
RETHINKING CHUNK SIZE FOR LONG-DOCUMENT RETRIEVAL: A MULTI-DATASET ANALYSIS

Sinchana Ramakanth Bhat

Fraunhofer IAIS

Germany

sinchana.ramakanth.bhat@iais.fraunhofer.de

Max Rudat

Fraunhofer IAIS

Germany

max.rudat@iais.fraunhofer.de

Jannis Spiekermann

Fraunhofer IAIS

Germany

jannis.spiekermann@iais.fraunhofer.de

Nicolas Flores-Herr

Fraunhofer IAIS

Germany

Nicolas.Flores-Herre@iais.fraunhofer.de

June 10, 2025

ABSTRACT

Chunking is a crucial preprocessing step in retrieval-augmented generation (RAG) systems, significantly impacting retrieval effectiveness across diverse datasets. In this study, we systematically evaluate fixed-size chunking strategies and their influence on retrieval performance using multiple embedding models. Our experiments, conducted on both short-form and long-form datasets, reveal that chunk size plays a critical role in retrieval effectiveness - smaller chunks (64-128 tokens) are optimal for datasets with concise, fact-based answers, whereas larger chunks (512-1024 tokens) improve retrieval in datasets requiring broader contextual understanding. We also analyze the impact of chunking on different embedding models, finding that they exhibit distinct chunking sensitivities. While models like Stella benefit from larger chunks, leveraging global context for long-range retrieval, Snowflake performs better with smaller chunks, excelling at fine-grained, entity-based matching. Our results underscore the trade-offs between chunk size, embedding models, and dataset characteristics, emphasizing the need for improved chunk quality measures, and more comprehensive datasets to advance chunk-based retrieval in long-document Information Retrieval (IR).

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a powerful paradigm in natural language processing (NLP), enabling large language models (LLMs) to enhance response accuracy by incorporating relevant external knowledge retrieved from document corpora [1] [2]. This approach has significantly improved performance in knowledge-intensive tasks by mitigating the limitations of parametric memory in LLMs and enhancing factual consistency [3]. The effectiveness of RAG systems heavily depend on document chunking strategies, which segment textual data into manageable units before retrieval. Among various chunking techniques, fixed-size token-based chunking remains a prevalent method due to its simplicity and ease of implementation [4]. Fixed-size chunking segments documents into uniform token-length chunks, ensuring compatibility with transformer-based architectures that have strict token limits [5].

However, despite its widespread adoption, the robustness of this approach across varied document lengths remains underexplored. The effectiveness of fixed-size chunking can be influenced by factors such as information dispersion across chunks, redundancy in retrieval, and retrieval precision [6]. Understanding these factors is crucial to optimizing retrieval performance and downstream generation quality in RAG applications.

The importance of dataset characteristics in retrieval performance has been well established. Some datasets exhibit high answer locality, where relevant spans are concentrated within a few sentences, while others require reasoning over long

Table 1: Overview of dataset statistics, including key properties. Datasets that were stitched together to create longer synthetic documents are marked with an asterisk (*) next to their names.

Dataset	# Docs	Q/Doc	Tokens/Doc	Tokens/Q	Tokens/A	Unique Tokens/Doc
NarrativeQA	1073	1.9	51830.4	8.5	12.6	8983.5
Natural Questions (NQ)	25010	1.0	6918.2	9.0	6.8	2864.8
NewsQA*	685	11.9	8484.5	6.5	5.2	3265.3
COVID-QA*	55	2.1	10009.2	8.8	11.1	2904.5
TechQA*	45	10.6	7597.2	51.1	46.9	2130.1
SQuAD*	306	43.7	7998.3	9.9	3.9	2949.3

contexts spanning multiple chunks [7]. Recent studies have highlighted that datasets with diverse document structures demand adaptive chunking strategies to optimize retrieval effectiveness [8]. Additionally, the role of embedding models in retrieval varies - encoder-based models tend to benefit from exact term matching, whereas decoder-based models leverage broader contextual cues [9].

Despite its widespread use, fixed-size chunking may not be optimal for all datasets. For instance, hierarchical chunking methods have been shown to improve retrieval by preserving semantic coherence within chunks [10]. Moreover, the challenge of evaluating chunking strategies stems from the difficulty of establishing ground truth relevance in retrieval [11]. While prior work has explored chunking in traditional retrieval systems, there is limited research on how fixed-size chunking impacts retrieval performance across documents of varying domains in modern RAG architectures. We address this research gap through a structured set of ablations on different domain-specific datasets across different chunk sizes and embedding models. For reproducibility of results, our code is available on Github ¹. Our key contributions can be summarized as follows:

- We systematically evaluate the impact of fixed-size chunking on retrieval effectiveness across multiple datasets, analyzing recall@k trends for different chunk sizes and embedding models.
- We investigate how document length, domain, and dataset diversity influence retrieval performance, providing insights into chunk size selection for different question-answering scenarios.
- We explore retrieval biases across different embedding models, highlighting how chunking strategies interact with model architectures to optimize retrieval outcomes of chunk size.

2 Related Work

Chunking Strategies: The effective segmentation of long documents into manageable chunks is a persistent challenge in information retrieval. We recognize the advancements in dynamic chunking [12] and neural-integrated chunking strategies [7], however, fixed-size chunking remains a widely used and foundational approach due to its simplicity and computational efficiency. Furthermore, understanding the behavior and limitations of fixed-size chunking provides a crucial baseline for evaluating the effectiveness of more complex methods. Therefore, we conduct a comprehensive empirical analysis of fixed-size token chunking across a diverse range of modern question answering datasets.

Embedding Models: Transformer-based embedding models form the backbone of modern retrieval systems, encoding text into dense representations for similarity-based retrieval. Early works leveraged pre-trained models like BERT [13] and RoBERTa [14] for retrieval, while more recent approaches introduced contrastive learning and hard negative mining to improve retrieval quality [15] [16]. Dense retrieval models such as DPR [15] and Contriever [17] have demonstrated significant improvements over sparse retrieval methods by leveraging deep contextualized representations. However, embedding models exhibit distinct retrieval biases based on their architectural design. Positional encoding mechanisms play a crucial role in how transformers interpret document structure and chunked inputs. Standard transformers use absolute or sinusoidal positional encodings [18], while newer approaches like Rotary Positional Encoding (RoPE) [19] and Attention with Linear Biases (ALiBi) [20] modify the handling of positional information. These encodings impact retrieval behavior, especially in chunk-based retrieval settings.

Dataset Considerations in Chunk-Based Retrieval: Understanding the characteristics of question answering datasets is vital for developing robust and generalizable retrieval systems. Recent studies have emphasized the importance of analyzing dataset biases and evaluating model performance across diverse question types [21]. Furthermore, researchers have explored the creation of challenging benchmark datasets that require reasoning over long contexts and multiple

¹Code and interface description: <https://github.com/fraunhofer-iaais/chunking-strategies>

documents [22]. We extend this line of research by conducting a detailed analysis of dataset characteristics and examining their correlation with retrieval performance. Additionally, we investigate the impact of stitching QA pairs within small datasets to form long documents, providing insights into real-world retrieval challenges. This approach is inspired by recent work that has demonstrated the feasibility of stitching datasets for similarity-based evaluations [11]

3 Methodology

Chunking and Retrieval We employ a RAG system built using *LlamaIndex* [23] to evaluate the impact of chunk size on retrieval performance across various datasets. Our retrieval process begins with fixed-size token splitting, where each document in each dataset is segmented into chunks of predetermined lengths using *LlamaIndex’s TokenTextSplitter*. This method ensures consistent chunk sizes, allowing us to isolate the effect of chunk length on retrieval effectiveness. The system first embeds all document chunks using a pre-trained embedding model. Subsequently, a user query is embedded using the same model. To retrieve the most relevant chunks, cosine similarity is calculated between the query embedding and each chunk embedding. The top k chunks with the highest cosine similarity scores are then selected as the retrieved context.

Datasets Our study utilizes a diverse set of long extractive question answering datasets, including NarrativeQA [24], Natural Questions (NQ) [25], NewsQA [26], COVID-QA [27], TechQA [28], and SQuAD [29]. Table 1 summarizes the key statistics for each dataset.

To ensure the relevance of our evaluation, especially since the challenge is in finding ground truth data for chunking, we filtered documents by performing a string match comparison between the expected answer and the document text. This step guarantees that the answer is present within the document, focusing our analysis on retrieval accuracy. We also performed cleaning and filtering checks to ensure that each answer appears only once in the document, avoiding ambiguity in our evaluation.

A significant challenge in evaluating long RAG datasets is the scarcity of datasets with sufficiently long documents. To address this, we stitched together shorter QA datasets to create longer synthetic documents, ensuring a minimum length of 50,000 characters.

4 Evaluation

We evaluate retrieval performance across multiple datasets using a fixed-size chunking strategy. The primary metric is Recall@ k , where a retrieval is considered successful if the relevant chunk appears within the top k results. While we report full Recall@ k values ($k = \{1, 2, 3, 4, 5\}$), the trends remain consistent across different k , so we primarily focus on Recall@1 in our analysis for brevity. Notable deviations or patterns at higher k are highlighted where relevant.

Each dataset varies in question length, document length, answer locality, and vocabulary diversity, which influences chunking effectiveness (Table 1). To account for these differences, we experiment with multiple fixed chunk sizes: 64, 128, 256, 512, and 1024 tokens using *stella_en_1.5B_v5* [1] and *snowflake-arctic-embed-l-v2.0* [30] as embedding models. These experiments were conducted without overlapping tokens.

4.1 Impact of Chunk Size on Retrieval Performance

Chunk size plays a significant role in retrieval effectiveness across datasets, impacting how well relevant spans are captured. Our results indicate that smaller chunks (64-128 tokens) perform best for datasets with short, fact-based answers, whereas larger chunks (512-1024 tokens) are necessary for datasets with descriptive or technical responses. For example, in SQuAD, which has feature concise, entity-based answers, 64-token chunks yield the highest recall@1 which is 64.1%. However, increasing chunk size reduces recall by 10-15% at 512 tokens, likely due to excessive context introducing noise. Conversely, NewsQA, with its entity-heavy questions, achieves peak recall@1 at 512 tokens (55.9%), suggesting that moderate context expansion enhances retrieval without overwhelming the model. In datasets with long and dispersed answers, such as NarrativeQA, NQ and TechQA, larger chunks significantly improve performance. NarrativeQA recall@1 increases from 4.2% (64 tokens) to 10.7% (1024 tokens), highlighting the need for broader context to capture dispersed answer locations. Also, NQ dataset shows similar pattern with highest recalls at 512 and 1024 token lengths respectively. Similarly, TechQA recall@1 improves from 16.5% (128 tokens) to 61.3% (512 tokens), demonstrating that wider retrieval windows benefit technical domains where context is essential. For the COVID-QA dataset, which comprises domain-specific biomedical texts, retrieval performance varies notably with chunk size. While Stella achieves its highest Recall@1 (52.1%) at 64 tokens, performance gradually declines with larger chunks, suggesting that smaller, focused spans are more effective for this model. In contrast, Snowflake shows a steady improvement, peaking at 1024 tokens with Recall@1 of 54.2% and Recall@5 of 80.2%, indicating its stronger ability to leverage extended context without losing relevance.

Table 2: Retrieval performance (R@K = Recall@K) across different dataset chunk sizes for Stella and Snowflake models.

Dataset	Chunk Size	Stella					Snowflake				
		R@1	R@2	R@3	R@4	R@5	R@1	R@2	R@3	R@4	R@5
NarrativeQA	64	0.0420	0.0685	0.0880	0.1001	0.1079	0.0343	0.0539	0.0712	0.0831	0.0882
	128	0.0571	0.0811	0.1044	0.1260	0.1374	0.0469	0.0701	0.0950	0.1100	0.1204
	256	0.0790	0.1217	0.1510	0.1697	0.1887	0.0616	0.0975	0.1189	0.1469	0.1657
	512	0.0894	0.1360	0.1749	0.1978	0.2263	0.0894	0.1333	0.1657	0.1866	0.2025
	1024	0.1071	0.1670	0.2109	0.2432	0.2695	0.1041	0.1653	0.2033	0.2287	0.2529
Natural Questions (NQ)	64	0.1718	0.2665	0.3338	0.3861	0.4255	0.1500	0.2330	0.2986	0.3464	0.3873
	128	0.2220	0.3474	0.4295	0.4960	0.5404	0.2040	0.3232	0.4136	0.4808	0.5355
	256	0.3228	0.4900	0.6043	0.6835	0.7435	0.2634	0.4294	0.5499	0.6333	0.6923
	512	0.3854	0.5539	0.6689	0.7448	0.7954	0.3895	0.5817	0.6937	0.7635	0.8125
	1024	0.3493	0.5561	0.6642	0.7273	0.7720	0.4774	0.6835	0.7937	0.8581	0.8982
NewsQA*	64	0.3780	0.5415	0.6328	0.6931	0.7419	0.3204	0.4596	0.5468	0.6083	0.6532
	128	0.4395	0.6157	0.7179	0.7859	0.8280	0.4156	0.5801	0.6736	0.7334	0.7760
	256	0.4906	0.6901	0.7978	0.8677	0.9021	0.4448	0.6328	0.7377	0.8007	0.8370
	512	0.5595	0.7869	0.8756	0.9078	0.9276	0.5734	0.7716	0.8398	0.8740	0.8974
	1024	0.5202	0.6639	0.7235	0.7663	0.7998	0.6668	0.8192	0.8765	0.9110	0.9320
COVID-QA*	64	0.5212	0.6354	0.6884	0.6975	0.7157	0.3899	0.5278	0.5748	0.6354	0.6672
	128	0.4181	0.5545	0.6551	0.6884	0.6975	0.5324	0.6460	0.7203	0.7475	0.7672
	256	0.4242	0.5921	0.6648	0.7087	0.7315	0.4087	0.5375	0.6815	0.7709	0.8072
	512	0.4060	0.5581	0.6642	0.7390	0.7578	0.4318	0.5739	0.6951	0.7315	0.7487
	1024	0.3075	0.4960	0.6684	0.7442	0.7715	0.5421	0.6709	0.7366	0.7745	0.8018
TechQA*	64	0.0486	0.0763	0.1089	0.1413	0.1497	0.0445	0.0771	0.0982	0.1232	0.1418
	128	0.1650	0.2740	0.3406	0.3680	0.3906	0.1838	0.2848	0.3337	0.3703	0.3885
	256	0.3995	0.5473	0.6059	0.6717	0.6896	0.4284	0.5826	0.6553	0.6799	0.6838
	512	0.6138	0.7482	0.8075	0.8677	0.8805	0.5811	0.7335	0.7866	0.8213	0.8405
	1024	0.6192	0.7020	0.7270	0.7834	0.8127	0.7154	0.8335	0.8801	0.8980	0.9107
SQuAD*	64	0.6419	0.7746	0.8263	0.8562	0.8742	0.6087	0.7424	0.7955	0.8277	0.8501
	128	0.6162	0.7542	0.8116	0.8485	0.8712	0.6000	0.7374	0.8032	0.8397	0.8667
	256	0.5662	0.7103	0.7794	0.8237	0.8506	0.5388	0.6839	0.7583	0.8034	0.8336
	512	0.4979	0.6518	0.7350	0.7887	0.8263	0.4621	0.6135	0.7023	0.7644	0.8064
	1024	0.3855	0.5171	0.6001	0.6596	0.7113	0.4294	0.6044	0.7062	0.7823	0.8368

4.2 Dataset-Specific Chunking Performance

The impact of chunk size varies significantly across datasets due to differences in document structure, question complexity, answer locality and other factors, which can be inferred from Table 1. Here, we analyze retrieval performance for each dataset in light of these characteristics.

In **NarrativeQA**, which consists of long, unstructured texts (51830.4 tokens/document), relevant answer spans are often far from the document part that is semantically similar to the question, making small chunks ineffective (recall@1 of 4.2% at 64 tokens). Performance improves significantly with larger chunks (10.7% at 1024 tokens), indicating the need for broader context to capture dispersed answer locations. In contrast, **NewsQA**, with shorter, well-structured news articles (8484.5 tokens/document) and a higher number of questions per document (11.9 on average), performs reasonably well even with smaller chunks across all recalls (37.8% recall@1 at 64 tokens) and steadily increases with increasing chunk sizes. The structured nature of the text allows relevant spans to be retrieved even with compact representations. In **NQ**, the chunk size impact is more gradual. Despite relatively short documents (6918.2 tokens/document) and brief answers (6.8 tokens on average), the retrieval performance improves steadily with chunk size, peaking at 1024 tokens for both models. This behavior likely stems from the naturalistic structure of web-sourced documents and the variability in

question focus, which often requires incorporating broader context to locate the correct span. Compared to datasets like SQuAD, where answers are tightly localized, NQ exhibits a more distributed answer locality, benefiting from moderate to large chunks without suffering from excessive noise.

For highly domain-specific datasets like TechQA (7597.2 tokens/document) and CovidQA (10009.2 tokens/document), chunking behavior differs due to the nature of the content. **TechQA** contains long, explanation-heavy answers (46.9 tokens on average) and highly structured text, leading to poor recall at small chunk sizes but substantial improvement as chunk sizes increase (recall@1 rises from 4.8% at 64 tokens to 71.5% at 1024 tokens). This suggests that larger chunks are necessary to capture sufficient technical context. In **COVID-QA**, retrieval performance is shaped by the nature of biomedical literature, moderately long documents (10,009.2 tokens/doc) with relatively verbose answers (11.1 tokens on average) and moderate lexical diversity (2904.5 unique tokens/doc). Compared to datasets like SQuAD, which has very short, factual answers (3.9 tokens) and benefits from compact chunks, COVID-QA spans require more context to capture domain-specific phrasing and nuanced biomedical references. This is reflected in Snowflake’s consistent improvement with larger chunks, peaking at 1024 tokens (Recall@1 = 54.2%). However, Stella achieves its highest Recall@1 (52.1%) at just 64 tokens, with diminishing returns as chunk size increases, suggesting that it may struggle with noise introduced by extended context. This divergence indicates that while COVID-QA requires moderately sized context windows to retrieve relevant spans, the optimal chunk size is also heavily influenced by model capabilities. Broader context is useful, but only when the retriever can effectively parse dense biomedical language without being overwhelmed by irrelevant content.

In **SQuAD**, a well-structured Wikipedia-based dataset with a high density of factual questions (43.7 questions per document), retrieval is less sensitive to chunk size. Even at 64 tokens, recall@1 remains high (64.1%), increasing marginally with larger chunks. This suggests that smaller chunks are sufficient due to the concise nature of the spans that contain the answers (3.9 tokens on average) and the structured format of Wikipedia text.

4.3 Chunking Impact Across Different Embedding Models

Chunking interacts differently with retrieval models, influencing their ability to retrieve relevant spans at varying chunk sizes. As highlighted in Section 4.2 with COVID-QA, chunk size interacts not only with dataset characteristics but also with the architecture of the underlying retrieval model. Different models exhibit varying sensitivities to chunk size, shaped by their design and pretraining context windows. Stella is a decoder-based model, whereas Snowflake is encoder-based. Hence, they exhibit distinct behaviors due to differences in their training and embedding structures. Stella demonstrates stronger performance at larger chunk sizes (512-1024 tokens), improving recall@1 by 5-8% compared to Snowflake in long-document datasets (NarrativeQA, NQ, TechQA). This suggests that Stella benefits from global chunk context, leveraging its large context window of more than 130,000 tokens and corresponding training on similarly long input texts (in the pretraining of its base model Qwen2 [31]). However, at smaller chunk sizes (64-128 tokens), Stella’s performance declines, likely due to loss of surrounding context, which weakens retrieval precision. Conversely, Snowflake maintains competitive performance on small-chunk datasets (SQuAD, CovidQA), where recall@1 remains within 1-2% of Stella. This highlights its strength in capturing fine-grained entity relationships and handling shorter context windows efficiently. However, its retrieval effectiveness deteriorates at larger chunk sizes, possibly because its pre-training objectives favor local token interactions, likely due to its shorter context window of 8194 tokens. These findings underscore that chunking effectiveness is model-dependent. While some models excel at long-range retrieval, others are better suited for precise, entity-based matching. Optimizing chunking strategies should take into account model-specific retrieval biases to maximize performance across different datasets.

5 Limitations

While our study highlights the impact of chunk size on retrieval performance, it has several limitations. Evaluation is based on string matching, which may not fully capture semantic relevance between queries and retrieved chunks. Additionally, while we include a range of datasets, they may not fully reflect real-world information retrieval scenarios - some contain synthetic structures or lack detailed relevance annotations. Finally, our analysis focuses on retrieval performance rather than directly assessing chunk quality, such as coherence or informativeness. Future work should incorporate semantic-aware evaluation metrics, intrinsic chunk assessments, and more realistic benchmarks to address these gaps.

6 Conclusion

In this study, we systematically evaluated the impact of fixed-size chunking strategies on retrieval performance across multiple datasets. Our results demonstrate that chunk size significantly influences retrieval effectiveness, with smaller

chunks (64-128 tokens) performing well for datasets with concise, fact-based answers, while larger chunks (512-1024 tokens) are necessary for datasets with long, dispersed answers. However, answer complexity alone does not fully explain chunking behavior. Factors such as document structure, question complexity, and answer locality also play critical roles, as reflected in the varying chunk sensitivity across datasets. Furthermore, retrieval performance varies across models, with Stella benefiting from larger chunks due to its ability to capture global context, whereas Snowflake performs better on smaller chunks, leveraging exact phrase retrieval. Despite these insights, our study highlights several limitations, including reliance on fixed token chunking, string-matching evaluation, and the absence of direct chunk quality measures. Additionally, dataset constraints limit the generalizability of our findings to real-world retrieval tasks. Addressing these challenges through intrinsic chunk coherence metrics, and more comprehensive datasets will be crucial for advancing chunk-based retrieval models. Ultimately, our findings emphasize the trade-offs between chunk size, retrieval model type, and dataset characteristics. Future work should focus on retrieval-specific embeddings, and fine-tuned evaluation metrics to further optimize retrieval performance in long-document and open-domain question-answering tasks.

References

- [1] Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and stella: distillation of sota embedding models, 2025.
- [2] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering, 2021.
- [3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022.
- [4] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.
- [5] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.
- [6] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.
- [7] Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. End-to-end training of neural retrievers for open-domain question answering. *arXiv preprint arXiv:2101.00408*, 2021.
- [8] Le Qi, Shangwen Lv, Hongyu Li, Jing Liu, Yu Zhang, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ting Liu. DuReader_{vis}: A Chinese dataset for open-domain document visual question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1338–1351, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS ’20, Red Hook, NY, USA, 2020. Curran Associates Inc.
- [10] Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip S. Yu. Dense hierarchical retrieval for open-domain question answering, 2021.
- [11] Renyi Qu, Ruixuan Tu, and Forrest Bao. Is semantic chunking worth the computational cost?, 2024.
- [12] Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. *arXiv preprint arXiv:2406.00456*, 2024.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- [14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

- [15] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781, 2020.
- [16] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022.
- [17] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.
- [18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.
- [19] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.
- [20] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.
- [21] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.
- [22] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.
- [23] Jerry Liu. LlamaIndex, 11 2022.
- [24] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.
- [25] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.
- [26] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih, editors, *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics.
- [27] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. COVID-QA: A question answering dataset for COVID-19. In Karin Verspoor, Kevin Bretonnel Cohen, Mark Dredze, Emilio Ferrara, Jonathan May, Robert Munro, Cecile Paris, and Byron Wallace, editors, *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online, July 2020. Association for Computational Linguistics.
- [28] Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. The TechQA dataset. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online, July 2020. Association for Computational Linguistics.
- [29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.
- [30] Puxuan Yu, Luke Merrick, Gaurav Nuti, and Daniel Campos. Arctic-embed 2.0: Multilingual retrieval without compromise, 2024.
- [31] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu,

Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.