BOOLQUESTIONS: Does Dense Retrieval Understand Boolean Logic in Language?

Zongmeng Zhang[1], Jinhua Zhu[1], Wengang Zhou[1,3], Xiang Qi[2], Peng Zhang[2], Houqiang Li[1,3]
[1]University of Science and Technology of China [2]Ant Group
[3]Institute of Artificial Intelligence, Hefei Comprehensive Nation Science Center
{zhangzm, teslazhu}@mail.ustc.edu.cn,
{zhwg, lihq}@ustc.edu.cn, {qixiang.qx, minghua.zp}@antgroup.com

Abstract

Dense retrieval, which aims to encode the semantic information of arbitrary text into dense vector representations or embeddings, has emerged as an effective and efficient paradigm for text retrieval, consequently becoming an essential component in various natural language processing systems. These systems typically focus on optimizing the embedding space by attending to the *relevance* of text pairs, while overlooking the *Boolean logic* inherent in language, which may not be captured by current training objectives. In this work, we first investigate whether current retrieval systems can comprehend the Boolean logic implied in language. To answer this question, we formulate the task of Boolean Dense Retrieval and collect a benchmark dataset, BOOLQUESTIONS, which covers complex queries containing basic Boolean logic and corresponding annotated passages. Through extensive experimental results on the proposed task and benchmark dataset, we draw the conclusion that current dense retrieval systems do not fully understand Boolean logic in language, and there is a long way to go to improve our dense retrieval systems. Furthermore, to promote further research on enhancing the understanding of Boolean logic for language models, we explore Boolean operation on decomposed query and propose a contrastive continual training method that serves as a strong baseline for the research community.[1]

1 Introduction

Text retrieval is a fundamental component of various natural language processing systems, including question answering (Chen et al., 2017a; Karpukhin et al., 2020), dialogue systems (Chen et al., 2017b), web search (Mitra et al., 2017) and so on. In the era of large language models, text retrieval has become increasingly critical as it provides an offline and incrementally updatable knowledge database, directly influencing the reliability and quality of generated responses (Karpukhin et al., 2020; Shuster et al., 2021) in the Retrieval-Augmented Generation paradigm (Lewis et al., 2020).

AND Question: How can I start a ***career in the accounting field*** **and** pursue an ***online degree program***?

OR Question: What are the impacts of ***global warming*** **or** ***climate change*** on nature and humans?

NOT Question: What causes ***upper abdomen pain*** **but is unrelated to** ***liver issues***?

Question in MS MARCO: What flower is symbol of endurance?

Question in Natural Questions: Who sings Does He Love me with Reba?

Figure 1: Examples of the AND, OR and NOT question in BOOLQUESTIONSand questions from MS MARCO and Natural Questions. Questions in BOOLQUESTIONS are more complex to understand than those in MS MARCO and Natural Questions.

Traditional text retrieval methods estimate the relevance of query and document based on lexical overlap (Salton et al., 1975; Salton and Buckley, 1988). Utilizing the "bag-of-words" assumption and set theory (Waller and Kraft, 1979; Bookstein, 1980), these methods organize text content in the form of inverted indexes (Zobel et al., 1998; Zobel and Moffat, 2006), which handle Boolean logic efficiently. The Boolean retrieval model was later extended to return ranked lists of documents by leveraging term weights (Salton et al., 1983). To more flexibly consider the weights of different words, probabilistic models such as BM25 (Robertson and Zaragoza, 2009) and statistical language modeling (Zhai, 2007) were introduced. Nevertheless, these probabilistic models still rely on lexical overlap while omitting Boolean operations, making it hard to handle Boolean logic in queries.

With the advent of deep learning, these handcrafted sparse text features have gradually been

[1]Code and dataset are available at https://github.com/zmzhang2000/boolean-dense-retrieval.

Findings of the Association for Computational Linguistics: EACL 2024, pages 2767–2779
November 12-16, 2024

replaced by low-dimensional dense vectors learned by neural networks (Reimers and Gurevych, 2019; Karpukhin et al., 2020; Qu et al., 2021), particularly using LSTM (Hochreiter and Schmidhuber, 1997) and the powerful Transformer architecture (Vaswani et al., 2017). Unlike sparse vector spaces, dense vectors are believed to capture the semantics implied in texts (Zhao et al., 2024). In these frameworks, retrieval systems are expected to assign high scores to ground-truth query-document pairs and relatively lower scores to irrelevant or randomly combined pairs, often framed within modern deep learning paradigms such as contrastive learning frameworks. Benefiting from pre-trained language models, text retrieval has achieved significant performance improvements (Guo et al., 2022; Fan et al., 2022; Yates et al., 2021).

However, dense retrieval systems primarily focus on encoding the *relevance* of texts, which may be insufficient for handling complex Boolean logic in natural language queries. Specifically, since current retrieval systems do not incorporate Boolean logic in their training paradigms, there is no guarantee of comparability in the output scores for query-document pairs involving Boolean logic. For example, the relevance scores for queries containing logical NOT, which exclude undesired information, have not been thoroughly investigated. In contrast, lexical-based retrieval systems effectively address complex Boolean logic using set theory, a method not directly applicable to dense retrieval. Consequently, it remains unclear whether dense retrieval systems can fully comprehend and process complex Boolean logic.

To answer this question, we formulate the task of Boolean Dense Retrieval (BDR) and collect a benchmark dataset, BOOLQUESTIONS, which includes complex queries containing basic Boolean logic and corresponding annotated passages. We evaluate several state-of-the-art dense retrieval systems on the proposed task and benchmark dataset, finding significant performance drops on NOT questions, as illustrated in Figure 1. Our findings indicate that current dense retrieval models do not fully understand Boolean logic in language.

Based on these observations, we generate additional training data for NOT questions and propose a contrastive continual learning baseline to enhance the understanding of logical NOT in natural language. While the proposed baseline reduces the negative rate of the passage list returned by retrieval systems, it also slightly sacrifices accuracy. This phenomenon underscores the difficulty of understanding logical NOT in natural language for dense retrieval systems. Nevertheless, the proposed baseline serves as a starting point for further research on dense retrieval in the community.

2 Related Work

In this section, we briefly introduce several highly related works about our work.

2.1 Sparse Retrieval

Sparse retrieval refers to the lexical-based retrieval model that conceives both the query and documents as a set of terms, known as the bag-of-words assumption. Boolean model is the most classical model developed in the early stage of information retrieval (Büttcher et al., 2016). Queries in Boolean retrieval model are formed as Boolean expressions, which comprise terms joint by Boolean operators including "AND", "OR" and "NOT". The retrieval process is typically based on set theory and Boolean algebra. Inverted index (Zobel et al., 1998; Zobel and Moffat, 2006) is utilized as the data structure to implement the Boolean model.

Due to the special form of queries, logic in queries is precisely delivered to the retrieval system. Despite that, the binary decision of Boolean model lacks the ability of providing the ranking of the documents. TF-IDF and BM25 (Robertson and Zaragoza, 2009) term weighting is then proposed to assign continuous relevance scores to documents, turning retrieval to a ranking task. However, the schema of retrieving documents barely on the ranking scores of documents discards the ability to express the Boolean logic explicitly, and thus struggles to tackle the complex Boolean logic in queries.

2.2 Dense Retrieval

With the re-surge of deep learning, the hand-crafted scoring function has been gradually replaced by learnable neural networks. Specifically, texts are encoded to low-dimensional dense vectors and the relevance of texts is measured in latent semantic space. LSTM (Hochreiter and Schmidhuber, 1997) and the powerful Transformer architecture (Vaswani et al., 2017) are typically used as encoders and trained with contrastive learning frameworks that pull together the representations of relevant texts and push apart those of irrelevant texts (Reimers and Gurevych, 2019; Karpukhin

et al., 2020; Qu et al., 2021). The pre-training then fine-tuning paradigm (Devlin et al., 2019; Liu et al., 2023) has significantly pushed forward the development of dense retrieval (Guo et al., 2022; Fan et al., 2022; Yates et al., 2021).

Despite these dedicated efforts, there is no evidence suggesting that dense retrievers truly understand logic in language. In this work, we propose the task of Boolean dense retrieval and collect a benchmark dataset to investigate whether dense retrieval models understand the Boolean logic in natural language.

2.3 Datasets for Modern Retrieval

A number of datasets are publicly released to provide large-scale relevance judgments or test beds for retrieval, significantly facilitating the research of modern text retrieval systems. Natural Questions (Kwiatkowski et al., 2019) includes questions from Google search engines, along with related paragraphs and answer spans from top-ranked Wikipedia documents. MS MARCO (Bajaj et al., 2018) consists of a large number of queries from Bing search logs and annotated relevant passages collected from Web pages. Variants of MS MARCO like mMARCO (Bonifacio et al., 2021) and MS MARCO Chameleons (Arabzadeh et al., 2021) are created to enrich the evaluation characteristics with respect to multilingual and question difficulty. Domain-specific retrieval datasets such as those for financial (Maia et al., 2018), scientific (Wadden et al., 2020) and biomedical (Tsatsaronis et al., 2015) fields are also released to advance the development of retrieval in other areas. Additionally, BEIR (Thakur et al., 2021) and KILT (Petroni et al., 2021) aggregate representative datasets to measure the overall performance of retrieval models.

To the best of our knowledge, none of these efforts probes into the logic implied in queries. BoolQ (Clark et al., 2019) contains questions that can be answered with barely "yes" or "no", but does not involve complex Boolean logic in its queries. Although Malaviya et al. (2023) and Zhong et al. (2023) construct questions with Boolean logic, their atomic questions are entity queries rather than natural language questions. This work is the first to construct a benchmark dataset for Boolean dense retrieval.

3 Task Definition

In this section, we initially review the technique of dense retrieval, subsequently give a formal definition of what we term as Boolean dense retrieval.

The objective of text retrieval is to identify documents within a substantial text corpus that are relevant to specific queries. We denote the query text by q and the corpus by $\mathcal{D} = \{d_i\}_{i=1}^{|\mathcal{D}|}$. Typically, retrieval models evaluate each document in the corpus, assigning relevance scores and returning the top k documents based on these scores. The scoring function is typically implemented through either lexical or semantic matching methodologies.

In this work, we focus on dense retrieval models, which are reputed for their capacity to encode the semantics of language into a dense vector representation. In dense retrieval, both query and document texts are represented as dense vectors, and their relevance is assessed within this vector space, expressed as:

$$\text{Rel}(q, d) = f_{\text{sim}}(\phi(q), \psi(d)), \tag{1}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ are encoding functions that transform text into dense vectors, and f_{sim} is a similarity function designed to measure the distance of these vectors as an indication of relevance between the encoded texts.

Building upon the conventional dense retrieval model, we introduce the concept of Boolean dense retrieval. This approach enables the integration of complex Boolean logic within text queries. Formally, a complex Boolean query is semantically equivalent to a Boolean logic expression consisting of several simpler queries, represented as:

$$q \iff q_1 \star q_2 \star \cdots \star q_m, \tag{2}$$

where q_i represents individual simple queries and $\star$ denote the Boolean operations chosen from $\{\text{AND}, \text{OR}, \text{NOT}\}$.

Apart from the construction of the query, the retrieval process remains consistent as outlined above. Moreover, because our query incorporates a more intricate construction method, we emphasize the need for more sophisticated evaluation metrics and training techniques for retrieval models.

4 Data Collection

In our work, we developed a question generation framework that progresses from simple to complex structures to gather questions embedded with

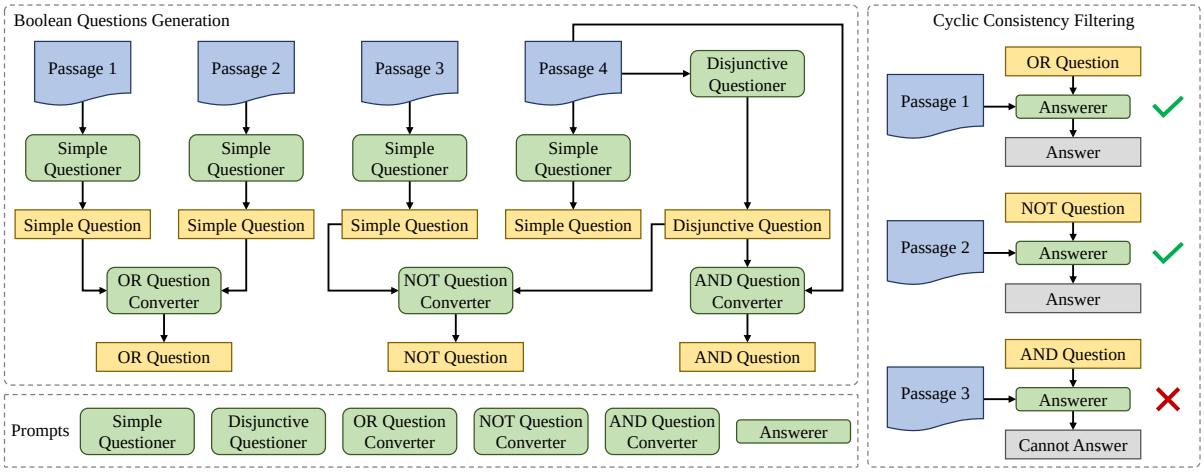

Figure 2: Data collection pipeline of BOOLQUESTIONS.

Boolean logic. Specifically, the process involves sampling text passages from a corpus, generating basic questions from these passages, and then combining these questions using Boolean operators. The combined questions are subsequently rephrased into natural language. To ensure the relevance between generated questions and corresponding passages, we introduce a cyclic consistency filtering strategy, retaining only those questions that meet this criterion.

Addressing the challenge posed by restricted access to query logs in widely used search engines, we leverage GPT-4[2], one of the most advanced language generation models currently available. This model is tasked with constructing Boolean questions tailored to the document collections found in prevalent retrieval datasets. Next, we will provide a detailed introduction to our data collection pipeline, which is shown in Figure 2. All prompts used are detailed in Appendix A.

4.1 Passage Clustering

A complex question containing Boolean logic consists of simple questions with interrelated topics. The passage sampling process must be meticulously designed, as combining questions about disparate objects could result in incoherent and peculiar complex questions. To address this, we propose a hierarchical passage sampling strategy for the subsequent question generation process. Specifically, we conduct clustering on passages of the corpus, assuming that passages from the same cluster discuss similar topics.

BOOLQUESTIONS is constructed based on the widely-used passage retrieval datasets MS MARCO (Bajaj et al., 2018) and Natural Questions (Kwiatkowski et al., 2019). For the passage collection in MS MARCO, we first encode the passages using a BERT-style encoder pre-trained for the dense retrieval task[3]. We then reduce the dimensionality of these vector representations via truncated SVD, with random sampling performed beforehand to lower computation costs. Finally, we apply agglomerative clustering from Sklearn[4] library to the compressed vectors. We also conduct clustering with TF-IDF vectorization but find that clusters formed in this manner focus more on word overlap, which is not suitable for selecting passages within the same topics compared with BERT-style encoder vectorization. The corpus of Natural Questions comprises split English Wikipedia pages, allowing similar topic passages to be directly drawn from successive passages within the same Wikipedia pages.

4.2 Atomic Question Generation

For each cluster, we randomly select n passages for atomic question generation. Since a larger number of candidate passages incurs higher computational costs, we randomly set n to 2 or 3. These selected passages are referred to as candidate passages $\{p_i\}_{i=1}^{n}$.

Simple Question We prompt GPT-4 to generate simple questions $\{q_i^{\text{simple}}\}_{i=1}^{n}$ for each candidate passage sampled from the same cluster. Naturally, candidate passages provide answers to their corre-

[2]We use the gpt-4-0125-preview version.

[3]https://huggingface.co/sentence-transformers/msmarco-distilbert-dot-v5.

[4]https://github.com/scikit-learn/scikit-learn.

sponding generated simple questions. Therefore, each simple question is paired with 1 positive passage, from which it is generated.

Disjunctive Question GPT-4 is also required to generate a disjunctive question q^{disj} that can be answered with any of the candidate passages individually. The disjunctive question is regarded as a more abstract question than a simple question. Any of the candidate passages provide answers to the disjunctive question and are annotated as positive passages for the generated question.

4.3 Boolean Question Generation

Simple questions and disjunctive questions operate at different levels of abstraction. Generally, simple questions focus on more detailed objects than disjunctive questions within the same cluster. We leverage this property to generate complex questions containing Boolean logic.

AND Questions Disjunctive questions encompass the topics discussed in all of their candidate passages, whereas simple questions within the same cluster address more concrete details of the topic. To simulate the AND operator, we prompt GPT-4 to add extra constraints to the disjunctive question, thus forming an AND question. This can be represented by the Boolean expression:

$$q^{\text{AND}} \iff q^{\text{disj}} \text{ [AND] constraints.} \quad (3)$$

We ensure that the generated question can be answered solely with a randomly selected candidate passage $\{p^{\text{AND}+}\}$. All remaining candidate passages in the same cluster are labeled as negative passages $\{p_i^{\text{AND}-}\}_{i=1}^{n-1} = \{p_i\}_{i=1}^{n} \setminus \{p^{\text{AND}+}\}$.

OR Questions Logical disjunction tends to describe the union of two topics at the same level of abstraction. We randomly selected two simple questions q_a^{OR} and q_b^{OR} from $\{q_i^{\text{simple}}\}_{i=1}^{n}$, join them with OR operator and ask GPT-4 to paraphrase the Boolean expression to a natural language question. Under these circumstances, the OR question is constructed by

$$q^{\text{OR}} \iff q_a^{\text{OR}} \text{ [OR] } q_b^{\text{OR}}. \quad (4)$$

Candidate passages p_a^{OR} and p_b^{OR} corresponding to q_a^{OR} and q_b^{OR} are annotated as positive passages $\{p_i^{\text{OR}+}\}_{i=1}^{2} = \{p_a^{\text{OR}}, p_b^{\text{OR}}\}$, while other candidate passages in the same cluster are labeled as negative passages $\{p_i^{\text{OR}-}\}_{i=1}^{n-2} = \{p_i\}_{i=1}^{n} \setminus \{p_i^{\text{OR}+}\}_{i=1}^{2}$.

NOT Questions In contrast to the construction of AND questions, NOT questions are created by adding an exclusion to the disjunctive question. Here, we adopt the information in simple questions as an exclusion to the disjunctive question. Concretely, we concatenate the disjunctive question and a randomly sampled simple question q^{NOT} with NOT operator to formulate the Boolean expression of NOT question. Then the expression is fed into GPT-4 to be paraphrased into a NOT question in the form of natural language. The creation of NOT questions is defined as

$$q^{\text{NOT}} \iff q^{\text{disj}} \text{ [NOT] } q_a^{\text{NOT}}. \quad (5)$$

Candidate passage p^{NOT} corresponding to q^{NOT} is annotated as negative passages $\{p^{\text{NOT}-}\}$ as it contradicts the negation in q^{NOT}, while other candidate passages in the same cluster are labeled as positive passages $\{p_i^{\text{NOT}+}\}_{i=1}^{n-1} = \{p_i\}_{i=1}^{n} \setminus \{p_i^{\text{NOT}-}\}$.

4.4 Cyclic Consistency Filtering

Despite that questions generated by strong language generation models are fluent, coherence is not guaranteed. Inspired by the cyclic consistency in image generation (Zhu et al., 2017), we filter the generated questions by checking the cyclic consistency in question answering. Formally, we ask the language generation model whether the passage contains the answer to questions generated from the passage itself. Only questions that can be answered with their associated passages are deemed to be valid questions. After the cyclic consistency filtering, 1151 and 1258 questions are obtained based on MS MARCO and Natural Questions, respectively.

5 Data Analysis

In this section, we detail statistics of the proposed BOOLQUESTIONS, analyze the distribution of question types and display examples to provide a more intuitive understanding of our data.

5.1 Data Statistics

We display the number of questions, average number of positives and negatives for each question of BOOLQUESTIONS built upon MS MARCO and Natural Questions. Statistics are calculated on whole datasets and subsets of each type of question individually. Notably, 500 questions are initially generated for each type and we only provide statistics for the final datasets which are filtered by cyclic consistency mentioned in Section 4.4.

Statistics	BQ-MARCO	BQ-NQ
#questions (ALL)	1151	1258
avg #pos (ALL)	1.27	1.32
avg #neg (ALL)	0.63	0.38
#questions (AND)	354	403
avg #pos (AND)	1.00	1.00
avg #neg (AND)	0.94	0.59
#questions (OR)	469	485
avg #pos (OR)	1.58	1.74
avg #neg (OR)	0.35	0.21
#questions (NOT)	328	370
avg #pos (NOT)	1.13	1.11
avg #neg (NOT)	0.69	0.37

Table 1: Data statistics of BOOLQUESTIONS built upon MS MARCO (BQ-MARCO) and Natural Questions (BQ-NQ).

	pos (prediction)	neg (prediction)
pos (ground truth)	62.50%	9.38%
neg (ground truth)	3.13%	25.00%

Table 2: Confusion matrix of 32 random samples in BOOLQUESTIONS under human evaluation.

As AND questions are constructed with the logical conjunction of a disjunctive and a simple question, there is only 1 positive passage for each AND question. OR questions are built by disjunction of several simple questions, and thus more than 1 passages are annotated as positives. NOT questions also have more than 1 positive passage on average since they are collected by removing irrelevant passages within a cluster, where left passages are labeled as positives.

5.2 Data Quality

This work utilizes generative language models to construct benchmark datasets without human involvement, and thus it is vital to ensure the data quality for more precise evaluation on retrieval models. Several studies (Li et al., 2023; Zhao et al., 2023; Lightman et al., 2024) have demonstrated that large language models excel at verification tasks compared to generating novel information. Building on this insight, we deem GPT-4 as a robust filter in our data generation process. However, human involvement is critical for the quality assessment of the generated dataset. Therefore, we randomly sample 32 questions from our dataset and manually check the false-negative and false-positive rates of our proposed Cyclic Consistency Filtering. As the confusion matrix in 2 indicated, the filtering process effectively eliminates most false positives. Meanwhile, discarding false negatives does not significantly compromise the quality of the filtered data.

5.3 Question Types

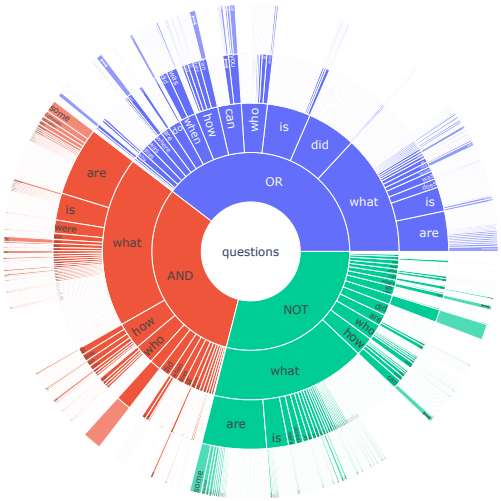

Figure 3: Question types covered in BOOLQUESTIONS. Questions are firstly grouped by the Boolean logic implied in questions and then heuristically categorized following the method in Yang et al. (2018). Colored blocks without labels indicate questions whose types can not be determined.

Following the approach in Yang et al. (2018), we heuristically analyze the question types of BOOLQUESTIONS, visualized in Figure 3. BOOLQUESTIONS covers a variety of question types including "what", "how", "who", yes/no questions and so on. General questions with leading "did" or "is" constitute a larger proportion in OR than in other types of questions. One possible explanation is that general questions have a large capacity for questions comprising the disjunction of multiple atomic questions.

5.4 Data Examples

We present examples of BOOLQUESTIONS in Table 3. AND and OR questions are easy to create as "and" and "or" serve as conjunctions in natural language. The main merit of leveraging GPT-4 to generate Boolean questions lies in NOT questions. As the example shown, GPT-4 paraphrases the NOT operator to "but is unrelated to" to form a more natural sentence while preserves the Boolean logic. Overall, the quality of generated questions is ensured in our data collection framework.

Question Type	Example(s)
AND	**Boolean Question:** How can I start a ***career in the accounting field*** **and** pursue an ***online degree program***? **Paragraph A (Positive):** If you want to work toward a ***career in the accounting field***, take the first steps with Penn Foster College. Contact us to learn more about our ***online Accounting degree program***. ... **Simple Question for Paragraph A:** Is Penn Foster College's ***online Accounting degree program*** designed for those aiming to start a career in accounting? **Paragraph B (Negative):** And indeed, if you're speaking primarily to promote your ***business*** ' say, you're offering a seminar on tax preparation ... **Simple Question for Paragraph B:** Do you need to charge for a seminar if promoting your ***accounting*** firm?
OR	**Boolean Question:** What are the impacts of ***global warming*** **or** ***climate change*** on nature and humans? **Paragraph A (Positive):** Global ***climate change*** will affect people and the environment in many ways. Some of these impacts, like stronger hurricanes and severe heat waves, ... **Simple Question for Paragraph A:** What are the potential impacts of global ***climate change*** on people and the environment? **Paragraph B (Positive):** ***Global warming*** is harming the environment in several ways including: 1 Desertification. 2 Increased melting of snow and ice. ... **Simple question for Paragraph B:** What environmental issues are caused by ***global warming***?
NOT	**Boolean Questions:** What causes ***upper abdomen pain*** **but is unrelated to** ***liver issues***? **Paragraph A (Positive):** Pain originating in the stomach or esophagus is often felt in the ***upper abdomen*** and can be due to heartburn, gastroesophageal reflux disease (GERD), or hiatal ... **Simple Question for Paragraph A:** What causes ***upper abdomen pain*** linked to the stomach or esophagus? **Paragraph B (Negative):** This pain is usually felt in the ***upper right part of the abdomen***, often under the rib cage, and is almost always associated with ***a swelling or enlargement of the liver***, acute inflammation or distention of the liver's surface, or any other ... **Simple Question for Paragraph B:** What symptoms are linked with ***liver enlargement*** or injury?

Table 3: Examples of BOOLQUESTIONS built upon MS MARCO corpus. We show in **blue bold** natural language phrases indicating the logical operation, ***green bold italics*** key information to retrieve the positive passages, and ***red bold italics*** key information to exclude the negative passages.

6 Experiments

In this section, we begin by describing the metrics employed in our experiments. We then evaluate the performance of prevalent dense retrievers on the proposed dataset, introduce two baseline methods for approaching the Boolean dense retrieval task, and finally, conduct a detailed analysis of these baseline methods.

6.1 Metrics for Boolean Dense Retrieval

Following previous works (Qu et al., 2021; Zhang et al., 2022; Ren et al., 2021), we adopt Mean Reciprocal Rank at top-k (MRR@k), which measures the average reciprocal rank of the first retrieved relevant passage, as the main metric of retrieval model performance in our experiments.

In addition, we propose an additional metric named Negative Recall at top-k (NegRecall@k) to evaluate the capability of retrieval models for tackling the logic of negation. Formally, NegRecall@k computes the recall of explicit negative passages for the top-k returned retrieval results. Lower NegRecall@k indicates that the retrieval system excludes the related passages in the negation more successfully.

6.2 Performance of Existing Dense Retrieval

We evaluate several strong dense retrievers on BOOLQUESTIONS, including those with BERT-style architecture and those fine-tuned from large language models that are capable of solving various natural language tasks.

From the results shown in Table 4 we can observe that higher retrieval performance on the original datasets indicates higher performance on our generated BOOLQUESTIONS. Notably, weaker retrievers like distilbert-base-v1 perform better on the original datasets than BOOLQUESTIONS, while stronger retrievers like gte-Qwen2-7B-instruct show better performance on BOOLQUESTIONS than the original datasets. However, stronger retrievers also suffer from high NegRecall on BOOLQUESTIONS, indicating that these retrievers return more passages possibly relevant to the queries but do not have the ability to distinguish the true positives.

Among the performance of AND, OR and NOT subset of BOOLQUESTIONS, OR subset enjoys the best performance, achieving 8.9% higher MRR@10 than AND subset on average. It can also be explained by the average number of posi-

Corpus	Model	MRR@10 ↑	BDR MRR@10 ↑				BDR NegRecall@10 ↓			
		ALL	ALL	AND	OR	NOT	ALL	AND	OR	NOT
MS MARCO	distilbert-cos-v5	33.78	29.11	31.46	39.44	11.80	**11.64**	**3.02**	**0.00**	33.08
	MiniLM-L12-cos-v5	32.75	31.00	32.32	43.56	11.60	11.88	3.77	**0.00**	**32.82**
	MiniLM-L6-cos-v5	32.25	31.09	33.80	43.98	9.73	13.08	4.15	**0.00**	36.15
	e5-base	36.26	36.68	41.91	50.60	11.14	12.92	3.58	**0.00**	36.41
	distilbert-dot-v5	37.25	37.61	39.92	53.04	13.04	19.50	5.28	0.61	54.62
	bert-base-dot-v5	38.08	39.89	44.36	55.31	13.01	17.01	4.15	**0.00**	48.72
	distilbert-base-tas-b	34.43	40.00	46.60	54.48	12.18	19.98	5.28	**0.00**	56.67
	e5-large-v2	35.74	41.20	49.66	54.84	12.56	16.37	3.58	**0.00**	47.44
	bge-large-en-v1.5	35.73	42.47	46.50	**57.90**	16.06	15.25	4.34	**0.00**	42.82
	gte-Qwen2-7B-instruct	**39.20**	**43.52**	**52.01**	55.54	**17.16**	15.01	5.28	**0.00**	40.77
Wikipedia	distilbert-base-v1	52.75	28.89	28.32	37.91	17.71	**20.39**	**15.84**	**14.42**	**32.42**
	dpr-single-nq-base	56.81	35.20	36.18	43.07	23.81	35.14	34.16	23.08	46.48
	stella-en-1.5B-v5	44.51	42.37	42.38	51.40	30.51	36.98	33.66	22.12	54.30
	bge-large-en-v1.5	57.54	64.98	67.47	78.34	44.77	61.29	55.94	39.42	87.50
	e5-base-v2	64.00	67.04	72.16	80.82	43.42	61.87	55.20	45.19	85.94
	e5-large-v2	65.93	68.61	74.20	**83.48**	43.04	66.01	60.64	48.08	89.06
	gte-Qwen2-7B-instruct	**66.64**	**71.55**	**75.65**	83.03	**52.04**	70.85	66.34	56.73	89.45

Table 4: Performance of strong dense retrieval models on the original dataset and BOOLQUESTIONS. MRR@10 without BDR prefix shows the model performance on the original dataset, while BDR MRR@10 and BDR NegRecall@10 denote the model performance on BOOLQUESTIONS. The performance of the whole dataset and subset for specific type of questions are shown individually.

tives. OR Questions own more positive passages than other subsets owing to the nature of logical disjunction in the construction of OR questions.

The most remarkable results lie in the NOT subset of BOOLQUESTIONS. MRR@10 for NOT subset of BOOLQUESTIONS is significantly worse than MRR@10 for other subsets. On the corpus of MS MARCO, most negative passages recalled are from the NOT questions in consideration of the limited NegRecall@10 on AND and OR subsets. Furthermore, even those retrievers fine-tuned from large language models that are believed to have strong language skills show the same trends with the traditional retrievers.

In view of these observations mentioned above, we draw the conclusion that logical negation is a challenging problem for current dense retrieval models. This aligns with our intuition that dense retrieval can only model the relevance of texts, but lack the capability of realizing the explicit irrelevance. We propose two baseline methods to tackle this problem, detailed in the next section.

6.3 Baseline Methods

We propose two baseline methods as a starting point for tackling the Boolean logic implied in language queries:

- **Boolean Operation on Decomposed Query** Inverted index is directly incorporated into the dense retrieval procedure in this baseline method. We ask the large language model to decompose complex Boolean questions into a Boolean expression of simple questions. For each simple question, we retrieve top-$2k$ relevant passages as the candidate retrieval results. Then candidate lists of these simple questions are merged based on the Boolean operator in the Boolean expression. Specifically, set intersection, union and difference are performed under AND, OR and NOT operations, respectively. Scores of candidates are recomputed by addition, maxing and subtraction in these situations. Finally, the merged candidate list is ranked according to the recomputed scores.

- **Boolean Contrastive Continuous Training** Following the data-driven schema of dense retrieval models, we propose to conduct continued training on pre-trained dense retrievers. We generate 2000 extra training data for AND, OR and NOT questions. These additional data are mixed with original training data to individually fine-tune the pre-trained model using the same objectives in pre-training. For reproducibility, we only conduct continued training on the distilbert-dot-v5 model whose training codes are publicly available.

6.4 Analysis

We implement the two baselines with distilbert-dot-v5, and show the results in Table 5. It can

Model	BDR MRR@10 ↑				BDR NegRecall@10 ↓			
	ALL	AND	OR	NOT	ALL	AND	OR	NOT
distilbert-dot-v5	**37.61**	**39.92**	**53.04**	13.04	19.50	5.28	0.61	54.62
Decomposed Query	32.34	26.73	51.55	10.93	**2.73**	3.21	0.61	**3.85**
Boolean Contrastive (AND)	35.98	37.44	51.49	12.24	10.87	**2.41**	**0.00**	31.44
Boolean Contrastive (OR)	37.23	39.07	52.92	12.83	11.29	2.50	**0.00**	32.65
Boolean Contrastive (NOT)	35.96	36.89	50.81	**13.71**	13.72	3.77	**0.00**	38.72

Table 5: Performance comparison of baselines on BOOLQUESTIONS. "Decomposed Query" and "Boolean Contrastive" denote the Boolean operation on decomposed query and Boolean contrastive continuous training on corresponding train set, respectively. The performance of the whole dataset and each subset are shown individually.

be observed from the results that both baselines reduce the NegRecall@k on all data samples. Boolean operation on decomposed query reduces the NegRecall@10 from 19.50% to 2.73%, nearly eliminating the false positives in the Boolean dense retrieval. However, MRR@10 also suffers from significant drops when conducting Boolean operation on decomposed queries. MRR@10 on the AND subset and the whole dataset drops 13.19% and 5.27%, respectively, indicating that many true positives are also rejected by this baseline. This is reasonable since there are various ways to decompose a complex question and the logic of decomposed questions and original questions may be not consistent. Naïve set subtraction on the retrieval results of decomposed questions exposes the low quality of question decomposition, resulting in low retrieval performance.

Boolean contrastive training on NOT questions impacts the pre-trained model more slightly, reducing MRR@10 from 37.61% to 35.96% and NegRecall@10 from 19.50% to 13.72% on the whole dataset. Interestingly, Boolean contrastive continuous training improves the MRR@10 and reduces NegRecall@10 simultaneously on the NOT subset of BOOLQUESTIONS, illustrating the improved ability to understand the logic of negation in natural language. Fine-tuning with AND and OR questions does not significantly improve MRR@10 but reduces NegRecall@10. A possible explanation is that AND and OR questions are relatively easier and fine-tuning on extra training set do not lead to further improvement but harm the performance. In contrast, NOT questions are more challenging for current retrievers, thus fine-tuning with them enhances their ability to tackle such questions. We expect that a better-designed learning-based approach would lead to more robust retrieval systems to address the Boolean dense retrieval task.

These results demonstrate that a dilemma exists in retrieving relevant passages for Boolean questions. While NOT questions are the main type of Boolean questions that are hard to tackle, overemphasizing them could lead to the exclusion of true positive passages in the returned list.

7 Conclusions

In this work, we formulate the task of Boolean dense retrieval and investigate whether current dense retrieval systems understand Boolean logic implied in natural language queries. By collecting a benchmark dataset BOOLQUESTIONS and evaluating the performance of prevalent dense retrievers, we find that NOT questions are challenging for current dense retrieval systems to understand correctly. Further, we explore Boolean operation on decomposed query and propose a contrastive continual training method that serves as a strong baseline for the research community.

Limitations

Dataset collected in this work is generated by large language models. We have not conduct human annotation or filtering on the generated dataset. Due to limited budget, the number of samples are not large enough to provide a complete training dataset. The diversity and coverage of proposed dataset are limited by the MS MARCO and Natural Questions dataset since we build our dataset based on their corpus. Besides, we only focus on English, evaluations on other languages is limited.

Acknowledgements

This work is supported by National Key R&D Program of China under Contract 2022ZD0119802, National Natural Science Foundation of China under Contract 623B2097 and the Youth Innovation Promotion Association CAS. It was supported by GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC, and the Supercomputing Center of the USTC. This work was also supported by Ant Group Research Fund.

References

Negar Arabzadeh, Bhaskar Mitra, and Ebrahim Bagheri. 2021. Ms marco chameleons: Challenging the ms marco leaderboard with extremely obstinate queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4426–4435, New York, NY, USA. Association for Computing Machinery.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A human generated machine reading comprehension dataset. *arXiv:1611.09268*.

Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, , Roberto Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of ms marco passage ranking dataset. *Preprint*, arXiv:2108.13897.

Abraham Bookstein. 1980. Fuzzy requests: An approach to weighted boolean searches. *Journal of the American Society for Information Science (pre-1986)*, 31(4):240.

Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. 2016. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017b. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. 2022. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval*, 16(3):178–317.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems*, 40(4).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The International Conference on Learning Representations*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9).

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In

Companion Proceedings of the The Web Conference 2018, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14032–14047, Toronto, Canada. Association for Computational Linguistics.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1291–1299. ACM.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artières, Axel-Cyrille Ngonga Ngomo, Norman Heino, Éric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

W.G. Waller and Donald H. Kraft. 1979. A mathematical model of a weighted boolean retrieval system. *Information Processing & Management*, 15(5):235–245.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages

2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

ChengXiang Zhai. 2007. Statistical language models for information retrieval. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Tutorial Abstracts*, pages 3–4, Rochester, New York. Association for Computational Linguistics.

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. In *The International Conference on Learning Representations*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4).

Victor Zhong, Weijia Shi, Wen-tau Yih, and Luke Zettlemoyer. 2023. RoMQA: A benchmark for robust, multi-evidence, multi-answer question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7055–7067, Singapore. Association for Computational Linguistics.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *ACM Computing Surveys*, 38(2):6–es.

Justin Zobel, Alistair Moffat, and Kotagiri Ramamohanarao. 1998. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems*, 23(4):453–490.

A Prompts

A.1 System Message for Questioner:

You are an experienced questioner and retrieval system tester. You need to generate questions based on the given paragraphs and related instructions, which will be used as queries to test if the retrieval system can understand the Boolean logic contained in natural language. The questions you pose should align as closely as possible with the retrieval system's scenario, meaning the language style of the questions should resemble that of a search engine user. Besides, please vary your expressions more and avoid sticking to just a few ways of saying things. Note, you only need to output one question no longer than 32 words, without any extra content.

A.2 System Message for Answerer:

You are an expert answerer who needs to provide answers to the questions based on the given paragraphs. If the question can be answered by the paragraph(s), please provide a brief answer. If the question cannot be answered by the paragraph(s), please respond with "Cannot answer". Note, you only need to output one answer no longer than 64 words or "Cannot answer", without any extra content.

A.3 Prompt for Simple Questioner:

Please propose a question that can be answered by the following paragraph.

[*PARAGRAPH*]

A.4 Prompt for Disjuntive Questioner:

Please propose a question that can be answered by any of the following paragraphs. Please make sure that each paragraph can provide answers to the question individually.

[*PARAGRAPH*]

A.5 Prompt for AND Question Converter:

I need to test whether the retrieval system can understand the logical conjunction (AND) implied in natural language. Please generate a new question by adding constraints to the question "[QUESTION]", so that only paragraphs marked with [positive] provide the answer to the new question, while paragraphs marked with [negative] cannot provide the answer.

[*POSITIVE PARAGRAPHS*]

[*NEGATIVE PARAGRAPHS*]

A.6 Prompt for OR Question Converter:

I need to test if the retrieval system can understand the logical disjunction (OR) implied in natural language. Please convert the following expression containing the logical disjunction (OR) into a natural language question.

[*LOGICAL EXPRESSION*]

A.7 Prompt for NOT Question Converter:

I need to test if the retrieval system can understand the logic of negation (NOT) implied in natural language. Please convert the following expression containing the logic of negation (NOT) into a natural language question.

[*LOGICAL EXPRESSION*]

A.8 Prompt for Answerer:

Please provide a brief answer to the following question according to the given paragraph(s). If the question cannot be answered by the paragraph(s), please respond with "Cannot answer".

question:
[*QUESTION*]

paragraphs:
[*PARAGRAPHS*]