationship and are reuniting after 12 years"
"Darrow is Anna's partner, and their conversation is tense and awkward" +
Anna and Darrow met in Paris, which is relevant to their conversation
+ "Anna and Darrow have a heart-wrenching goodbye, with Darrow revealing he won't return"

Figure 1: Retrieval comparison for a narrative query. (a) Fine-grained indexing returns six standalone sentences, leaving key clues detached. (b) Our chronological assembling retrieves passages that include their immediate chronological context, preserving the narrative flow. Boxes indicate the directly retrieved sentences.

1 Introduction

Long-context question answering tasks, which require the ability to utilize one or more long documents (Pang et al., 2022), present a significant challenge in natural language processing. While modern transformer-based Large Language Models (LLMs) have shown a remarkable ability to handle long contexts (Liu et al., 2025; Wang et al., 2024), they face fundamental limitations when confronted with extremely long-form text. Processing extensive documents for every query leads to major computational inefficiency, and as the context grows longer, the models' ability to accurately identify and prioritize relevant information decreases, impacting the reliability of their outputs.

To address these challenges, Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has become a standard approach, focusing on efficiently retrieving only relevant segments from large documents to integrate into the model's context window. This selective retrieval method helps models leverage vast knowledge bases far beyond their built-in context limits.

However, a fundamental methodological gap exists in most RAG frameworks (Lewis et al., 2020; Sarthi et al., 2024): they primarily treat documents as a collection of short, independently-retrieved snippets of information. This methodology fundamentally conflicts with the sequential nature of long-form narratives, such as those found in history, literature, and film. Narrative texts are uniquely defined by their structure; they can be **extremely**

[*] Corresponding Author.

long, their individual passages often fail to convey the full story unless **read in order**, and grasping the **chronological and relational connections between passages is essential** for comprehension. Treating passages as isolated facts severs these critical links, fragmenting the narrative timeline.

Figure 1 illustrates the mismatch between conventional retrieval strategies and the characteristics of narrative data. A common approach, as shown in (a) of Figure 1, is to retrieve as many sentences as possible that are likely to match the query, often based on textual similarity. To do so, documents are typically stored as isolated sentences. While such methods may successfully retrieve a sentence containing the correct answer, they often fail to provide sufficient contextual cues. For instance, this can create ambiguity, making it unclear whether "London" or "Paris" is the location relevant to the question, even if both are mentioned in the retrieved results.

To address this issue, we introduce ChronoRAG, a novel RAG-based approach that embodies an alternative strategy grounded in the principle that solving narrative-based problems fundamentally requires recognizing the chronological order of events. Instead of maximizing the number of retrieved sentences, our framework, as shown in (b) of Figure 1, retrieves fewer distinct informational units but includes their **surrounding context** to disambiguate meaning. This approach provides the crucial contextual clues—indicating that "London" is associated with a reunion while "Paris" pertains to a farewell—that are essential for accurate question answering. ChronoRAG achieves this by clarifying dispersed narrative content into structured passages and explicitly capturing the temporal relationships between them, enabling the retrieval of a coherent narrative flow rather than a collection of isolated facts.

We empirically validate our proposed approach on the NarrativeQA (Kočiskỳ et al., 2018) dataset. To rigorously test temporal reasoning, we isolate a subset of "Time Questions" that require understanding event sequences. Our experiments show that our method achieves significant improvements in both the complete dataset and the specialized Time Question set. Notably, these results are achieved using lighter graph construction and retrieval mechanisms than those found in existing summary and graph-based methods, demonstrating enhanced performance in identifying individual facts and comprehending complex relational structures.

- We find that resolving narrative QA requires leveraging event chronology and preserving contextual flow, which guides our method in distilling dispersed story elements into coherent, temporally aware passages.
- We introduce a novel RAG framework, ChronoRAG, which refines raw text into structured passages, explicitly maintains temporal links between events, and incorporates adjacent context.
- Experiments on the NarrativeQA dataset demonstrate the effectiveness of our framework, and emphasizing event-to-event relations drives performance gains for both factual and temporal queries, highlighting the critical role of relational understanding over entity extraction.

2 Related Work

Passage Granularity. Document indexing approaches have been explored with varying passage granularity to improve retrieval precision. DenseXRetrieval (Chen et al., 2024) advocates finer granularities to enhance information precision. Conversely, MolecularFacts (Gunjal and Durrett, 2024) demonstrates that overly granular decompositions such as atomic facts or propositions often lose critical contextual cues, advocating instead for concise yet contextually coherent units. Our method employs atomic facts as keys for retrieval while preserving broader narrative flows as the retrieved values.

Summary-Based Document Augmentation. Summary-based indexing methods, including RAPTOR (Sarthi et al., 2024), MemWalker (Chen et al., 2023), and ReadAgent (Lee et al., 2024) leverage iterative summarization to build hierarchical structures that improve retrieval accuracy and contextual coherence. However, such methods often suffer from high computational costs due to deep hierarchical structures and redundant overlapping information. Our approach simplifies the hierarchical concept by adopting a single-layer summary, significantly reducing computation and overlap issues while maintaining contextual effectiveness.

Knowledge Graph-Based Document Augmentation. Graph-based augmentation methods, such as GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024), typically construct knowledge graphs by extracting entities and relationships from documents. These methods are good at capturing entitycentric information but struggle to represent relationships between entities explicitly, a crucial element in narratives. Our proposed framework explicitly incorporates sequential relations among narrative elements, addressing this critical limitation of traditional knowledge graphs.

3 ChronoRAG

We present ChronoRAG, a novel Retrieval-Augmented Generation (RAG) framework specialized for narrative texts where chronological context is crucial. As described in Figure 2, our framework is composed of two primary stages: an offline **Graph Construction** phase where the original documents are processed into a hierarchical, linked structure, and an online **Passage Retrieval and Answer Generation** phase where the constructed graph is used to answer queries.

3.1 Graph Construction

The goal of this offline phase is to transform a raw document into a structured, two-layer graph that captures both factual information and narrative chronology. This process involves the following steps:

Document Chunking. Due to inherent limitations in handling entire documents simultaneously, we first divide the original document into fixed-length chunks (e.g., 100 tokens each). These chunks constitute *Layer 0* of our hierarchical graph, facilitating consistent and manageable retrieval. While fixed-length chunking may disrupt internal narrative coherence, maintaining a consistent token length is crucial for retrieval. Hence, we avoid semantic chunking with variable lengths.

Summarizing Chunks. Next, we group chunks (e.g., 10 chunks per group) and summarize them by instructing LLM. This summarization distills complex narrative passages, clarifying overall content and creating more manageable retrieval units.

Entity-Relation Extraction. Then we instruct the LLM to extract entities from summarized texts and generate relational descriptions between entities. We utilize only relation descriptions for indexing and retrieval, avoiding overlapping entity descriptions that might disrupt narrative flow. These relations, functioning like atomic facts, constitute *Layer 1* in the hierarchical graph, enhancing retrieval precision due to their focused and coherent informational structure.

Indexing. We assign narrative-order indices to both relation description sentences derived from summaries and to original document chunks. Relation descriptions are indexed according to the position of their source chunks in the document and the order in which the descriptions were generated, with lower indices assigned to those derived from earlier chunks or generated earlier in the process. Furthermore, each relation sentence stores the index of the original chunk it was derived from as a child index, enabling quick access to neighboring relations and original chunks.

Neighborhood Assembling. We augment retrieved relational descriptions with their surrounding context to reconstruct a narrative flow. Rather than relying on isolated facts, we aim to provide contextually rich information. Specifically, for *Layer 1* retrievals, we separate the role of the retrieved item into a key (the relation description itself) and a value (neighboring *Layer 1* passages concatenated in index order). This approach allows us to preserve a coherent local storyline rather than referencing fragmented facts.

3.2 Retrieving Passage

At inference time, a query is handled through a hierarchical retrieval process that leverages the constructed graph to assemble a rich, chronologically-aware context for the LLM.

Hierarchical Retrieving. We leverage the hierarchical granularity of *Layer 1* and *Layer 0* for retrieval. We begin by retrieving high-precision relation descriptions from *Layer 1*. Then, using the associated child indices, we retrieve related *Layer 0* chunks, ensuring a comprehensive and balanced context. This is crucial because Layer 0 often retains omitted details and original dialogues that are valuable for question answering.

Answer Generation. We combine the original query with the context obtained through hierarchical retrieval and feed them into the language model. Each passage is separated by double line breaks and ordered by relevance, enabling accurate and coherent answer generation.

4 Experiments

4.1 Experimental Setup

Dataset. We employ the NarrativeQA (Kočiskỳ et al., 2018) as our primary dataset. NarrativeQA comprises 355 stories and scripts with a total of 10,557 question–answer pairs. From this pool, we identify and separate a subset of 1,111 Time Ques-

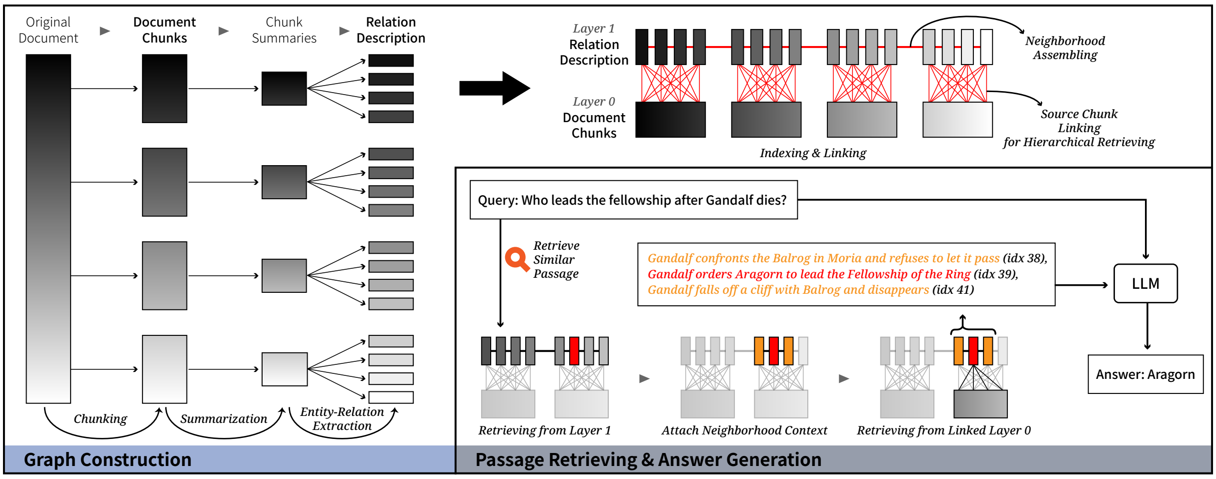

Original Document
Document Chunks
Chunk Summaries
Relation Description
Chunking
Summarization
Entity-Relation Extraction
Graph Construction
Layer 1
Relation Description
Layer 0
Document Chunks
Neighborhood Assembling
Source Chunk Linking for Hierarchical Retrieving
Indexing & Linking
Query: Who leads the fellowship after Gandalf dies?
Retrieve Similar Passage
Gandalf confronts the Balrog in Moria and refuses to let it pass (idx 38),
Gandalf orders Aragorn to lead the Fellowship of the Ring (idx 39),
Gandalf falls off a cliff with Balrog and disappears (idx 41)
LLM
Answer: Aragorn
Retrieving from Layer 1
Attach Neighborhood Context
Retrieving from Linked Layer 0
Passage Retrieving & Answer Generation

Figure 2: Overall Pipeline of ChronoRAG.

tions, defined as those containing temporal keywords {"When," "While," "During," "After," "Before"}. These Time Questions require retrieving and reasoning over multiple related events, making them a particularly challenging subset for temporal understanding and reasoning.

Evaluation Metric. We measure answer quality using ROUGE-L (Lin, 2004), which computes the Longest Common Subsequence overlap between a generated answer and its corresponding human reference. Due to the short and pronoun-heavy nature of NarrativeQA answers, ROUGE-L effectively captures agreement in key word sequences without penalizing minor rephrasings.

Baselines. We compare against five existing methods that differ in information extraction, representation, and retrieval structure:

- **NaiveRAG:** A standard RAG pipeline that performs chunk-level retrieval only, without further structuring (Lewis et al., 2020).
- **RAPTOR:** Clusters semantically similar chunks via embedding similarity and builds a recursive summarization tree over clusters to guide retrieval—CT (Collapsed Tree) flattens each root-to-leaf path into one high-level summary, whereas TT (Tree Traversal) retains the full hierarchy and drills down level-by-level to gather finer-grained context (Sarthi et al., 2024).
- **LightRAG:** Constructs a lightweight entity–relation graph to enable fast context retrieval using dual-level extraction, prioritizing computational efficiency and incremental updates (Guo et al., 2024).
- **GraphRAG:** Builds a richer graph with detailed relation weighting and neighborhood assembly to support deeper multi-hop retrieval, capturing both high-level relation summaries and their underlying chunks (Edge et al., 2024).
- **Propositionizer:** Transforms the entire source text into fine-grained propositions (atomic sentences) and treats each proposition as a retrieval unit, then feeds retrieved propositions into the generation model (Chen et al., 2024).

Implementation Details. All baselines share identical hyperparameter settings: top-k of 20 for retrieval, contextTokenLengthLimit of 1,500 tokens, and the same sampling strategy during generation. We perform all summarization and entity–relation extraction steps with meta-llama-3-8B-Instruct (Grattafiori et al., 2024). We compute retrieval scores using embedding similarity exclusively; we don't use BM25 (Robertson et al., 2009) to prevent distortion of the original text during generation. Specifically, we employ the arctic-Snowflake-embed-l (Merrick et al., 2024) for generating embeddings, and use unifiedqa-v2-t5-3b-1363200 (Khashabi et al., 2022) for final answer generation. All retrieved contexts fed into the generator respect the 1,500-token length limit to ensure fair comparison across methods.

4.2 Main Results

Performance Comparison. Table 1 shows that ChronoRAG, our proposed approach outperforms on both the full NarrativeQA full dataset and the Time Question subset compared to baselines. Summarization based baselines such as RAPTOR-CT and RAPTOR-TT follow, but they still lag. The results indicate that restructuring events into a clear temporal order supplies the language model with the most coherent context for narrative reasoning and question answering. GraphRAG records the lowest score among all methods for several reasons. Its exhaustive entity–relation extraction adds thousands of trivial nodes, inflating the graph and burying key plot elements under noise. And it omits the covariate filtering stage, so graph expansion begins from noisy entity-relations and quickly drifts into irrelevant subgraphs. These compounded issues dilute precision so severely that GraphRAG scores lowest.

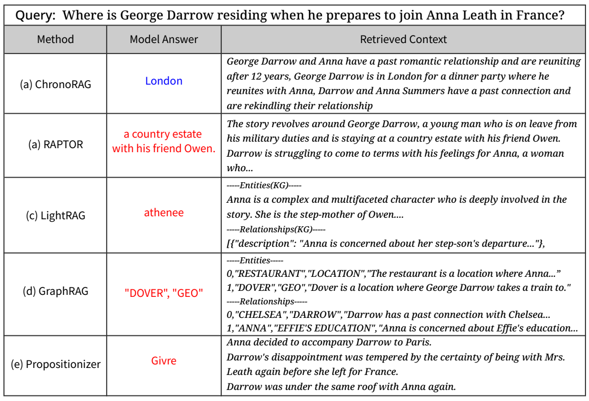
Method	Whole Data	Time Question
NaiveRAG	0.255	0.227
Propositionizer	0.262	0.238
RAPTOR_CT	0.297	0.261
RAPTOR_TT	0.295	0.259
LightRAG	0.240	0.214
GraphRAG	0.200	0.185
ChronoRAG	**0.308**	**0.268**

Table 1: QA Performance on NarrativeQA (ROUGE-L). Top performance is bolded, Second best is underlined.

Ablation Study. In this section, we conduct ablation studies to investigate the effectiveness of different components and settings of ChronoRAG. As shown in the Table 2, without summarizing the original text and extracting entity relations shows a significant performance degradation, showing the importance of chunk summarization. The effects of summarization are twofold: it leaves only important information, making retrieval easier, and when assembling, it clarifies the flow. The results without passage assembling are obtained by individually searching for entity relations extracted from the summary, while the results without chunk summarization are obtained by searching for entity relations directly extracted from the 10 chunks. Despite not connecting nearby passages in both settings, a significant performance difference is observed in the TimeQuestion.

Ablation	Whole Data	Time Question
ChronoRAG	**0.308**	**0.268**
w/o Passage Assembling	0.295	0.252
w/o Chunk Summarization	0.272	0.233
w/o Relation Extraction	0.255	0.227

Table 2: Ablation Study on NarrativeQA (ROUGE-L)



Query: Where is George Darrow residing when he prepares to join Anna Leath in France?		
Method	Model Answer	Retrieved Context
(a) ChronoRAG	London	*George Darrow and Anna have a past romantic relationship and are reuniting after 12 years, George Darrow is in London for a dinner party where he reunites with Anna, Darrow and Anna Summers have a past connection and are rekindling their relationship*
(a) RAPTOR	a country estate with his friend Owen.	*The story revolves around George Darrow, a young man who is on leave from his military duties and is staying at a country estate with his friend Owen. Darrow is struggling to come to terms with his feelings for Anna, a woman who...*
(c) LightRAG	athenee	*-----Entities(KG)-----* *Anna is a complex and multifaceted character who is deeply involved in the story. She is the step-mother of Owen....* *-----Relationships(KG)-----* *[{"description": "Anna is concerned about her step-son's departure..."},*
(d) GraphRAG	"DOVER", "GEO"	*-----Entities-----* *0,"RESTAURANT","LOCATION","The restaurant is a location where Anna..."* *1,"DOVER","GEO","Dover is a location where George Darrow takes a train to."* *-----Relationships-----* *0,"CHELSEA","DARROW","Darrow has a past connection with Chelsea...* *1,"ANNA","EFFIE'S EDUCATION","Anna is concerned about Effie's education...*
(e) Propositionizer	Givre	*Anna decided to accompany Darrow to Paris.* *Darrow's disappointment was tempered by the certainty of being with Mrs. Leath again before she left for France.* *Darrow was under the same roof with Anna again.*

Figure 3: Retrieved context per method.

Trade-off between Linking Window and the Number of Retrieved Passage. While extending the connection beyond adjacent text segments can enhance the local contextual coherence of retrieved passages, it also increases the length of each segment, thereby reducing the total number of passages that can be retrieved within a fixed token limit. Our experiments revealed that this trade-off has a detrimental effect on retrieval quality. when the number of adjacent sentences included in each passage was increased to two, overall and temporal questions performance declined to 0.300 and 0.258, respectively.

Computation Costs. Our pipeline is computationally efficient, requiring just two LLM calls per 1,000 tokens for graph construction. Although this cost increases linearly with document length, it remains lower than competing methods like recursive summarization. Furthermore, only one LLM call is required for answer generation during search, with our method still attaining the highest performance despite its efficiency.

Case Study. Figure 3 presents excerpts of the original passages retrieved by each method for the example shown in Figure 1. RAPTOR retrieves summary passages, which enables access to content covering a wide range of information. However, these summaries frequently include information that is not pertinent to the query, or conversely, omit critical details necessary for answering the question due to length constraints imposed by the summarization process. LightRAG and GraphRAG extract entities and relations directly from the original text. In particular, GraphRAG was found to underperform compared to direct retrieval to the source chunk, likely due to its tendency to include exhaustive explanations of all elements. Propositionizer and LightRAG offer relatively general-level granularity explanations, yet they still struggle to address questions that require understanding the changes in the relationship between Anna and George. In contrast, ChronoRAG identifies the minimal set of chronologically adjacent passages while suppressing unrelated narrative details, illustrating its strength in maintaining temporal coherence and reducing retrieval noise.

5 Conclusion

We present ChronoRAG, an RAG Framework that can effectively and efficiently handle narrative text. Our framework refines content through summarization and relation extraction, and improves overall performance through simple passage augmentation that connects adjacent events via an index. This suggests that it is important not only to organize individual events and elements in narrative texts but also to connect events that are spatially and temporally close to each other.

Acknowledgement

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)] and the Chung-Ang University Graduate Research Scholarship in 2023 and was improved by the helpful input and collaboration of researchers from LG AI Research.

References

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*.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense x retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Anisha Gunjal and Greg Durrett. 2024. Molecular facts: Desiderata for decontextualization in llm fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3751–3768.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. Unifiedqa-v2: Stronger generalization via broader cross-format training. *arXiv preprint arXiv:2202.12359*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the 41st International Conference on Machine Learning*, pages 26396–26415.

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.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *arXiv preprint arXiv:2405.05374*.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. 2022. Quality: Question answering with long input texts, yes! In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 5336–5358. Association for Computational Linguistics (ACL).

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.