Next Sentence Prediction with BERT as a Dynamic Chunking Mechanism for Retrieval-Augmented Generation Systems

Alexandre T. Bender, Gabriel A. Gomes, Ulisses B. Corrêa, Ricardo M. Araujo
Computer Science Graduation Program (PPGC),
Artificial Intelligence Innovation Hub (H2IA),
Center for Technological Advancement (CDTec),
Federal University of Pelotas (UFPel)
{atbender,gagomes,ulisses,ricardo}@inf.ufpel.edu.br

Abstract

Retrieval-Augmented Generation systems enhance the generative capabilities of large language models by grounding their responses in external knowledge bases, addressing some of their major limitations and improving their reliability for tasks requiring factual accuracy or domain-specific information. Chunking is a critical step in Retrieval-Augmented Generation pipelines, where text is divided into smaller segments to facilitate efficient retrieval and optimize the use of model context. This paper introduces a method that uses BERT's Next Sentence Prediction to adaptively merge related sentences into context-aware chunks. We evaluate the approach on the SQuAD v2 dataset, comparing it to standard chunking methods using Recall@k, Precision@k, Contextual-Precision@k, and processing time as metrics. Results indicate that the proposed method achieves competitive retrieval performance while reducing computational time by roughly 60%, demonstrating its potential to improve Retrieval-Augmented Generation systems.

Introduction

The Transformer architecture (Vaswani 2017) has become a notable development in natural language processing, achieved state-of-the-art results in numerous text-processing tasks. Before transformers, recurrent neural networks (RNNs) and their variants were the dominant architectures for sequence modeling tasks. The sequential nature of RNNs inherently prevents parallelism during training and inference since each token's processing depends on the completion of the preceding token. Transformers, by contrast, addressed these limitations through the introduction of the attention mechanism, which allows the model to weigh token relationships, regardless of their position. More importantly, this mechanism operates without the need for sequential processing, removing the bottleneck imposed by sequential architectures.

The transformer architecture's ability to model long-range dependencies laid the groundwork for the development of large language models (LLMs), which often boast billions of parameters and excel at generating coherent, contextually relevant text. These models have demonstrated emergent abilities, i.e. capabilities that were not explicitly trained for, and yet arise as a consequence of scaling model size and dataset diversity (Wei et al. 2022). These abilities include tasks like in-context learning, reasoning, and the ability to generalize knowledge across domains. These models have quickly become prominent in applications like chatbots (e.g., ChatGPT (OpenAI 2023)) and have been widely adopted for a myriad of tasks.

While LLMs have achieved remarkable capabilities, their limitations are increasingly evident in critical use cases (Cuskley, Woods, and Flaherty 2024; Wolf et al. 2024; Adewumi et al. 2024). Chief among these is the issue of hallucination, where models confidently produce incorrect or fabricated information (Huang et al. 2025). This behavior arises from the parametric nature of LLMs' knowledge, which is fixed during training and cannot be easily updated. It is an innate byproduct of their generative nature (Xu, Jain, and Kankanhalli 2024). As new knowledge emerges or context-specific nuances become relevant, existing models fail to incorporate these updates without retraining (a process that is often computationally prohibitive and impractical for real-time use cases). This limitation points out the need for strategies that allow LLMs to dynamically integrate external, up-to-date information.

A promising approach to address this issue is Retrieval-Augmented Generation (RAG) (Lee, Chang, and Toutanova 2019; Lewis et al. 2021). RAG systems combine the generative power of LLMs with retrieval-based mechanisms by integrating an external, dynamically updateable knowledge source. By detaching the knowledge from the parametric memory, RAG systems significantly enhance the adaptability and relevance of large language model outputs.

However, RAG pipelines are not without challenges. Their effectiveness hinges on the performance of the retrieval process, which includes how information is divided into retrievable chunks. The increasing size of LLMs' context windows has amplified the importance of retrieval strategies, as it is now possible to include multiple chunks of potentially relevant information in a single query. Simple chunking methods, such as splitting text by sentences or paragraphs, take advantage of natural linguistic boundaries to segment text. While straightforward, these meth-

ods can miss nuanced relationships between sentences or ideas, particularly when dealing with complex or domain-specific texts. The complexity of chunking is further exacerbated by the diversity of text structures across domains. Text from books, for example, follows a linear narrative structure, while academic papers often include tables, figures, and references that disrupt the flow of plain text. This structural variation demands chunking methods that can adapt to different formats, ensuring that critical information is neither fragmented nor lost. The development of domain-specific chunking strategies, therefore, remains an open area of research.

Semantic chunking methods attempt to address this limitation by embedding text fragments using models like BERT (Devlin et al. 2019), which can encode useful contextual information. These embeddings are then compared to assess the similarity between fragments, guiding the merging of related chunks.

Building on this foundation, this paper explores an alternative dynamic chunking mechanism for RAG systems. Specifically, we propose using BERT's next-sentence prediction (NSP) (Sun et al. 2022) capabilities as a means of forming coherent, contextually relevant chunks. Unlike traditional semantic methods that rely solely on contextual embeddings, our approach leverages the predictive capabilities of BERT for NSP to dynamically determine sentence boundaries and relationships in addition to context encoding. This enables the formation of adaptive chunks that are better suited for retrieval tasks.

Through this work, we hypothesize that the integration of BERT-based NSP for chunking can enhance RAG pipelines by improving the relevance and coherence of retrieved information, while also speeding up the process by eliminating the need to compute similarity scores. The remainder of this paper is structured as follows: the Background section reviews related work and introduces concepts such as the Transformer architecture and Next Sentence Prediction. In the Methodology, we present our proposed chunking mechanism. The Results section details our experiments and the findings that support the effectiveness of our approach. Finally, the Conclusion summarizes the main contributions and suggests directions for future research.

Background

This section covers important concepts to our approach, starting with the Transformer architecture. It then covers Retrieval-Augmented Generation systems, chunking techniques for managing large datasets, and BERT's Next Sentence Prediction task, which enhances understanding of sentence relationships.

The Transformer Architecture

The Transformer architecture (Vaswani 2017) has become a foundational model in natural language processing. It introduced the concept of self-attention, which allows each token in a sequence to consider the entire context of the other tokens simultaneously, enabling more efficient and flexible handling of long-range dependencies compared to previous models. For the purposes of this work, we will focus on describing encoder-only models and decoder-only models.

Encoder-Only Models. As the name suggests, they only use the encoder portion of the Transformer architecture. These models are designed to process and encode input sequences into fixed-length representations, capturing the contextual relationships between the tokens. One of the most well-known examples of an encoder-only model is BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2019). Ultimately, Encoder-only models focus on generating useful representations of input text, which can then be used for downstream tasks, but they do not directly generate output text. BERT is highly effective for tasks such as text classification, named entity recognition, and question answering.

Decoder-Only Models. These models rely solely on the decoder component of the Transformer, which processes the encoded input and generates token predictions one at a time. A notable example of a decoder-only model is GPT (Generative Pretrained Transformer) (Radford 2018). GPT is trained using causal language modeling, predicting the next word in a sequence given the preceding words. By learning from vast amounts of data, these models can generate text that is not only syntactically correct but also contextually relevant, being useful for tasks such as creative writing, summarization, and dialogue systems.

Retrieval-Augmented Generation

Retrieval-Augmented Generation (Lee, Chang, and Toutanova 2019; Lewis et al. 2021) is a framework that enhances generative models by integrating information retrieval techniques, enabling more informed and contextually relevant generation. The main components of RAG are the retrieval module and the generative model.

The Retrieval Module component is responsible for fetching relevant information from a large corpus of data based on a given query or context. Typically, this is done using a retrieval model, dense retrieval with embeddings, or a combination of both. The goal is to gather documents or passages that can provide additional context for the generation task.

Once the relevant information is retrieved, the generative model takes this information along with the original query to produce a response or output. The generative model can then integrate the retrieved content with its own knowledge to generate more accurate, detailed, and contextually appropriate text.

Chunking Techniques

There are various chunking techniques, each with its own advantages and challenges. The most common approaches include fixed-size chunking, sentence chunking, and semantic chunking.

Fixed-size chunking involves dividing a document into uniform segments, typically based on a predetermined number of tokens. This method is straightforward to implement and computationally efficient, making it a popular choice. Since the chunk boundaries are determined arbitrarily, they may split sentences, phrases, or other meaningful linguistic structures, disrupting the flow of information.

Sentence chunking divides text based on its inherent structure, rather than using a fixed token size, text is segmented into sentences. The primary advantage of sentence chunking is that it preserves the natural linguistic structure of the text, ensuring that each chunk contains coherent information.

More advanced methods like semantic chunking, aim to capture the underlying meaning of text fragments by leveraging techniques like embedding-based similarity scoring. These methods iteratively merge adjacent chunks that meet a predefined similarity threshold, creating semantically coherent units, such as topics, entities, or key concepts. However, this method is more complex to implement, as it requires a deeper understanding of the content and context of the text, often relying on pre-trained language models.

BERT And Next Sentence Prediction

BERT (Devlin et al. 2019) is a transformer-based encoder model that has become popular in the field of representation learning. Unlike traditional encoder models, which process text in a unidirectional manner (either left-to-right or right-to-left), BERT is designed to read text in both directions simultaneously, capturing the full context of each word in a sentence.

BERT is most commonly used as an embedding model, but it can also be used for Next Sentence Prediction (Sun et al. 2022). During its pre-training, BERT for NSP is provided with pairs of sentences and learns to predict whether the second sentence in a pair logically follows the first one. The NSP task helps BERT understand the relationship between sentences. This model operates differently than BERT for embeddings, as it processes two concatenated sentences, and outputs a score based on if the second sentence follows the first. This contrasts with embedding similarity calculation, where first the embeddings have to be generated with BERT and only then a similarity score can be calculated using a metric such as cosine distance.

Related Works

This section provides an overview of the state of current research in retrieval-augmented generation and chunking mechanisms.

Previous works have proposed improvements to the RAG pipeline by introducing query and retrieval transformations to enhance the interaction between user input and the retrieval mechanism, ultimately leading to increased response relevance and accuracy.

Query Transformation

TOC (Kim et al. 2023) breaks down input queries into multiple sub-queries, supplying the RAG LLM with the aggregated information retrieved from these sub-queries.

HyDE (Gao et al. 2022) and Query2Doc (Wang, Yang, and Wei 2023) recognize that user queries are not always ideal for retrieving relevant answers, as aligning questions and answers in latent space is challenging. Instead, they propose generating a hypothetical answer based on the user input and then searching for chunks most similar to this hypothetical document. Their experiments demonstrate a significant performance improvement compared to state-of-the-art retrievers across various tasks and languages.

Some studies take the opposite approach, proposing modifications to the source documents by generating pseudo-queries for each one, with the goal of improving document matching (Liu 2022).

Other approaches operate directly at the embedding level, using contrastive learning (Zhang et al. 2023; Xiao et al. 2024) to align document and query embeddings more closely in latent space.

Some works address the issue of user prompts being redundant and unnecessarily lengthy. These studies focus on improving the user prompt through summarization and extractive methods (Wang et al. 2023; Xu, Shi, and Choi 2023).

Retriever Optimization

Embedding methods and chunking strategies have a significant impact on retrieval performance. Retrieved chunks can vary in both quantity and relevance. A common optimization is to rerank the retrieved documents. When constraints allow for slower inference during reranking, strategies such as those using language models like T5 (Nogueira, Jiang, and Lin 2020) and BERT (Nogueira et al. 2019) deliver better results, albeit with slower inference times. Alternatively, TILDE (Zhuang and Zuccon 2021) precomputes the likelihood of query terms and reranks the documents based on these precomputed scores.

Methodology

This section describes the methodology employed in this study, detailing procedures used to carry out the research and analyze the results.

Proposed Method

An alternative approach to semantic chunking leverages BERT's Next Sentence Prediction capability to dynamically merge sentences based on their semantic continuity. In this method, sentences are first treated as independent units and then evaluated pairwise using the NSP task, which predicts the likelihood of one sentence following another. Sentences are merged into a chunk if their NSP probability surpasses a predefined threshold.

This approach contrasts with cosine similarity-based chunking, which assesses semantic closeness using embeddings. While cosine similarity captures topical relationships, NSP-based chunking focuses on the logical and sequential flow of sentences, resulting in chunks that better reflect the natural progression of ideas.

Baseline Methods

We evaluate the method against other chunking techniques, including sentence chunking and semantic chunking using BERT embeddings and cosine similarity.

Sentence chunking divides text into individual sentences or groups of consecutive sentences. By using sentence

boundaries, this method preserves the grammatical and semantic integrity of each chunk. However, it may fail to capture relationships between sentences that form a cohesive idea, especially when complex concepts span multiple sentences. While it provides an improvement over fixed-size chunking in terms of coherence, it lacks flexibility in adapting to the semantic structure of longer passages.

Semantic Chunking Using BERT Embeddings and cosine similarity creates chunks by grouping sentences based on their semantic closeness. Sentence embeddings are generated using a transformer model like BERT, and cosine similarity is calculated between consecutive sentence embeddings. Sentences with high similarity scores are merged into the same chunk, while lower scores signal a break between ideas. This method aligns chunks with topics or concepts but can struggle to differentiate between sentences that are topically similar yet belong to separate logical flows, leading to over-segmentation or overly broad chunks.

Experimental Setup

For our experiments, we utilize the SQuAD v2 dataset (Rajpurkar, Jia, and Liang 2018). Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

SQuAD is commonly used to assess question answering tasks. In this study, instead of evaluating reading comprehension, we exclude the unanswerable questions and use the dataset to evaluate information retrieval. The remaining data is then split into sample and test sets. Table 1 provides a description of this process.

Group	Quantity
Total Questions	11,873
Answerable Questions	5,928 (49.93%)
Unanswerable Questions	5,945 (50.07%)
Sample Fold	1778 (29.99%)
Test Fold	4150 (70.01%)

Table 1: Overview of SQuAD v2 Validation Split Statistics.

We split the answerable questions into two groups using cross-validation to evaluate the chunking methods. This is necessary because some methods are parameterized, and using the entire dataset for fitting or statistical analysis could lead to an overly optimistic evaluation. Traditionally, the training fold is larger than the test/validation fold. However, since these methods do not require the intensive training typical of more complex machine learning approaches, a smaller sample is sufficient to parameterize them. Therefore, we divide the dataset into 30% for the sample set and 70% for the test set. All experiments use a random seed (42) to ensure reproducibility.

In our experiments, we use the BERT-base-uncased model, a pre-trained transformer capable of processing text with a maximum context size of 512 tokens. This model is trained on lowercase English text, making it case-insensitive. We utilize it to compute either cosine similarity between sentence embeddings or Next Sentence Prediction probabilities, depending on the chunking method. For the retrieval task, we use the all-MiniLM-L6-v2 model, a lightweight sentence transformer designed for efficient semantic search.

For the semantic approaches using BERT (cosine similarity and NSP), we calculate the similarity between pairs of sentences in the documents and use the average similarity from the sample set as a threshold. This fitting step serves to parametrize the model, while ensuring that the test set data is withheld, following best practices to prevent overfitting.

Evaluation Metrics

While chunking does impact the generation step of a RAG pipeline, its most immediate effects are seen in the retrieval step. For this reason, we choose to evaluate the methods using information retrieval practices and metrics, including recall@k, precision@k, and semantic precision.

Recall@k in information retrieval measures whether the relevant documents were correctly retrieved, or how much of the total relevant information was retrieved. The variation, recall@k, specifically considers the top k retrieved chunks.

Precision@k complements recall@k, as it evaluates the relevance of the retrieved information. Recall@k alone is insufficient because it only considers whether relevant information is included. For example, a retrieval model that returns all chunks would achieve a perfect recall@k score, even if many of the chunks are irrelevant. To address this, we introduce a version of precision@k, which calculates the ratio of relevant words (ground truth answer) per chunk.

Semantic precision@k is another metric that helps assess the chunking methods by measuring the proximity of the ground truth and the retrieved chunk in latent semantic space. This is done by calculating the cosine distance between the ground truth and the retrieved chunks. We report the average cosine similarity for the top k retrieved chunks.

Results

We conducted the experiments on Google Colab using an A100 compute engine backend GPU, and the total experiments consumed up to 12 compute units of credits. Figure 1 illustrates the distribution of chunk sizes for each method. The sentence chunking method produces chunks that roughly follow a normal distribution, with an average of 26 words per chunk. It is important to note the semantic methods merge chunks that were originally split by the sentence chunking method, resulting in larger chunk sizes by definition. The BERT cosine method generates chunks of around 100 words, while the BERT NSP method creates even larger chunks with a broader distribution and less pronounced peak.

Table 2 presents the evaluation metrics for each method using $k = 3$. As expected, sentence chunking is the fastest approach by a significant margin and is highly cost-efficient. Additionally, since the semantic approaches discussed here involve chunk merging, sentence chunking results in shorter

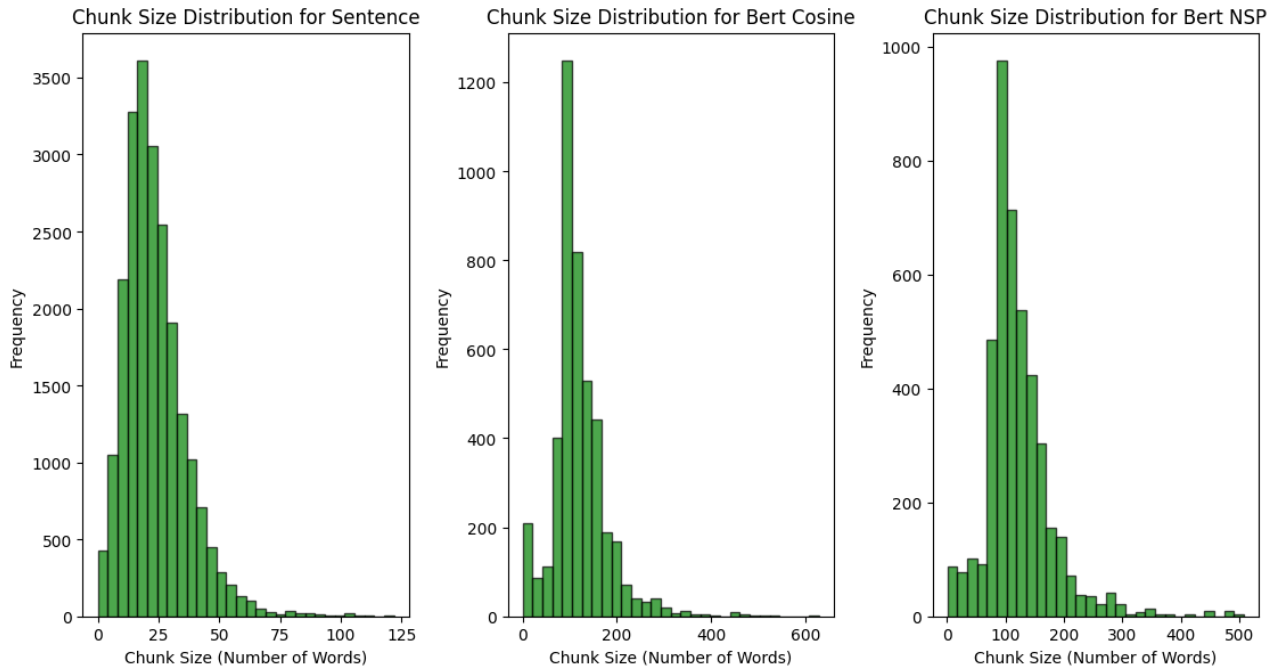

Figure 1: Chunk size frequency distributions for Sentence Chunking, Semantic Chunking using BERT Embeddings using Cosine Similarity, and Semantic Chunking using BERT for Next Sentence Prediction, respectively. Source: The authors.

Table 2: Test Fold Metric Results.

Approaches	Recall@3	Precision@3	Semantic-Precision@3	Processing Time
Sentence	0.9371	**0.0547**	0.2994	**3 min**
BERT Cosine	0.9829	0.0340	0.3198	44 min
BERT NSP (Ours)	**0.9834**	0.0344	**0.3211**	27 min

chunks, leading to a high precision@3 score. The relatively low precision values are expected, as they reflect the characteristics of the SQuAD2 dataset, which contains smaller ground truths compared to document sizes. Even with sentence-based chunking, which averages about 25 words per chunk, the precision remains low due to the inclusion of unnecessary extra information. When comparing the two semantic merging approaches, the proposed BERT NSP method outperforms BERT embeddings with cosine similarity by a small margin across all three performance metrics. While recall@3 and precision@3 show only slight improvements, semantic precision shows a moderate improvement. More importantly, the processing time for the BERT NSP approach is 61.36% faster than the commonly used BERT embeddings and cosine similarity method. This improvement in speed is notable, as it not only boosts efficiency but also makes the approach more practical for real-world applications, where processing time is often a limiting factor.

BERT NSP avoids the need to compute dot products and norms directly, a process that can be computationally expensive, particularly when dealing with a large number of embedding pairs. Instead, it leverages its pretraining for Next Sentence Prediction by processing concatenated inputs in a single forward pass. This design eliminates the overhead of generating individual embeddings for separate inputs, as required in methods like cosine similarity, which involve embedding each chunk independently before calculating their similarity. As a result, BERT NSP can achieve faster performance.

Conclusion

In this study, we introduced an alternative chunking method utilizing BERT trained for Next Sentence Prediction. The proposed approach demonstrates competitive performance, achieving higher semantic precision and comparable recall@3 and precision@3 metrics. Notably, by using NSP to merge sentences, our method is roughly 60% faster in processing time compared to the baseline semantic method, making it significantly less computationally demanding during inference.

Future research could explore the use of alternative embedding models to assess their interactions with BERT for NSP as a chunking method and their overall impact. There is also significant potential to investigate how this chunking

method influences the downstream generation step in RAG workflows. Furthermore, other RAG optimizations such as reranking may complement this chunking approach, potentially improving its performance in various ways. Testing this chunking method across different languages could provide deeper insights into chunking mechanisms and their behavior. Hybrid approaches might also benefit from incorporating NSP as an additional similarity metric. Lastly, further efforts could focus on optimizing the parametrization of BERT for NSP to enhance its effectiveness as a chunking mechanism.

By demonstrating competitive results and reduced computational demands compared to traditional semantic chunking methods, we aim to encourage further research into alternative chunking approaches that have yet to be fully explored. Further research into the retrieval step of RAG systems offers a promising path for advancing state-of-the-art developments in retrieval-augmented generation and information retrieval as a whole.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. We would like to thank the FAPERGS - Brasil for Financial Support, Award Agreement 22/2551-0000598-5.W. Lastly, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU.

References

Adewumi, T.; Habib, N.; Alkhaled, L.; and Barney, E. 2024. On the limitations of large language models (llms): False attribution.

Cuskley, C.; Woods, R.; and Flaherty, M. 2024. The limitations of large language models for understanding human language and cognition. *Open Mind* 8:1058–1083.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gao, L.; Ma, X.; Lin, J.; and Callan, J. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43(2):1–55.

Kim, G.; Kim, S.; Jeon, B.; Park, J.; and Kang, J. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. *arXiv preprint arXiv:2310.14696*.

Lee, K.; Chang, M.-W.; and Toutanova, K. 2019. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Liu, J. 2022. LlamaIndex.

Nogueira, R.; Yang, W.; Cho, K.; and Lin, J. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Nogueira, R.; Jiang, Z.; and Lin, J. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

OpenAI. 2023. Chatgpt (mar 14 version). https://chat.openai.com/chat. [Large language model].

Radford, A. 2018. Improving language understanding by generative pre-training.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for squad.

Sun, Y.; Zheng, Y.; Hao, C.; and Qiu, H. 2022. Nsp-bert: A prompt-based few-shot learner through an original pre-training task–next sentence prediction.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, Z.; Araki, J.; Jiang, Z.; Parvez, M. R.; and Neubig, G. 2023. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Wang, L.; Yang, N.; and Wei, F. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wolf, Y.; Wies, N.; Avnery, O.; Levine, Y.; and Shashua, A. 2024. Fundamental limitations of alignment in large language models.

Xiao, S.; Liu, Z.; Zhang, P.; Muennighoff, N.; Lian, D.; and Nie, J.-Y. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 641–649.

Xu, Z.; Jain, S.; and Kankanhalli, M. 2024. Hallucination is inevitable: An innate limitation of large language models.

Xu, F.; Shi, W.; and Choi, E. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.

Zhang, P.; Xiao, S.; Liu, Z.; Dou, Z.; and Nie, J.-Y. 2023. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*.

Zhuang, S., and Zuccon, G. 2021. Tilde: Term independent likelihood model for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1483–1492.