IDMS: Information Density-driven Multi-scale Segmentation

Jiasong Chen 30920241154546[1], Yazhe Chen 23020241154376[1], Zhicheng Wu 30920241154579[1], Zhongwei Xiong 23020241154461[1], Gefei Zhang 30920241154584[1]

[1]Deep Learning Information School Class

Abstract

Retrieval-Augmented Generation (RAG), as a technological paradigm to alleviate the problems caused by large language models (LLMs), often leverages the knowledge retrieved from documents to assist generation. However the most existing research overlook the appropriate segmentation granularity for different types of documents and poorly consider the discontinuity of relevant information within the document. To address this, we propose an Information Density-driven Multi-scale Segmentation (IDMS) algorithm that improves retrieval efficiency and optimizes resource utilization. We address segmentation granularity issue by leveraging LLMs to dynamically partition documents into logically coherent, independent chunks, ensuring each segment maintains a complete expression of ideas. Additionally, we calculate sentence information density, selectively reduce tokens in low information-density texts and enrich entity information in high density texts, improving efficiency while retaining key semantic information.

Introduction

Although large models have made significant breakthroughs in many areas, they still face several challenges, such as the issue of hallucinations(Xu, Jain, and Kankanhalli 2024), and the lack of domain-specific data(Li et al. 2024). The introduction of Retrieval-Augmented Generation (RAG) has addressed some of the challenges faced by large models to a certain extent(Lewis et al. 2021). The process of large model Retrieval-Augmented Generation (RAG) involves first retrieving relevant external knowledge and then generating a response based on the retrieval results. It primarily consists of two components: a retriever and a generator.

Since the accuracy of retrieval significantly impacts the performance of the model, current research on RAG primarily focuses on improving the precision of the retrieval process(Besta et al. 2024). However, in RAG research, the study of text chunking methods is often overlooked, despite the fact that chunking has a significant impact on retrieval results(Xu et al. 2023). Early chunking methods primarily involved splitting the text into fixed-length segments with a certain degree of overlap between each segment. More recent studies have proposed content-based chunking methods. For example, some approaches segment documents based on structural elements, such as paragraphs or chapters(Wang, Chang, and Sui 2020), while others use key sentences or semantic units for segmentation(Liu and Lapata 2019). However, most existing text segmentation methods in RAG face the following issues: (1) It is difficult to determine the appropriate segmentation granularity for different types of documents, making it challenging to choose whether to split by sentence, paragraph, or another level of granularity. (2) Document segmentation may result in the lack of relevant entities in the segmented chunks, resulting in invalid related information.

To address the issues mentioned above, we propose an **Information Density-driven Multi-scale Segmentation (IDMS)** algorithm. To address the first issue mentioned earlier, we compare the probability difference of a binary classification with a set threshold, which leverages the capabilities of LLMs to flexibly partition documents into logically coherent, independent chunks. And to address the second issue, we calculate the information density of sentences and group them accordingly. For instance, paragraphs with high information density should be added related entities description, while sections with a more narrative style and lower information density should be reduced the number of tokens.

The main contributions of this paper are as follows:

- **Margin sampling segmentation**: We introduce margin sampling chunking, which leverages the capabilities of LLMs to flexibly partition documents into logically coherent, independent chunks. This dynamic adjustment of granularity ensures that each segmented chunk contains a complete and independent expression of ideas, thereby avoiding breaks in the logical chain during the segmentation process.
- **Information density-based segmentation**: After text segmentation, we calculates the information density of each sentence, allowing sentences to be dealt differently based on their information density. This solves the similarity issue by reducing the tokens numbers to low-density sections (e.g., narrative descriptions) and supplement the description of related entities to high-density sections (e.g., important thematic transitions), enhancing efficiency by reducing redundant text without losing critical semantic information.

- **IDMS algorithm**: By integrating the above two steps, we develop the IDMS algorithm, which effectively balances segmentation granularity and semantic coherence. This approach enables the algorithm to adaptively segment documents based on varying information density and improve segmentation performance, particularly in complex document structures.

Related Work

Retrieval-Augmented Generation

When large language models perform poorly in the face of new or proprietary knowledge, a common practice is to use knowledge augmentation to improve their performance. As a specific method of knowledge augmentation, Retrieval-Augmented Generation (RAG, Lewis et al. 2020) combines retrieval technology with generative models, allowing the model to first retrieve relevant text or knowledge, and then generate answers based on the retrieval results. Traditional generative models such as GPT (Radford et al. 2019) rely solely on pre-trained knowledge, leading to issues such as hallucinations or incorrect information in scenarios that require external factual information. To address these limitations, RAG combines the capabilities of a retriever with that of a generator. The retriever fetches relevant documents from a large corpus, which are then used by the generator to produce more accurate and contextually relevant text.

Previous retrieval-based approaches like REALM (Guu et al. 2020) and DPR (Karpukhin et al. 2020) have laid the groundwork for using external knowledge during generation tasks. However, these models operate in two separate phases, where retrieval and generation occur sequentially. RAG innovates by merging these two components into a single differentiable framework, allowing both retrieval and generation to influence each other dynamically. This approach has demonstrated success in tasks such as open-domain question answering (Izacard and Grave 2020b) and dialogue generation (Komeili 2021), outperforming purely generative models in terms of factual correctness and diversity.

Document Chunking for RAG

Document Chunking plays a crucial role in retrieval-based models like RAG, especially when dealing with lengthy or complex documents. Traditional models often suffer from issues related to the length of input sequences, as most transformer-based architectures have a limited token capacity. Several earlier studies have explored different strategies for segmenting documents. For instance, Wang, Chang, and Sui (2020) proposed splitting documents based on discourse-level structures, such as paragraphs or sections, while others have explored content-based segmentation using key sentences or semantic units (Liu and Lapata 2019). A key focus in recent research is finding methods to dynamically segment documents, where segmentation is tailored to the query or downstream task, rather than applying a one-size-fits-all approach. This dynamic segmentation has shown promise in reducing retrieval noise and improving the overall relevance of the retrieved passages.

In the context of RAG, chunking documents into manageable segments has been shown to improve both retrieval accuracy and generation fluency. For instance, Kim, Seo, and Shin (2021) explored hierarchical approaches for splitting large documents, improving the efficiency of both retrieval and generation tasks. Additionally, Izacard and Grave (2020a) proposed a model that splits documents into multiple chunks and applies retrieval to each, generating results based on relevant sections only.

Token Reduction for RAG

Sequence length has become a significant factor limiting the scalability of transformer models (Vaswani 2017). Token reduction techniques are essential for improving the efficiency of retrieval and generation models, particularly when dealing with extensive document collections. By reducing the number of tokens or simplifying input sequences, models can better handle large datasets without sacrificing performance. Sanh (2019) introduced DistilBERT, which reduces the number of parameters while retaining high accuracy by distilling the knowledge of larger models. Additionally, methods such as pruning and token pooling (Kitaev, Kaiser, and Levskaya 2020) have been explored to reduce model complexity and accelerate inference.

In RAG-based systems, token simplification directly impacts the quality and speed of both retrieval and generation. Reducing redundant or irrelevant tokens allows the model to focus on key information, thus improving the relevance of the retrieved documents. For example, Li, Ji, and Han (2021) proposed an approach to filter out uninformative tokens during the retrieval phase, leading to faster and more accurate document retrieval.

Proposed Solution

Our main contribution is an innovative text segmentation technique named IDMS, which can use the power of large language models to divide documents into independent and logically coherent chunks. Our approach allows variable chunk size, thereby more effectively maintaining the integrity of chunk content. When it is difficult to determine the segmentation granularity for different types of documents, this technique can be flexibly adjusted to avoid interruption of the logical chain during segmentation, thereby improving the quality of document segmentation.

As illustrated in Figure 1, our method absorbs the advantages of traditional segmentation strategies without destroying the coherence of sentences and the integrity of sentence structure. Specifically, we hope to find a semantically complete chunk that contains several sentences. These sentences are not only semantically related, but also contain deep language logical connections, including but not limited to causal relationships, transitional relationships, parallel relationships, and progressive relationships. At the same time, the entities and their relationships in this chunk must be clearly expressed. In order to achieve this goal, we have designed and implemented the following two strategies.

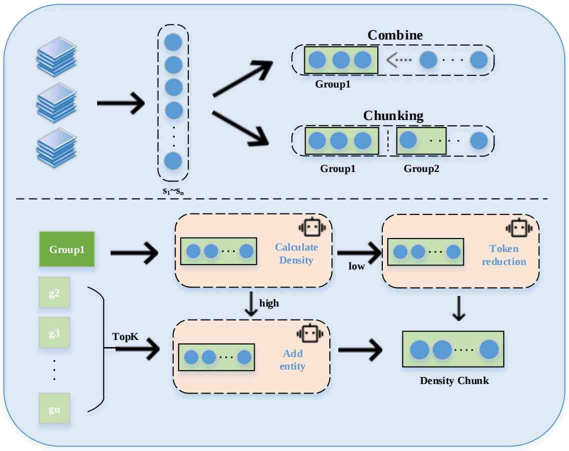

Figure 1: An illustration of IDMS model structure.

Margin Sampling Chunking

Given a text, the initial step involves segmenting it into a collection of sentences denoted as $(x_1, x_2, \ldots, x_n)$, with the ultimate goal being to further partition these sentences into several chunks, forming a new set $(X_1, X_2, \ldots, X_k)$, where each chunk comprises a coherent grouping of the original sentences. The method can be formulated as:

$$\begin{aligned}\text{Margin}_M(x_i) = P_M\left(y = k_1 \mid \text{Prompt}(x_i, X^{'})\right) - \\ P_M\left(y = k_2 \mid \text{Prompt}(x_i, X^{'})\right)\end{aligned} \tag{1}$$

where (k_1, k_2) indicates a binary decision between yes or no for a segmentation judgment. $\text{Prompt}(x_i, X^{'})$ represents forming an instruction between $x_i \in \{x_l\}_{l=1}^{n}$ and $X^{'}$, regarding whether they should be merged, where $X^{'}$ encompasses either a single sentence or multiple sentences. Through the probability P_M obtained by model M, we can derive the probability difference $\text{Margin}_M(x_i)$ between the two options. Subsequently, by contrasting $\text{Margin}_M(x_i)$ with the threshold θ, a conclusion can be drawn regarding whether the two sentences should be segmented. For the setting of θ, we initially assign it a value of 0 and then adjust it by recording historical $\text{Margin}_M(x_i)$ and calculating their average.

After completing the above steps, the document has been preliminarily divided into chunks. However, in order to maintain the coherence of the sentence structure, the sentences in the same chunk must be continuous. For some types of documents, some related information is not continuous, and there may be some irrelevant sentences in the middle. Therefore, our framework also adopts a segmentation method based on information density. This method allows LLM to judge the information density in the current chunk from multiple angles and adopt different strategies according to the different information density.

Density-driven Chunking

After document segmentation, the relevant information may be invalid due to the lack of relevant entities in the segmented chunks. Therefore, we borrowed the idea of Chain of Dense (COD)(Adams et al. 2023) and proposed an information density prompt module. Our goal is to hope that the chunks we segmented should be detailed, entity-centric, and not entity-dense and difficult to understand.

Specifically, calculate the information density of the generated chunks. If the density is low but entities are relatively complete, use the LLM to summarize and reduce token count. If the density is high, there may be missing entities, then iteratively add them and supplement with non-adjacent, semantically similar sentences. For high-density chunks, encode sentences, calculate cosine similarity, and select the top-k as supplementary sources of relevant information. Use the LLM to check for missing entities, expanding the chunk while keeping its size stable. After several iterations, achieve a balance with moderate density, ensuring the chunk is informative yet concise and easy to understand.

Below is the detailed formula definition of information density. We use the following steps to refine the calculation method of each item:

$$\rho(B) = \frac{\sum_{i=1}^{m} I(B_i) \cdot W(B_i) \cdot C(B_i) \cdot S(B_i) \cdot \alpha(B_i)}{L(B)} \tag{2}$$

Effectiveness of entities $I(B_i)$. Effectiveness measures how informative and descriptive the entity description in a chunk is. It is quantified by the amount of information in the context, $I(B_i)$, which can be evaluated using similarity methods based on language models (e.g., BERT, GPT) or external knowledge bases (e.g., Wikipedia, WordNet).

$$\begin{aligned}I(B_i) = \text{cosine\_similarity}(E(B_i), \\ \text{Contextual\_Embedding})\end{aligned} \tag{3}$$

where $E(B_i)$ is the word vector of entity B_i, and 'Contextual_Embedding' represents its embedding in the context. A high $I(B_i)$ indicates a complete, detailed description with strong information, while a low $I(B_i)$ suggests the description is brief or lacks relevant context.

Number of entities $W(B_i)$. The number of entities measures the diversity of different entities in a chunk, which can usually be calculated by the number of entities that actually appear in the chunk:

$$W(B_i) = \text{Number of Unique Entities in Chunk B} \tag{4}$$

where $W(B_i)$ is the count of the i-th entity in chunk B. For each chunk, count and weight all independent entities. Generally, a higher entity count indicates greater information density, as it includes more important concepts and details.

Correlation within chunk $C(B_i)$. The correlation within a chunk reflects the semantic connection between its sentences or entities. It is measured by calculating the cosine similarity between all pairs of sentences or entities, resulting in an overall correlation score. For n sentences or entities $\{S_1, S_2, ..., S_n\}$, the overall correlation is the sum of pairwise similarities:

$$C(B) = \frac{1}{n(n-1)} \sum_{i \neq j} \text{cosine\_similarity}(S_i, S_j) \quad (5)$$

where S_i and S_j are the i-th and j-th sentences or entities in chunk B. A high correlation between sentences or entities results in $C(B)$ close to 1, indicating strong internal consistency, while low correlation leads to $C(B)$ near 0, indicating weak information coherence.

Rating of large language model $S(B_i)$. The large model rating $S(B_i)$ assesses the quality of chunk B_i using a model (e.g., GPT-4, BERT), considering factors like semantic consistency, redundancy, and completeness. The score is calculated by evaluating the chunk with the pre-trained model M, yielding a value $S(B_i) \in [0, 1]$, where 1 indicates high quality and 0 indicates poor quality.

$$S(B_i) = \text{ModelScore}(B_i) \quad (6)$$

this metric $S(B_i)$ will comprehensively consider multiple aspects such as chunk coherence, contextual consistency, and semantic completeness.

Weighted similarity between chunks $\alpha(B_i)$. The similarity weighting factor measures the similarity between the current chunk and others using cosine similarity. The weighting coefficient $\alpha(B)$ is calculated by measuring the cosine similarity between chunk B and other chunks B'.

$$\alpha(B) = \text{cosine\_similarity}(B, B') \quad (7)$$

where B and B' are different chunks represented by their embedding vectors. A higher $\alpha(B)$ indicates greater similarity, allowing complementary information to enhance density. A low $\alpha(B)$ suggests weak relevance between the chunks.

Chunk length $L(B)$. The length of a chunk $L(B)$ directly represents the number of tokens in the chunk, which is usually measured by the number of tokens in the vocabulary. Longer chunks may contain more information, but they may also lead to more verbose information, so we include it in the calculation of information density to balance it.

$$L(B) = \text{Token Count}(B) \quad (8)$$

Combining all the above dimensions, the final information density formula can be written as formula 2 that evaluates information density comprehensively from multiple dimensions, such as entity effectiveness, number of entities, the correlation within chunk, the rating of large language model, and the weighted similarity between chunks, by refining the calculation method of each item. The calculation method of each factor ensures that the information density can reflect the semantic depth, information completeness, and similarity between chunks of the text block, while avoiding the problem of redundant or overly dispersed information. This method can provide a comprehensive and quantitative evaluation standard for text segmentation in the paper.

Experiments

This section mainly introduces the experimental settings and reports the results of IDMS. We use various datasets and LLMs to compare IDMS with state-of-the-arts, and conduct ablation study on the results to verify the effects of different designs.

Experimental Settings

We evaluate various methods using popular question answering (QA) datasets in the RAG field, focusing on Chinese and English performance with metrics on correctness, authenticity, and recall. The datasets include CRUD, RAGBench, and LongBench. The CRUD dataset, containing single-hop, two-hop, and three-hop questions, is evaluated with BLEU, ROUGE-L, and BERTScore (Lyu et al. 2024). From RAGBench, we use the CUAD dataset, applying the same metrics (Friel, Belyi, and Sanyal 2024). LongBench includes eight Chinese and English datasets covering single and multi-hop QA, evaluated with F1 and ROUGE-L (Bai et al. 2023).

RAG chunking methods include rule-based and dynamic chunking, the latter using semantic similarity models or large language models (LLMs). Rule-based chunking splits text into fixed-length chunks, ignoring sentence boundaries. Llama indexing (LangchainAI 2022) balances sentence boundaries and token count. Semantic similarity chunking (Xiao et al. 2023) groups related sentences using sentence embeddings. LumberChunker (Duarte et al. 2024) predicts optimal segmentation points with LLMs, with each method offering context-specific advantages.

Our method heavily relies on LLMs, and we test various models including Qwen2.5-1.5B, Internlm2-1.8B, Baichuan2-7B, and Qwen2.5-7B (Yang et al. 2024; Cai et al. 2024; Yang et al. 2023), along with smaller models such as Pythia-0.16B, Pythia-0.41B, and Qwen2-0.5B (Biderman et al. 2023; Yang et al. 2024). For longer texts, we use a KV cache method to maintain logical consistency while preventing GPU memory overflow. Text segmentation and metric evaluation are performed on NVIDIA A800, with consistent block lengths across methods.

Main Results

Comparison against Baselines. We conducted a systematic evaluation of the performance of two baseline methods, as presented in Table 1. Overall, our IDMS method demonstrates strong performance across various datasets. Our IDMS method demonstrates strong performance across all datasets when utilizing the Qwen2.5-7B model. Moreover, on the Qasper and MultiHop-RAG datasets, our model achieves performance comparable to Lumber-Chunker, further demonstrating the effectiveness and feasibility of our approach. Specifically, our method achieves an F1 score of 13.11 on the 2WikiMultihopQA dataset, significantly outperforming other approaches. Overall, the Qwen2.5-7B model achieves superior performance on the 2WikiMultihopQA and Qasper datasets. On the MultiHop-RAG dataset, which prioritizes recall as the evaluation metric, smaller models such as Qwen2-0.5B and Qwen2.5-1.5B also deliver competitive results. Notably, within our method, the

Dataset	2WikiMultihopQA		Qasper		MultiHop-RAG	
Chunking Method	F1	Time	F1	Time	Hits@10	Hits@4
			Rule-based or similarity-based chunking			
Original	11.89	0.21	9.45	0.13	0.6027	0.4523
Llama_index	11.74	8.12	10.15	5.81	0.7366	0.5437
Similarity Chunking	12.00	416.45	9.93	307.05	0.7232	0.5362
			Lumber-Chunker			
Qwen2.5-1.5B	11.18	1908.25	10.09	1401.30	**0.7805**	0.6089
Qwen2.5-7B	12.94	8781.82	**11.73**	5755.79	0.7175	0.5415
Baichuan2-7B	11.23	9926.29	9.76	6498.46	0.7059	0.5596
			IDMS			
Qwen2-0.5B	11.65	253.32	9.56	143.46	0.6762	0.5342
Qwen2.5-1.5B	12.21	378.53	10.21	204.69	0.7393	0.6154
Qwen2.5-7B	**13.11**	698.93	11.67	522.44	0.7187	**0.6797**
Baichuan2-7B	12.79	753.78	10.04	569.72	0.6923	0.5896

Table 1: Performance of different chunking methods on various datasets.

Qwen2.5-1.5B model achieves a recall score of 0.7393, surpassing even the larger Qwen2.5-7B model.

Efficiency and Accuracy Trade-off. As shown in the experimental results in Table 1, our IDMS model significantly outperforms the Lumber-Chunker model in terms of runtime efficiency across all datasets, while achieving comparable or superior performance. Specifically, on the Qasper dataset, our runtime is nearly one-tenth that of the Lumber-Chunker method. This highlights the ability of our approach to effectively balance speed and overall performance. Therefore, our method enables more efficient utilization of larger parameter models, such as Qwen2.5-7B, while maintaining reduced runtime. Compared to traditional rule-based or similarity-based models, our method requires more runtime but achieves a significantly improved performance. This further demonstrates the capability of our model to effectively balance speed and accuracy.

Ablation Study

In order to study the contribution of different components and optional modules in IDMS to performance, we conducted a series of experiments for different design schemes. All sub-experiments in the ablation study used the same experimental setting, based on the qasper dataset of the LongBench benchmark, using Qwen2.5-1.5B.

Initial chunking. In the initial chunking, IDMS uses LLM to cluster sentences based on the coarse chunking based on punctuation to obtain the primary chunks. In order to study the contribution of primary chunking to the overall work, our experiment simply ablates this step and the results are shown in Table 2.

Method	F1	Time
Full IDMS	**10.21**	204.69
No Initial Chunking	9.52	**171.02**

Table 2: Ablation Study Results for Initial Chunking

Chunk enhancement based on information density. In the second stage of IDMS, i.e., chunk enhancement based on information density, we perform different operations on chunks containing different information entity densities, including supplementing entities and reducing tokens. In our ablation experiments, we perform ablation of chunk enhancement on initial blocks, and the results are shown in Table 3.

Supplement Entities	Reduce Tokens	F1	Time
✓	✓	**10.21**	204.69
✓	×	9.72	121.82
×	✓	10.14	167.24
×	×	9.45	**85.98**

Table 3: Ablation Study Results for Chunk Enhancement

Conclusion

In this paper, we proposed an innovative text segmentation method—IDMS, which leverages the capabilities of large language models to divide documents into independent and logically consistent chunks. By introducing boundary sampling-based segmentation and information density-driven optimization strategies, we can flexibly adjust the segmentation granularity while maintaining the semantic coherence of the text. Experimental results show that IDMS outperforms existing methods, especially the Lumber-Chunker, in terms of recall and accuracy in multi-hop question answering tasks, while also demonstrating improved efficiency. Overall, IDMS provides an efficient and flexible solution for text segmentation, applicable to various downstream tasks, with broad potential for practical applications.

References

Adams, G.; Fabbri, A. R.; Ladhak, F.; Lehman, E.; and Elhadad, N. 2023. From sparse to dense: GPT-4 summarization with chain of density prompting. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2023, 68.

Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Besta, M.; Kubicek, A.; Niggli, R.; Gerstenberger, R.; Weitzendorf, L.; Chi, M.; Iff, P.; Gajda, J.; Nyczyk, P.; Müller, J.; Niewiadomski, H.; Chrapek, M.; Podstawski, M.; and Hoefler, T. 2024. Multi-Head RAG: Solving Multi-Aspect Problems with LLMs. arXiv:2406.05085.

Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2397–2430. PMLR.

Cai, Z.; Cao, M.; Chen, H.; Chen, K.; Chen, K.; Chen, X.; Chen, X.; Chen, Z.; Chen, Z.; Chu, P.; et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

Duarte, A. V.; Marques, J.; Graça, M.; Freire, M.; Li, L.; and Oliveira, A. L. 2024. Lumberchunker: Long-form narrative document segmentation. *arXiv preprint arXiv:2406.17526*.

Friel, R.; Belyi, M.; and Sanyal, A. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, 3929–3938. PMLR.

Izacard, G.; and Grave, E. 2020a. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Izacard, G.; and Grave, E. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Kim, J.; Seo, Y.; and Shin, J. 2021. Landmark-guided subgoal generation in hierarchical reinforcement learning. *Advances in neural information processing systems*, 34: 28336–28349.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Komeili, M. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

LangchainAI. 2022. LangChain. https://github.com/langchain-ai/langchain.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9459–9474. Curran Associates, Inc.

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. arXiv:2005.11401.

Li, S.; Ji, H.; and Han, J. 2021. Document-level event argument extraction by conditional generation. *arXiv preprint arXiv:2104.05919*.

Li, Z.; Hu, X.; Liu, A.; Zheng, K.; Huang, S.; and Xiong, H. 2024. Refiner: Restructure Retrieval Content Efficiently to Advance Question-Answering Capabilities. arXiv:2406.11357.

Liu, Y.; and Lapata, M. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Lyu, Y.; Li, Z.; Niu, S.; Xiong, F.; Tang, B.; Wang, W.; Wu, H.; Liu, H.; Xu, T.; and Chen, E. 2024. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Transactions on Information Systems*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Sanh, V. 2019. DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.

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

Wang, K.; Chang, B.; and Sui, Z. 2020. A spectral method for unsupervised multi-document summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 435–445.

Xiao, S.; Liu, Z.; Zhang, P.; and Muennighof, N. 2023. C-pack: packaged resources to advance general Chinese embedding. 2023. *arXiv preprint arXiv:2309.07597*.

Xu, S.; Pang, L.; Shen, H.; and Cheng, X. 2023. BERM: Training the Balanced and Extractable Representation for Matching to Improve Generalization Ability of Dense Retrieval. arXiv:2305.11052.

Xu, Z.; Jain, S.; and Kankanhalli, M. 2024. Hallucination is Inevitable: An Innate Limitation of Large Language Models. arXiv:2401.11817.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.