information

Article

Enhancing Biomedical Question Answering with Large Language Models

Hua Yang [1], Shilong Li [1,2,*] and Teresa Gonçalves [3,4]

[1] School of Artificial Intelligence, Zhongyuan University of Technology, Zhengzhou 450007, China; huayang@zut.edu.cn
[2] School of Computer Science, Zhongyuan University of Technology, Zhengzhou 450007, China
[3] Department of Computer Science, University of Évora, 7000-671 Évora, Portugal; tcg@uevora.pt
[4] VISTA Lab, Algoritmi Center, University of Évora, 7000-671 Évora, Portugal
* Correspondence: lishilong@zut.edu.cn

Abstract: In the field of Information Retrieval, biomedical question answering is a specialized task that focuses on answering questions related to medical and healthcare domains. The goal is to provide accurate and relevant answers to the posed queries related to medical conditions, treatments, procedures, medications, and other healthcare-related topics. Well-designed models should efficiently retrieve relevant passages. Early retrieval models can quickly retrieve passages but often with low precision. In contrast, recently developed Large Language Models can retrieve documents with high precision but at a slower pace. To tackle this issue, we propose a two-stage retrieval approach that initially utilizes BM25 for a preliminary search to identify potential candidate documents; subsequently, a Large Language Model is fine-tuned to evaluate the relevance of query–document pairs. Experimental results indicate that our approach achieves comparative performances on the BioASQ and the TREC-COVID datasets.

Keywords: biomedical question answering; large language models; BM25

check for updates

Citation: Yang, H.; Li, S.; Gonçalves, T. Enhancing Biomedical Question Answering with Large Language Models. *Information* **2024**, *15*, 494. https://doi.org/10.3390/info15080494

Academic Editors: Zhigang Chu, Heming Jia and Emilio Matricciani

Received: 2 July 2024
Revised: 2 August 2024
Accepted: 13 August 2024
Published: 19 August 2024

1. Introduction

Question Answering (QA) is the basis for advanced tools such as chatbots [1–4], search engines [5], and virtual assistants [6–8]. As a downstream task, Question Answering suffers from pipeline errors because it often depends on the quality of multiple upstream tasks, such as co-reference resolution [9], anaphora resolution [4], named entity recognition [10], Information Retrieval [11], and tokenization [12]. As a result, the QA systems are driving substantial research focused on enhancing Natural Language Processing methods [13], QA datasets [14,15], and Information Retrieval techniques [11,16,17]. These advancements have enabled the field to progress from simple keyword matching to sophisticated contextual and semantic retrieval systems [5]. However, most of these technologies are concentrated in open-domain applications [18], and the specific challenges faced by the biomedical field remain largely unresolved.

In the biomedical field, one of the primary challenges in addressing complex questions is the precise formulation of medical queries. Not only does this type of query require in-depth medical expertise and precise knowledge of terminology, but it also requires exceptional accuracy in question formulation. Even slight variations in phrasing can lead to vastly different answers. Furthermore, as medical questions typically require thorough research and validation by medical experts, the process can be quite time-consuming, with each question potentially taking up to four hours to answer [19]. This high complexity and time requirement underscore the unique need for high-quality QA systems in the biomedical domain, systems that must handle intricate and rigorous medical queries.

Meanwhile, developing high-quality QA systems for the biomedical field is not an easy task. The foremost challenge is the extreme scarcity of high-quality datasets. This scarcity is primarily due to the necessity for deep professional knowledge in creating these datasets. Additionally, the confidentiality and ethical constraints surrounding medical data limit its scale and availability. The high costs associated with data collection and annotation by domain experts further reduce the availability of these datasets. Consequently, techniques commonly used in open-domain QA, which rely on abundant data, may not be applicable in the biomedical context. These challenges collectively constitute the major obstacles in developing biomedical QA systems.

In order to enhance system performance, researchers have employed various methods. Traditional sparse retrievers like BM25 [20,21] rely on lexical matching, whereas dense retrievers like DPR [22] utilize deep learning models to extract features from both questions and documents, enabling a deeper semantic understanding. Sparse retrievers, while cost-effective and practical, often fall short in semantic matching due to the vocabulary mismatch problem [23,24]; dense retrievers, although superior in semantic matching [25,26], require substantial data and computational resources during training and inference. Thus, term-based sparse retrieval still has a place in document retrieval [27,28].

To address these limitations, we propose a BM25-LLMs biomedical question retrieval system that combines the classical BM25 [20] algorithm with Large Language Models (LLMs). First, BM25 [20] is used to rank the potentially relevant documents, and then the query and candidate documents are re-ranked based on the knowledge of the LLMs. The final step is to calculate and return the exact document similarity score. Our comparative experiments indicate that the proposed BM25-LLMs system demonstrates superior performance over several existing retrieval models. The system achieved competitive results across the evaluated datasets, suggesting notable improvements in retrieval effectiveness. The experiments carried out on the two datasets, the BioASQ [29] dataset and the TREC-COVID [30] dataset search tasks, proved the effectiveness of our method.

2. Related Work

In this section, we begin by discussing the representatives of biomedical Question Answering systems. Next, we detail the related work on text re-ranking. Following that, we present the representative retrieval models.

2.1. Biomedical Question Answering Systems

This section explores several key biomedical Question Answering systems, highlighting their functionalities and limitations within the domain.

2.1.1. MedQA: A Medical Quality Assurance System

Researchers [31] have developed the MedQA medical quality assurance system, comprising five key components: (1) question classification, (2) query generation, (3) document retrieval, (4) answer extraction, and (5) text summarization. Although the MedQA system provides concise summaries that may address medical inquiries, its current functionality is constrained: it can solely respond to definitional queries.

2.1.2. HONQA: Quality Assurance via Certified Websites

Cruchet et al. [32] introduced HONQA, a biomedical quality assurance system designed to retrieve phrases from HON-certified websites and present them as responses to biomedical queries. However, its current functionality falls short in providing accurate replies to question types such as yes/no inquiries and factoids.

2.1.3. EAGLi: Extracting Answers from MEDLINE Records

Additionally, Gobeill et al. [33] developed EAGLi, another biomedical QA system aimed at extracting solutions to biological queries from MEDLINE records. However, its scope is limited to definitional and factoid inquiries, thus constraining EAGLi's ability to effectively handle Wh-type queries.

2.1.4. AskHERMES: Clinical QA with Concise Summaries

Cao et al. [34] introduced AskHERMES, a clinical QA system that provides concise summaries in response to ad hoc clinical queries. However, the system is limited in its response capabilities, as it only offers a single answer type in the form of multi-sentence passages across all question types.

2.1.5. SemBT: Biomedical QA via Semantic Relations

Hristovski et al. [35] presented SemBT, a biomedical QA system that relies on semantic relations extracted from biomedical literature.

2.2. Text Re-Ranking

In the context of machine learning, text ranking is predominantly achieved through supervised learning to rank [36]. This approach involves designing a feature-based ranking function, utilizing hand-crafted features as input, and training the ranking function using relevance judgments. Despite its flexibility, the learning-to-rank method still depends on human efforts in feature engineering.

With the introduction of pre-trained language models with context, researchers have moved away from the manual specification of text features for similarity modeling, revolutionizing the field. Notably, MonoBERT [37], which employs a cross-encoder architecture, was the first to use pre-trained models for text ranking, demonstrating its effectiveness as a re-ranking method and representing the next generation of interactive ranking methods. In contrast to the extensive research on dense retrieval, studies on cross-encoders have remained relatively stagnant, partly due to the efficiency and speed of retrieval models in identifying relevant documents from large-scale texts; however, re-ranking models remain crucial as even the best retrieval model outputs can be enhanced through re-ranking, with optimal results achieved on popular text ordering benchmark datasets when an effective first-stage retrieval is combined with multi-stage re-ranking [38].

2.3. Retrieval Models

The early retrieval models were primarily vector space models such as TF-IDF [39], probabilistic models such as BM25 [20], and statistic language models such as N-gram [40]. These models typically construct representations of queries and documents based on the Bag of Words assumption, which treats each text as a set of words without considering grammatical structure or word order.

Recently, Large Language Models that can capture substantial syntactic and semantic information have led to the development of more sophisticated text retrieval models. These LLMs, applied in the text retrieval area, can usually be classified as sparse retrieval and dense retrieval. Sparse retrieval encompasses a range of techniques, including neural weighting schemes and sparse representation learning. The fundamental approach to neural weighting is to develop a neural model based on semantics, rather than predefined heuristic algorithms, to predict the weight of terms. This can be achieved through techniques such as DeepTR [41] and DeepCT [42]. In addition to predicting the weight of terms, another method is to utilize a Seq2Seq model to enrich documents with additional terms. This can be exemplified by DocTTTTTQuery [43] and SparTerm [44]. DeepImpact [45] integrates both approaches. It employs Doc2Query to enrich documents and then utilizes a pre-trained language model with context to estimate the importance of words in the document. Sparse representation learning is centered on the construction of sparse vectors for queries and documents, thereby capturing the semantics of each input text, such as UHD BERT [46]. In contrast to sparse retrieval, dense retrieval transforms sparse representations into dense representations, thereby enhancing the ability to capture semantics. Dense retrieval typically employs a dual-encoder architecture, which utilizes independent network structures to learn the representations of queries and texts separately. The matching layer is frequently implemented using a straightforward similarity function. To facilitate online services, a neighborhood algorithm is typically employed to index and search the learned vector representations [47]. Based on the learned document representation, dense retrieval can be classified into two categories: term-level representation learning and document-level representation learning. Examples of term-level representation learning include DESM [48], COIL [49], and ColBERT [25]. Examples of document-level representation learning include DSSM [50], ARC-I [51], and DPR [22].

3. Methodology

This section presents the methodologies and techniques employed in the development of a QA system, which is specifically designed for use in the biomedical field.

3.1. Two-Stage Retrieval

A two-stage re-ranking process is proposed, as illustrated in Figure 1.

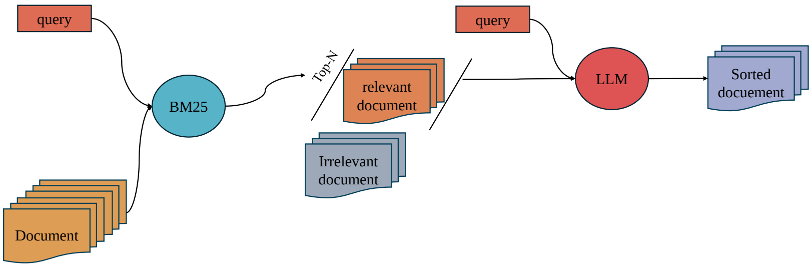


Figure 1. An overview of the proposed two-stage BM25-LLMs retrieval model. The first stage employs BM25 for sparse search, and the second stage utilizes a Large Language Model for accurate re-ranking.

In the initial stage, the BM25 [20] algorithm is employed. BM25 is a probabilistic retrieval model known for its efficiency and effectiveness in identifying documents that match the query terms. The primary objective of this stage is to maximize the retrieval rate, ensuring that the candidate document set contains as many relevant documents as possible.

In the second stage, we select results from the first-stage search results at various intervals. These selected results are then expanded to the top 1000 documents to enhance coverage. As depicted in Figure 2, the efficiency of the search process does not exhibit a significant increase when the search range surpasses 100 documents. Therefore, we select 100 documents as the result of the first stage.

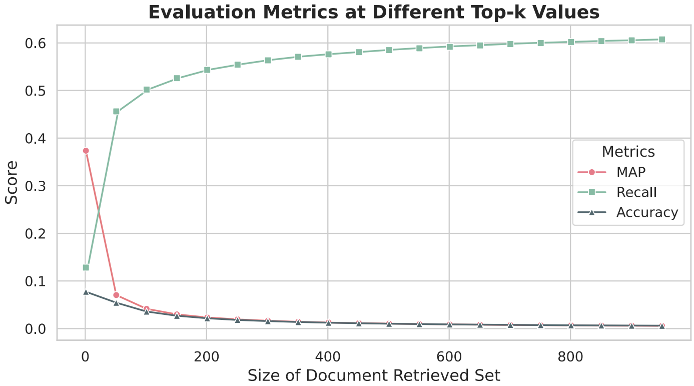


Figure 2. Evaluation metrics at various top-k values, illustrating the Mean Average Precision (MAP), Recall, and Accuracy for varying sizes of the document retrieval set in this experiment. It demonstrates that a top-100 retrieval set is the k-value beyond which increases in the retrieval set size do not result in proportional improvements in accuracy.

Subsequently, the top 100 documents retrieved are processed by a Large Language Model. This model generates a normalized probability of "Yes" based on its logits output sequence. The resulting probability serves as the relevance score for each document. Consequently, the candidate documents are reordered according to these relevance scores.

We utilized the BAAI General Embedding (BGE) model [52,53], which utilizes the Gemma model [54] as its initial model, and during the pre-training stage it performs two pre-training tasks that are well-suited for global semantic representation: Embedding-Based Auto-Encoding (EBAE) and Embedding-Based Auto-Regression (EBAR). EBAE is a self-encoding task that employs text embeddings to predict the tokens of the input sentence itself. The objective is to enable the text embeddings to capture the global semantics of the entire input text through the self-encoding process. EBAR is a self-regression task that employs text embeddings to predict the next sentence of the input text, which facilitates the establishment of a relationship between the query and the document.

3.2. Prompt Strategy

A query will serve as the input to the system, which will employ the BM25 algorithm to retrieve a set of documents. These documents will be paired with the query and presented with the following prompt: *"Given a query A and a passage B, indicate whether the passage contains an answer to the query by selecting either 'Yes' or 'No'"*, as shown in Figure 3.

Subsequently, enter the query–document pair into the model in the form "[Prompt] A:query_text <space> B:passage_text" and identify the score corresponding to the token 'Yes' in the model output. Thereafter, use the Sigmoid function to map the score to the $[0, 1]$ interval and reorder the documents according to this score.

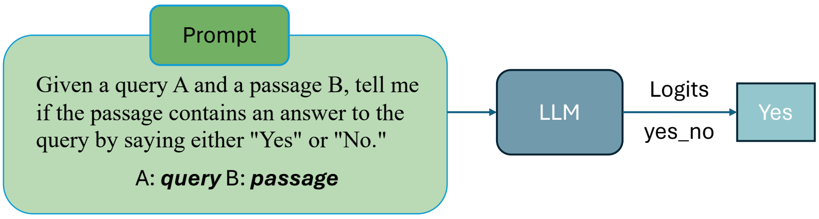

Prompt
Given a query A and a passage B, tell me if the passage contains an answer to the query by saying either "Yes" or "No."
A: query B: passage
LLM
Logits
yes_no
Yes

Figure 3. The prompt instructs the model to decide if the passage answers the query by responding with 'Yes' or 'No'. The input consists of a query (A) and a passage (B), which are processed by a Large Language Model. The value of the token 'Yes' is found in the logits generated by the model for use.

3.3. Hard Negative Mining and Data Preparation

To enhance the dataset for fine-tuning Large Language Models (LLMs), we implemented hard negative mining techniques during the data preparation process. This approach significantly improved the model's capacity to distinguish between relevant and irrelevant paragraphs, thereby enhancing the accuracy of paragraph re-ranking.

Our process comprises several key steps to effectively implement hard negative mining. Initially, for each query we retrieve the top 200 documents based on their BM25 score. These documents are then encoded using the bge-reranker [53], with the instruction "Generate a representation of this sentence for use in retrieving related articles" appended to each document. This encoding process yields a 768-dimensional vector for each document. To facilitate efficient retrieval, we employ FAISS [55] to create an index and store these vectors. Subsequently, utilizing this FAISS index, we retrieve the k-nearest neighbors for each query. In this selection process, we consider the first retrieved document as a positive sample, while the subsequent k-1 documents are treated as negative samples. This approach allows us to create a diverse and challenging set of training examples, enhancing the model's ability to discriminate between relevant and irrelevant paragraphs in the context of our re-ranking task.

As illustrated in Figure 4, each sentence is converted into an embedding vector, which represents the semantic features of the sentence in a high-dimensional space. A similarity of 1 indicates that the sentences are identical. A similarity of 0.3 signifies that the text embeddings for "A group is playing basketball" and "A child is reading a book" are markedly disparate, whereas a similarity of 0.8 indicates that the text embeddings for "A group is playing basketball" and "A group is watching a basketball" are slightly similar, although the meaning differs in the real content.

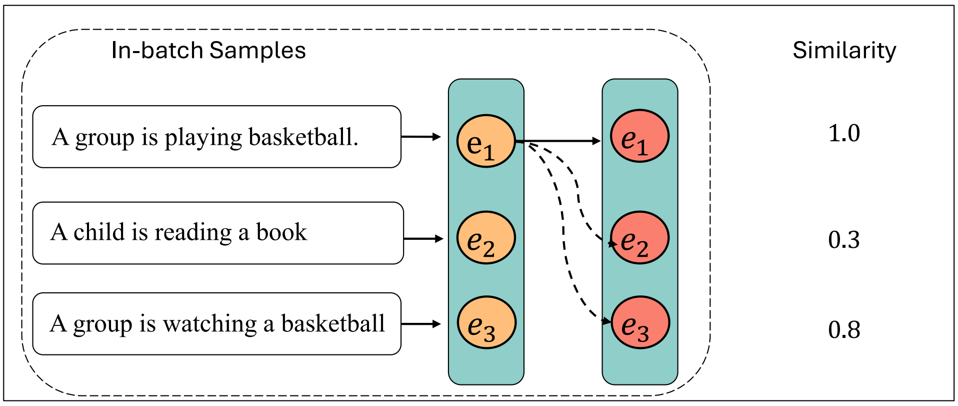

In-batch Samples
Similarity
A group is playing basketball.
A child is reading a book
A group is watching a basketball
e_1
e_2
e_3
e_1
e_2
e_3
1.0
0.3
0.8

Figure 4. In-batch hard negative mining. The figure illustrates the process of in-batch hard negative mining, where the negative samples are selected from the same batch as the positive samples. Solid arrows represent positive sample pairs with high similarity (1.0), while dashed arrows indicate negative sample pairs with lower similarity.

4. Datasets Analysis

4.1. BioASQ

BioASQ [29] is an EU-funded biomedical semantic indexing and Question Answering challenge that provides accumulated sets of biomedical questions and gold standard answer data. Questions within the BioASQ data are associated with scientific articles from PubMed (https://pubmed.ncbi.nlm.nih.gov, accessed on 15 August 2024) and GoPubMed [56], which are journals for publishing scientific research.

To investigate the distribution of query lengths in the BioASQ 11b dataset, we analyzed the histogram of query lengths and the Q–Q plot of log-transformed frequencies. Figure 5 shows the histogram of query lengths with a fitted log-normal distribution. The observed data generally follows the shape of the log-normal curve, with some deviations, particularly at the tails. Figure 6 presents the Q–Q plot of log-transformed frequencies. While there are some deviations from the theoretical line, especially at the extremes, the overall trend suggests that the log-normal distribution provides a reasonable approximation for the query length distribution. Table 1 presents examples from the BioASQ dataset.

Table 1. BioASQ dataset entry example.

	Question	**Answer**
1	What is CHARMS with respect to medical review of predictive modeling?	Checklist for critical Appraisal and data extraction for systematic Reviews of predictive Modelling Studies (CHARMS)
2	What is AUROC in the context of predictive modeling?	Area under the receiver operator characteristics curve
3	Is casimersen effective for the treatment of Duchenne muscular dystrophy?	Yes

Following this idea [34], we extract the cleaned text segment from the PubMed articles. This step involves the removal of tables, diagrams, boxes, and lists. The BioASQ Task11 B question–answer dataset, comprising approximately 4700 question–answer pairs for training purposes, was utilized. Each question is accompanied by relevant documents, segments, concepts, RDF triples, and exact and ideal answers. RDF, or Resource Description Framework, is a standard model for data interchange on the Web, particularly useful for representing linked data. In the context of BioASQ, RDF triples provide a structured way to represent biomedical knowledge, allowing for efficient querying and integration of complex information. Each RDF triple consists of a subject, predicate, and object, which together express a specific fact or relationship within the biomedical domain. Since some questions in the dataset lack exact answers, and each question has an ideal answer, the ideal answer was employed as the label data.

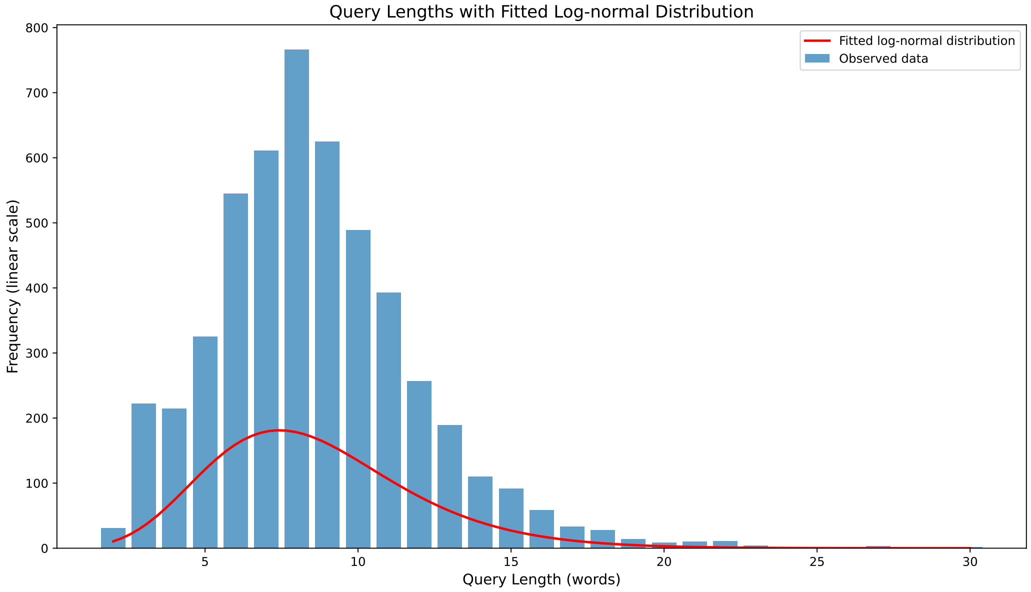


Figure 5. Distribution of query lengths with fitted log-normal curve. Observed data (blue) shows raw frequencies; fitted log-normal distribution (red) indicates approximate log-normal behavior with modal length around 5 words.

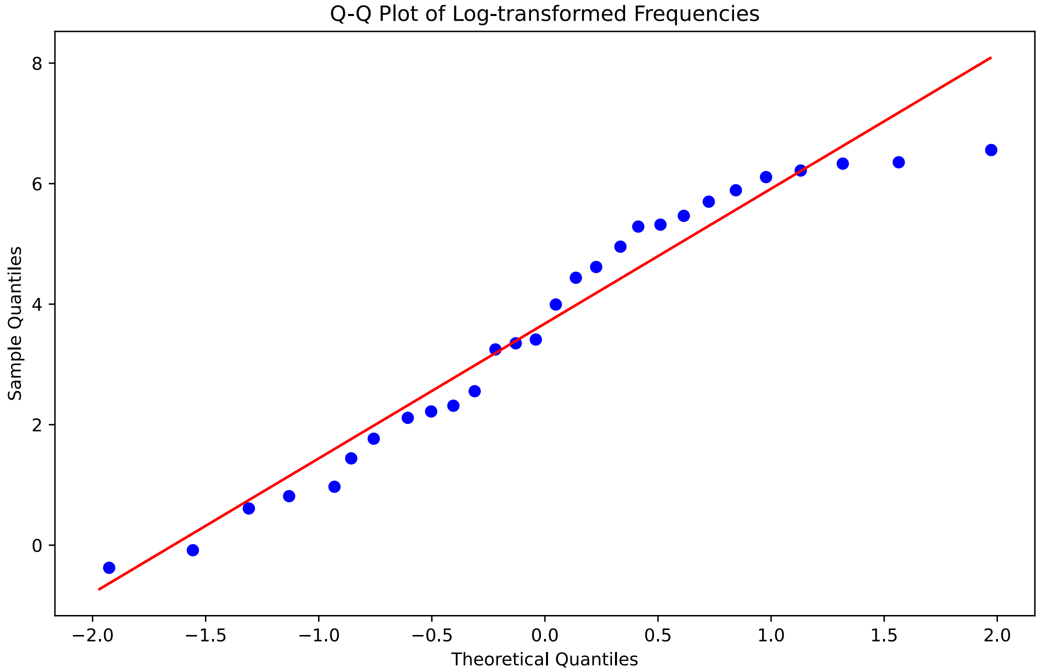


Figure 6. Q–Q plot of log-transformed frequencies. Points roughly follow a straight line, suggesting reasonable agreement with a normal distribution.

Consistent with preceding versions, this edition examines four distinct categories of inquiries: ‘yes/no’, ‘factoid’, ‘list’, and ‘summary’ questions [29], as shown in Table 2:

(1) Yes/no questions: questions that require either a 'yes' or 'no' answer. (2) Factoid questions: questions that require a particular entity name, a number, or a similar short expression as an answer. (3) List questions: questions that expect a list of entity names. (4) Summary questions: questions that expect short summaries as the answer.

Table 2. Statistics of the BioASQ dataset.

Partition	Yes/No	Factoid	List	Summary	Documents	Snippets	Total
Training	1271	1417	901	1130	9.01	12.03	4719
Test 1	24	19	12	20	2.48	3.28	75
Test 2	24	22	12	17	2.95	4.29	75
Test 3	24	26	18	22	2.66	3.77	90
Test 4	14	31	24	21	2.80	3.91	90

To build a corpus suitable for BioASQ, we utilized the PubMed annual baseline document collections, which span from 2002 to 2023. This extensive corpus comprises titles and abstracts of approximately 35 million documents, with the 2023 collection being the most recent. In the process of constructing the corpus, in order to ensure the integrity of the data, we identified and excluded documents that lacked a title, abstract, or both. This was often due to licensing restrictions or language issues.

4.2. TREC-COVID

TREC-COVID [30] stands as a pivotal dataset tailored for Information Retrieval and text mining, aimed at facilitating researchers' exploration of scientific literature pertaining to COVID-19. For the TREC-COVID dataset, we used the corpus CORD-19 [57], which contains new publications and preprints on the topic of COVID-19, as well as relevant historical studies on coronaviruses, including SARS and MERS. The distribution of query lengths within the TREC-COVID dataset is presented in Figure 7.

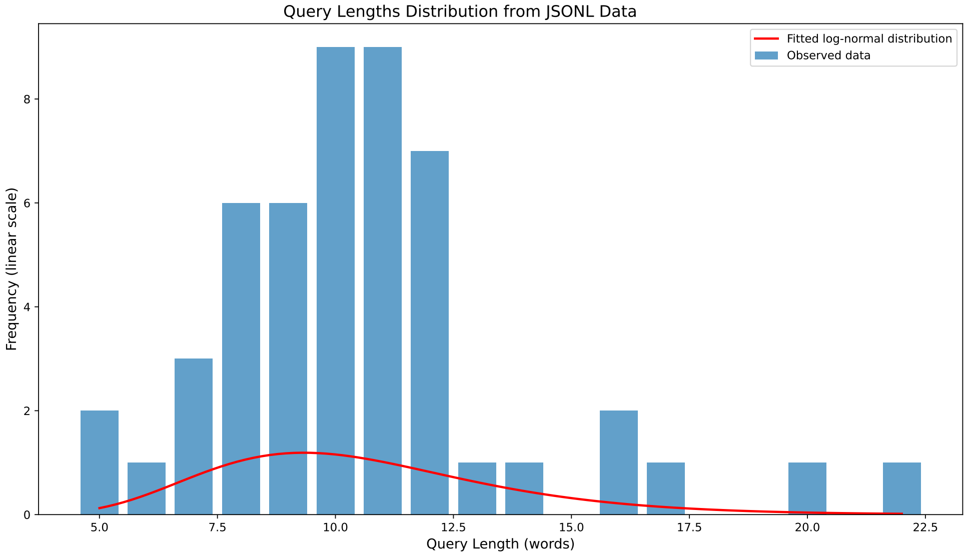


Figure 7. This histogram illustrates the distribution of query lengths within the TREC-COVID dataset.

To illustrate the types of questions about COVID-19 and the nature of the corresponding answers, Table 3 randomly presents three questions from the TREC-COVID dataset and the corresponding answers.

Table 3. TREC-COVID dataset entry example.

	Question	Answer
1	What is the origin of COVID-19?	Although primary genomic analysis has revealed that severe acute respiratory syndrome coronavirus (SARS CoV) is a new type of coronavirus...
2	How does the coronavirus respond to changes in the weather?	Abstract: In this study, we aimed at analyzing the associations between transmission of and deaths caused by SARS-CoV-2 and meteorological variables...
3	Will SARS-CoV2 infected people develop immunity? Is cross protection possible?	Of the seven coronaviruses associated with disease in humans, SARS-CoV, MERS-CoV and SARS-CoV-2 ...

5. Experiments and Results

This section first presents the setup for our experiments and the evaluation metrics used to assess the performance of the proposed method; next, the indexing strategy is discussed; then, the detailed fine-tuning of the BM25 model is presented; and finally, the experimental results of the proposed two-stage Retrieval with LLMs are detailed.

5.1. Setup

The experiments were carried out in PyTorch 2.0.1, CUDA 12.2, and cuDNN 9.1.0 software environments, and the hardware environment was an Intel i7-10700K CPU, Nvidia RTX A5000 GPU.

To facilitate efficient and precise retrieval, we created sparse indexes for each year Using Pyserini [58]. Pyserini [58] is a robust toolkit for replicable Information Retrieval research, which facilitated the construction of an index for these documents. This indexing approach significantly enhanced the search accuracy for documents relevant to specific annual queries, thereby improving the overall precision of our Information Retrieval process.

5.2. Evaluation Metrics

We evaluate the performance of our model using standard Information Retrieval metrics: Precision, Recall, and the F_1 Score. To account for the ranking order of retrieved documents, we also employ Average Precision (AP) and Mean Average Precision (MAP), which are widely used to measure the effectiveness of ranked retrieval systems. While Precision, Recall, and F_1 Score provide a foundational evaluation, AP and MAP offer insights into the order sensitivity of retrieval results.

AP considers the precision at each relevant item in the ranked list, and MAP, calculated as the mean of AP values across all queries, quantifies the overall quality of the search results. Specifically, MAP is defined as:

$$\text{MAP} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{m_q}\sum_{k=1}^{m_q}\text{Precision}(R_{qk}) \tag{1}$$

where Q is the total number of queries in the evaluation set, m_q is the number of relevant documents for query q, Precision(R_{qk}) is the precision at the rank of the k-th relevant document for query q, and R_{qk} represents the set of ranked retrieval results from the top result until you get to the k-th relevant document.

This formulation of MAP takes into account both the precision and the ranking of relevant documents, providing a comprehensive measure of retrieval performance across multiple queries.

Additionally, for the BioASQ dataset, the evaluation also incorporates the Geometric Mean Average Precision (GMAP). GMAP is particularly sensitive to the performance of difficult queries, as it emphasizes the geometric mean of individual average precisions across the queries, thus penalizing poor performance on any single query more heavily than MAP.

$$\text{GMAP} = \exp\left(\frac{1}{n} \cdot \sum_{i=1}^{n} \ln(AP_i + \epsilon)\right) \tag{2}$$

where ϵ is a small constant to prevent the logarithm from being undefined.

Similarly, for the TREC-COVID dataset, the evaluation also incorporates the Normalized Discounted Cumulative Gain (NDCG). NDCG is a robust evaluation metric widely used to assess the performance of search engines and recommendation systems.

5.3. Indexing

In order to index the content, stop words were initially removed, and then the document title and abstract were connected. This provided the BM25 algorithm with a more comprehensive context, enabling it to more effectively identify potentially relevant documents.

During the indexing of the PubMed baseline 2023 corpus, when both PDF and PubMed XML versions of documents are available, the text from the PubMed XML is preferred. This preference is due to the PubMed XML text being more concise and structured. For the CORD-19 corpus, only the text version is available. Consequently, the title and abstract are concatenated and used directly as the index content.

5.4. Fine-Tuned BM25 Model

The BM25 algorithm is a well-established model in document retrieval. It operates as a bag-of-words model, which evaluates the relevance of documents based on the frequency and distribution of query terms within them.

Given a query, q, comprising terms, $q_1, q_2, \ldots, q_n$, the BM25 score of a document, D, is calculated as follows:

$$\text{score}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \tag{3}$$

In this equation, $f(q_i, D)$ represents the term frequency of the query term q_i, within document D. $|D|$ denotes the length of document D in terms of words, while avgdl signifies the average document length within the corpus of documents under consideration. $IDF(q_i)$ denotes the inverse document frequency of the query term q_i, calculated as:

$$\text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right) \tag{4}$$

where N signifies the total number of documents in the corpus and $n(q_i)$ denotes the number of documents containing the query term q_i.

While parameters k_1 and b are typically regarded as free parameters, advanced optimization techniques can be employed for their selection. To identify the optimal values of these parameters, we conducted a grid search on the BioASQ [29] and TREC-COVID [30] datasets.

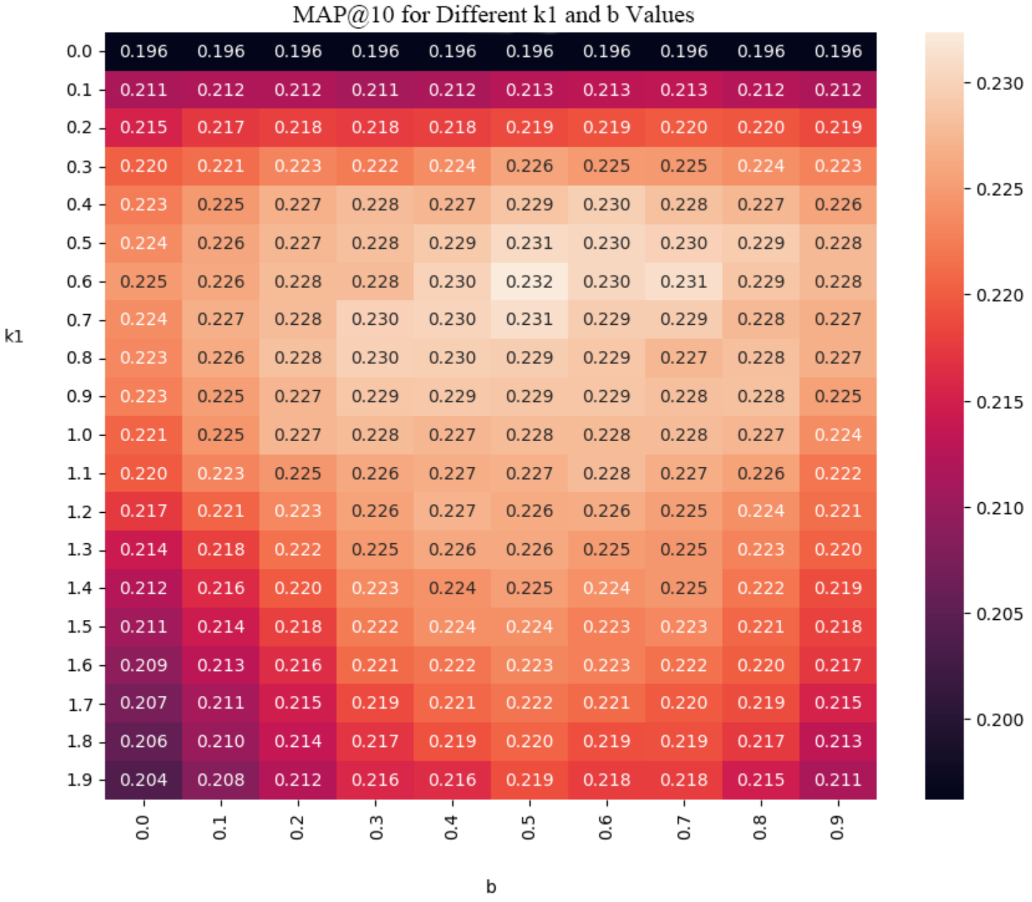
The grid search was designed to cover a broad spectrum of potential values for both parameters. For k_1, we examined values ranging from 0.0 to 1.9, incrementing by 0.1 at each step. Similarly, for the b parameter, we investigated values from 0.0 to 0.9, also with 0.1 increments.

For hyperparameter optimization, we employed a data partitioning strategy. Each dataset was divided into three subsets: a training set comprising 60% of the data, a validation set with 20%, and a test set containing the remaining 20%.

The grid search was performed using the validation set to identify the best hyperparameters, while preserving the integrity of the test set for final evaluation. This approach was consistently applied to both the BioASQ and TREC-COVID datasets, maintaining the same proportions for training, validation, and test sets.

To assess the performance of each parameter combination, we employed the MAP@10 metric. MAP@10 is a widely adopted evaluation metric in information retrieval, which quantifies the quality of search results by ranking relevant documents within the top 10 results for a given set of queries. The values of the k_1 and b parameters were used to determine these rankings. A higher MAP@10 score denotes superior performance, indicating that a greater number of relevant documents are positioned higher in the search results (the code repository for grid search can be found at https://github.com/Firestl/Enhancing-Biomedical-Question-Answering-with-Large-Language-Models, accessed on 15 August 2024.

For the BioASQ dataset, the optimal settings found according to Figure 8 are $k_1 = 0.6$ and $b = 0.5$, while for the TREC-COVID dataset, the optimal settings found according to Figure 9 are $k_1 = 1.9$ and $b = 0.3$. These parameters significantly improved the retrieval performance, demonstrating their effectiveness over the commonly used settings in traditional BM25 implementations, where $k_1 = 1.2$ and $b = 0.75$.



	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196
0.1	0.211	0.212	0.212	0.211	0.212	0.213	0.213	0.213	0.212	0.212
0.2	0.215	0.217	0.218	0.218	0.218	0.219	0.219	0.220	0.220	0.219
0.3	0.220	0.221	0.223	0.222	0.224	0.226	0.225	0.225	0.224	0.223
0.4	0.223	0.225	0.227	0.228	0.227	0.229	0.230	0.228	0.227	0.226
0.5	0.224	0.226	0.227	0.228	0.229	0.231	0.230	0.230	0.229	0.228
0.6	0.225	0.226	0.228	0.228	0.230	0.232	0.230	0.231	0.229	0.228
0.7	0.224	0.227	0.228	0.230	0.230	0.231	0.229	0.229	0.228	0.227
0.8	0.223	0.226	0.228	0.230	0.230	0.229	0.229	0.227	0.228	0.227
0.9	0.223	0.225	0.227	0.229	0.229	0.229	0.229	0.228	0.228	0.225
1.0	0.221	0.225	0.227	0.228	0.227	0.228	0.228	0.228	0.227	0.224
1.1	0.220	0.223	0.225	0.226	0.227	0.227	0.228	0.227	0.226	0.222
1.2	0.217	0.221	0.223	0.226	0.227	0.226	0.226	0.225	0.224	0.221
1.3	0.214	0.218	0.222	0.225	0.226	0.226	0.225	0.225	0.223	0.220
1.4	0.212	0.216	0.220	0.223	0.224	0.225	0.224	0.225	0.222	0.219
1.5	0.211	0.214	0.218	0.222	0.224	0.224	0.223	0.223	0.221	0.218
1.6	0.209	0.213	0.216	0.221	0.222	0.223	0.223	0.222	0.220	0.217
1.7	0.207	0.211	0.215	0.219	0.221	0.222	0.221	0.220	0.219	0.215
1.8	0.206	0.210	0.214	0.217	0.219	0.220	0.219	0.219	0.217	0.213
1.9	0.204	0.208	0.212	0.216	0.216	0.219	0.218	0.218	0.215	0.211

Figure 8. BioASQ BM25 hyperparameter tuning.

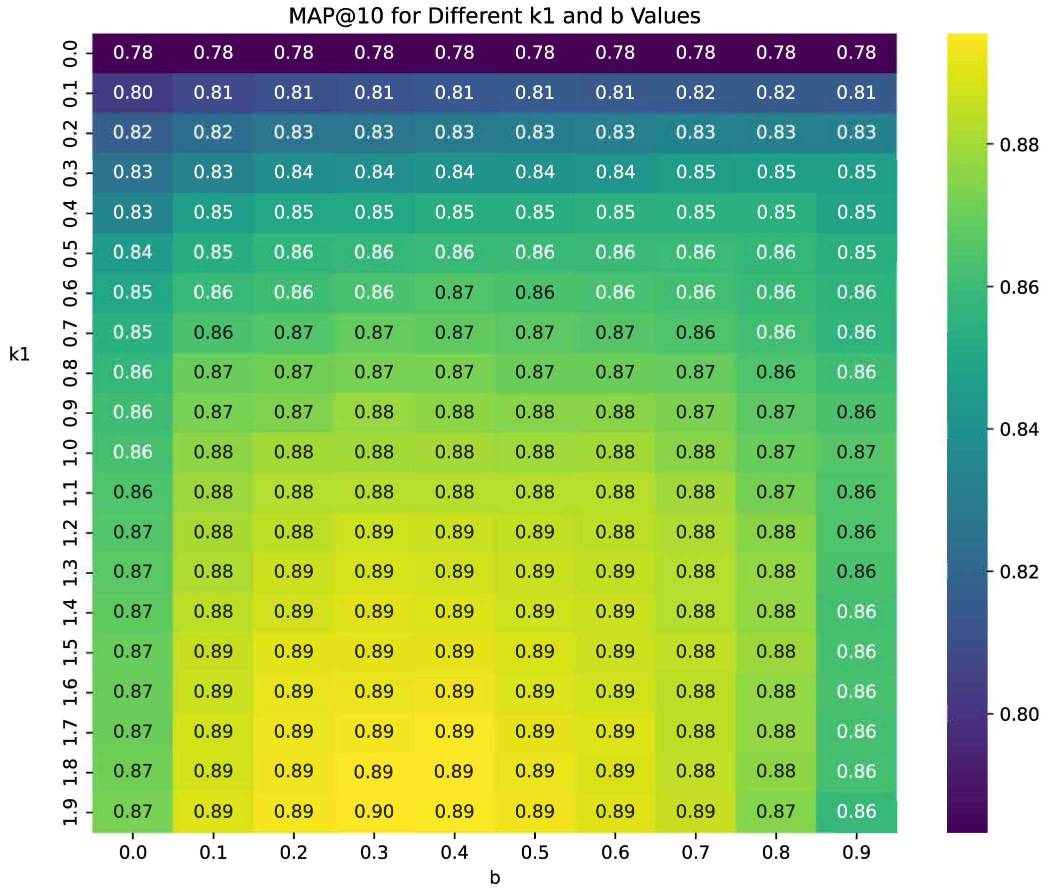


Figure 9. TREC-COVID BM25 hyperparameter tuning.

5.5. LLM Selection and k-Value Optimization

Following our investigation of BM25 hyperparameters, we conducted a series of experiments to optimize our two-stage retrieval system. This section focuses on determining the most effective open-source Large Language Model (LLM) and the optimal number of nearest neighbors (k-value) for our retrieval task.

5.5.1. Experimental Setup

We evaluated three open-source LLMs: BGE-reranker-gemma [53], LLaMA-3.1-8B-Instruct [59], and Qwen2-7B-Instruct [60]. To ensure a comprehensive assessment, we utilized test sets from both the BioASQ and TREC-COVID datasets. Performance was measured using MAP and NDCG metrics.

5.5.2. Results and Analysis

Figure 10 illustrates the impact of different k-values on the performance of various models after fine-tuning. In this figure, k represents the number of samples retrieved by the nearest neighbor algorithm for constructing training pairs. Specifically, for each query, the top-ranked sample is used as a positive example, while the remaining $k - 1$ samples serve as negative examples. As observed in the figure, the choice of k-value influences the fine-tuned model's performance, with different models exhibiting varying sensitivities to this parameter. The BGE-reranker model demonstrates better performance across all k-values for both MAP and NDCG metrics. Its performance remains consistently high when k-values are between 1 and 10, with a slight peak at $k = 5$. This suggests that for the BGE-reranker, a moderate number of negative samples provides an optimal balance between diverse training data and computational efficiency. Comparative analysis reveals that while LLaMA-3.1-8B-Instruct [59] and Qwen2-7B-Instruct [60] models also show improved performance with increasing k-values, they consistently underperform compared to the BGE-reranker.

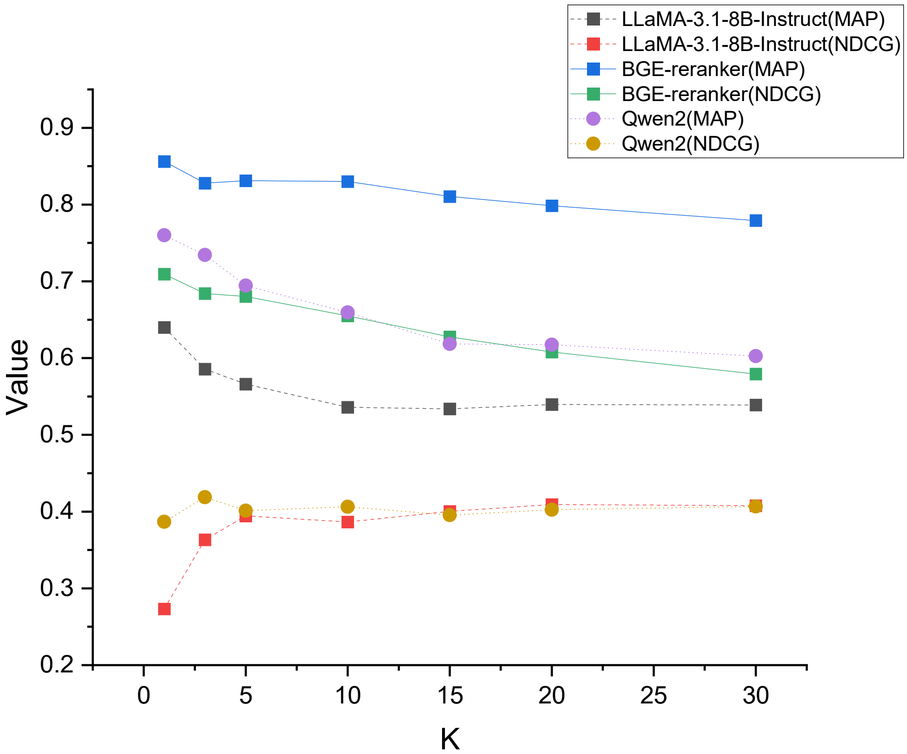


Figure 10. Performance comparison of different models across various k-values using MAP and NDCG metrics.

5.5.3. Optimal Configuration

Based on these experimental results, we selected the BGE-reranker as our core model for the retrieval system. The optimal configuration uses k = 5 for constructing training pairs, which means retrieving the 5 nearest neighbors for each query during the fine-tuning process. This configuration ensures one positive sample and four negative samples per query, striking an effective balance between training data diversity and model performance.

5.6. BioASQ Results Analysis

Table 4 presents a comparison of our BM25-LLMs method, with the best models from the BioASQ Challenge Task 11B across four test batches. The numbers in bold represent the best performance for each batch. To provide a more comprehensive analysis, we have conducted additional experiments to examine the performance across different question types: Factoid, List, Yes/No, and Summary. These detailed results are presented in Table 5.

In Test Batch-1, our system demonstrates competitive performance across various metrics. While bioinfo-0 leads in Mean Precision (0.3052) and F-Measure (0.3381), our system excels in Recall (0.6753) and MAP (0.6292), indicating strong retrieval consistency and coverage of relevant documents. Test Batch-2 shows improvement in our system's performance, with a Mean Precision of 0.3267, outperforming all other systems. Our system achieves a competitive Recall of 0.4942, closely approaching the performance of bioinfo-0 (0.4993). The F-Measure (0.3843) and MAP (0.4597) demonstrate balanced and quality retrieval performance. Similar trends continue in Test Batch-3, with our system showing strengths in Mean Precision (0.3879) and competitive performance in other metrics. For Test Batch-4, where only bioinfo-0 results are available for comparison, our system shows superior Mean Precision (0.4447 vs. 0.3327) but lower Recall (0.3443 vs. 0.4323).

Table 4. Comparison of the proposed BM25-LLMs method with the best models from BioASQ Challenge Task 11B.

Test Batch-1					
System	Mean Precision	Recall	F-Measure	MAP	GMAP
A&Q [61]	0.1427	0.4814	0.1733	0.2931	0.0115
A&Q2 [61]	0.1747	0.5378	0.2069	0.3995	0.0465
MindLab QA System [62]	0.2820	0.3369	0.2692	0.2631	0.0039
bioinfo-0 [63]	**0.3052**	0.6100	**0.3381**	0.5053	0.1014
Our system	0.2653	**0.6753**	0.2915	**0.6292**	**0.1735**
Test Batch-2					
System	Mean Precision	Recall	F-Measure	MAP	GMAP
A&Q [61]	0.1987	0.4428	0.2079	0.3494	0.0255
A&Q2 [61]	0.1747	0.3728	0.1761	0.2339	0.0111
MindLab QA System [62]	0.2283	0.2835	0.1876	0.1661	0.0063
bioinfo-0 [63]	0.2841	**0.4993**	0.2913	0.4244	**0.0858**
Our system	**0.3267**	0.4942	**0.3843**	**0.4597**	0.0609
Test Batch-3					
System	Mean Precision	Recall	F-Measure	MAP	GMAP
A&Q [61]	0.2144	0.4502	0.2120	0.3603	0.0439
A&Q2 [61]	0.2289	0.4574	0.2236	0.3775	0.0525
MindLab QA System [62]	0.1600	0.2100	0.1440	0.1312	0.0025
bioinfo-0 [63]	0.2823	**0.4794**	0.2808	0.3568	**0.0646**
Our system	**0.3879**	0.4521	**0.3405**	**0.4192**	0.0451
Test Batch-4					
System	Mean Precision	Recall	F-Measure	MAP	GMAP
A&Q [61]	-	-	-	-	-
A&Q2 [61]	-	-	-	-	-
MindLab QA System [62]	-	-	-	-	-
bioinfo-0 [63]	0.3327	**0.4323**	0.3066	**0.3751**	**0.0595**
Our system	**0.4447**	0.3443	**0.3092**	0.3623	0.0282

Note: The dashes ("-") in the table indicate that the corresponding systems were not listed in the rankings for this batch. Bold values indicate the best performance for each metric.

Table 5. Detailed results for different question types in BioASQ.

Test Batch-1				
Type	Precision	Recall	F-Measure	MAP
Factoid	0.2105	0.6451	0.2415	0.5491
List	0.2667	0.5464	0.3050	0.3470
Yes/No	0.2458	0.6578	0.2590	0.5248
Summary	0.1850	0.6208	0.2329	0.4658
Test Batch-2				
Type	Precision	Recall	F-Measure	MAP
Factoid	0.2955	0.5426	0.3108	0.4383
List	0.3250	0.3723	0.2226	0.2831
Yes/No	0.2708	0.6397	0.2955	0.4875
Summary	0.2706	0.6110	0.3025	0.4558
Test Batch-3				
Type	Precision	Recall	F-Measure	MAP
Factoid	0.2154	0.5137	0.2166	0.3093
List	0.3833	0.3985	0.3150	0.3294
Yes/No	0.2000	0.5706	0.2188	0.4455
Summary	0.2545	0.5378	0.2661	0.3386
Test Batch-4				
Type	Precision	Recall	F-Measure	MAP
Factoid	0.1774	0.5288	0.2287	0.3930
List	0.3083	0.3665	0.2631	0.3027
Yes/No	0.2643	0.4406	0.2173	0.3812
Summary	0.3524	0.3284	0.2796	0.2494

Examining the performance across different question types reveals interesting patterns. Factoid questions show relatively consistent performance across batches, with Recall ranging from 0.5137 to 0.6451, while Precision shows more variation (0.1774 to 0.2955). List questions demonstrate the highest Precision among all types, particularly in Batch-3

(0.3833), although Recall is generally lower compared to other question types. Yes/No questions exhibit considerable performance variation across batches, with Precision (0.2708) and Recall (0.6397) in Batch-2. Summary questions demonstrate more stable performance across batches, generally showing a good balance between Precision and Recall.

5.7. Results on TREC-COVID Dataset

To evaluate our proposed system, we conducted experiments on the TREC-COVID dataset [30]. Due to the unavailability of historical records for participating systems in previous years, we implemented a set of strong baselines for comparison. These baselines represent a range of traditional and state-of-the-art information retrieval methods, providing a comprehensive evaluation framework for our proposed approach.

The baseline systems we employed include:

- BM25 [20]: A classical probabilistic retrieval model that serves as a fundamental sanity check by directly using the ranking results from the first-stage retrieval.
- Sentence-BERT [64]: A modification of the BERT model that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings.
- DPR [22]: A dense retrieval method that uses neural networks to encode queries and passages into a low-dimensional space.
- docT5query [43]: A document expansion technique that uses a sequence-to-sequence model to predict queries that a document might be relevant to.
- Manual Prompt: To benchmark against cutting-edge manual prompting techniques, we incorporated RankGPT [65], the current state-of-the-art approach in this domain.
- CoT [66]: This approach extends the manual prompt by appending the phrase “Let’s think step by step” to encourage more structured and detailed reasoning.
- ListT5 [67]: A re-ranking methodology leveraging the Fusion-in-Decoder architecture, which processes multiple candidate passages concurrently during both training and inference phases.
- COCO-DR [68]: A zero-shot dense retrieval method designed to enhance generalization by addressing distribution shifts between training and target scenarios.
- RaMDA [69]: A novel model addressing the domain adaptation for dense retrievers by synthesizing domain-specific data through pseudo queries.

We employed NDCG@10 as our primary evaluation metric. This measure was chosen for its ability to capture both the relevance and ranking quality of the retrieved documents, with a particular focus on the top 10 results. Table 6 presents a comprehensive comparison of our proposed system against various models and approaches from the TREC-COVID dataset. The values in bold typeface indicate the highest NDCG@10 score achieved across all systems.

Table 6. Comparison of NDCG@10 scores for various document retrieval systems.

System	NDCG@10
Our system	0.8326
BM25	0.4812
Sentence-BERT	0.3334
DPR	0.6420
docT5query	0.7420
GPT-3.5-Manual	0.7667
GPT-3.5-CoT	0.8213
LLaMA3-Manual	0.7746
LLaMA3-CoT	0.7454
Qwen2-Manual	0.8145
Qwen2-CoT	0.7972
ListT5-base	0.7830
ListT5-3B	**0.8440**
PE-Rank	0.7772
COCO-DR	0.7890
RaMDA	0.8143

Note: Bold values indicate the best performance.

6. Conclusions

In the context of biomedical retrieval, enhancing the efficacy of traditional retrieval models for case retrieval remains a critical challenge. To address this, we propose a BM25-LLMs biomedical question retrieval system that integrates the traditional BM25 [20] algorithm with LLMs. This approach leverages the strengths of both techniques to improve retrieval accuracy and efficiency. Our comparative experiments indicate that the proposed BM25-LLMs biomedical retrieval system demonstrates competitive performance when evaluated against other current models in the field. The observed improvements in retrieval effectiveness suggest that combining traditional retrieval techniques with large language models may offer certain benefits in biomedical information retrieval tasks. These results contribute to the ongoing research on hybrid approaches aimed at enhancing retrieval capabilities within the biomedical domain.

Author Contributions: H.Y., conceptualization, methodology, data curation, validation, writing—review and editing, funding acquisition, supervision; S.L., methodology, formal analysis, validation, software, writing—original draft; T.G., resources, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: Key Scientific Research Project of Higher Education Institutions in Henan Province. Grant No. 24A520060.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets analyzed for this study are publicly accessible.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average Precision
BERT	Bidirectional Encoder Representations from Transformers
BM25	Best Match 25
CoT	Chain of Thought
DCG	Discounted Cumulative Gain
DESM	Dual Embedding Space Model
DPR	Dense Passage Retrieval
DSSM	Deep Structured Semantic Models
EBAE	Embedding-Based Auto-Encoding
EBAR	Embedding-Based Auto-Regression
FN	False Negatives
FP	False Positives
GMAP	Geometric Mean Average Precision
IDCG	Ideal Discounted Cumulative Gain
IR	Information Retrieval
LoRA	Low-Rank Adaptation
MAP	Mean Average Precision
NDCG	Normalized Discounted Cumulative Gain
QA	Question Answering
RDF	Resource Description Framework
TREC-COVID	Text REtrieval Conference COVID
TP	True Positives

References

1. Qiu, M.; Li, F.L.; Wang, S.; Gao, X.; Chen, Y.; Zhao, W.; Chen, H.; Huang, J.; Chu, W. Alime chat: A sequence to sequence and rerank based chatbot engine. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 498–503.

2. Yan, Z.; Duan, N.; Bao, J.; Chen, P.; Zhou, M.; Li, Z.; Zhou, J. Docchat: An information retrieval approach for chatbot engines using unstructured documents. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 516–525.
3. Amato, F.; Marrone, S.; Moscato, V.; Piantadosi, G.; Picariello, A.; Sansone, C. Chatbots Meet eHealth: Automatizing Healthcare. In Proceedings of the WAIAH@ AI* IA, Bari, Italy, 14 November 2017; pp. 40–49.
4. Ram, A.; Prasad, R.; Khatri, C.; Venkatesh, A.; Gabriel, R.; Liu, Q.; Nunn, J.; Hedayatnia, B.; Cheng, M.; Nagar, A.; et al. Conversational ai: The science behind the alexa prize. *arXiv* **2018**, arXiv:1801.03604.
5. Kadam, A.D.; Joshi, S.D.; Shinde, S.V.; Medhane, S.P. Notice of Removal: Question Answering Search engine short review and road-map to future QA Search Engine. In Proceedings of the 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, India, 24–25 January 2015; pp. 1–8. [CrossRef]
6. Yaghoubzadeh, R.; Kopp, S. Toward a virtual assistant for vulnerable users: Designing careful interaction. In Proceedings of the 1st Workshop on Speech and Multimodal Interaction in Assistive Environments, Jeju Island, Republic of Korea, 12 July 2012; pp. 13–17.
7. Austerjost, J.; Porr, M.; Riedel, N.; Geier, D.; Becker, T.; Scheper, T.; Marquard, D.; Lindner, P.; Beutel, S. Introducing a virtual assistant to the lab: A voice user interface for the intuitive control of laboratory instruments. *SLAS TECHNOLOGY Transl. Life Sci. Innov.* **2018**, *23*, 476–482. [CrossRef] [PubMed]
8. Bradley, N.C.; Fritz, T.; Holmes, R. Context-aware conversational developer assistants. In Proceedings of the 40th International Conference on Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018; pp. 993–1003.
9. Vicedo, J.L.; Ferrández, A. Importance of pronominal anaphora resolution in question answering systems. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, 1–8 October 2000; pp. 555–562.
10. Mollá, D.; Van Zaanen, M.; Smith, D. Named entity recognition for question answering. In Proceedings of the Australasian Language Technology Association Workshop. Australasian Language Technology Association, Canberra, Australia, 1–2 December 2006; pp. 51–58.
11. Mao, Y.; Wei, C.H.; Lu, Z. NCBI at the 2014 BioASQ Challenge Task: Large-scale Biomedical Semantic Indexing and Question Answering. In Proceedings of the CLEF (Working Notes), Sheffield, UK, 15–18 September 2014; pp. 1319–1327.
12. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
13. Malik, N.; Sharan, A.; Biswas, P. Domain knowledge enriched framework for restricted domain question answering system. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research, Madurai, India, 26–28 December 2013; IEEE: New York, NY, USA, 2013; pp. 1–7.
14. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv* **2016**, arXiv:1606.05250.
15. Kočiskỳ, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K.M.; Melis, G.; Grefenstette, E. The narrativeqa reading comprehension challenge. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 317–328. [CrossRef]
16. Weissenborn, D.; Tsatsaronis, G.; Schroeder, M. Answering factoid questions in the biomedical domain. *BioASQ@ CLEF* **2013**, *1094*. Available online: https://ceur-ws.org/Vol-1094/bioasq2013_submission_5.pdf (accessed on 15 August 2024).
17. Yang, H.; Gonçalves, T. Field features: The impact in learning to rank approaches. *Appl. Soft Comput.* **2023**, *138*, 110183. [CrossRef]
18. Antonio, M.; Soares, C.; Parreiras, F. A literature review on question answering techniques, paradigms and systems. *J. King Saud Univ. Comput. Inf. Sci.* **2018**, *8*, 1–12.
19. Russell-Rose, T.; Chamberlain, J. Expert search strategies: The information retrieval practices of healthcare information professionals. *JMIR Med. Inform.* **2017**, *5*, e7680. [CrossRef]
20. Robertson, S.; Zaragoza, H. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* **2009**, *3*, 333–389. [CrossRef]
21. Robertson, S.E.; Walker, S. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In Proceedings of the SIGIR ’94, Dublin, Ireland, 3–6 July 1994; Croft, B.W., van Rijsbergen, C.J., Eds.; Springer: London, UK, 1994; pp. 232–241.
22. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.T. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6769–6781. [CrossRef]
23. Berger, A.; Caruana, R.; Cohn, D.; Freitag, D.; Mittal, V. Bridging the lexical chasm: Statistical approaches to answer-finding. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 24–28 July 2000; SIGIR ’00; pp. 192–199. [CrossRef]
24. Fang, H.; Zhai, C. Semantic term matching in axiomatic approaches to information retrieval. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 6 August 2006; SIGIR ’06; pp. 115–122. [CrossRef]
25. Humeau, S.; Shuster, K.; Lachaux, M.A.; Weston, J. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. *arXiv* **2019**, arXiv:1905.01969.
26. Soldaini, L.; Moschitti, A. The Cascade Transformer: An Application for Efficient Answer Sentence Selection. *arXiv* **2020**, arXiv:2005.02534.

27. Guo, J.; Fan, Y.; Ai, Q.; Croft, W.B. A Deep Relevance Matching Model for Ad-hoc Retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, New York, NY, USA, 24–28 October 2016; CIKM '16; pp. 55–64. [CrossRef]
28. Ma, X.; Sun, K.; Pradeep, R.; Li, M.; Lin, J. Another Look at DPR: Reproduction of Training and Replication of Retrieval. In Proceedings of the Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, 10–14 April 2022; Part I; Springer: Berlin/Heidelberg, Germany, 2022; pp. 613–626. [CrossRef]
29. Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M.R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinform.* **2015**, *16*, 138. [CrossRef]
30. Voorhees, E.; Alam, T.; Bedrick, S.; Demner-Fushman, D.; Hersh, W.R.; Lo, K.; Roberts, K.; Soboroff, I.; Wang, L.L. TREC-COVID: Constructing a pandemic information retrieval test collection. In Proceedings of the ACM SIGIR Forum, New York, NY, USA, 19 February 2021; Volume 54, pp. 1–12.
31. Lee, M.; Cimino, J.; Zhu, H.R.; Sable, C.; Shanker, V.; Ely, J.; Yu, H. Beyond information retrieval—Medical question answering. *AMIA Annu. Symp. Proc.* **2006**, *2006*, 469–473.
32. Cruchet, S.; Gaudinat, A.; Rindflesch, T.; Boyer, C. What about trust in the question answering world. In Proceedings of the AMIA 2009 Annual Symposium, San Francisco, CA, USA, 14–18 November 2009.
33. Gobeill, J.; Patsche, E.; Theodoro, D.; Veuthey, A.L.; Lovis, C.; Ruch, P. Question answering for biology and medicine. In Proceedings of the 2009 9th International Conference on Information Technology and Applications in Biomedicine, Larnaka, Cyprus, 4–7 November 2009; pp. 1–5.
34. Cao, Y.; Liu, F.; Simpson, P.; Antieau, L.; Bennett, A.; Cimino, J.J.; Ely, J.; Yu, H. AskHERMES: An online question answering system for complex clinical questions. *J. Biomed. Inform.* **2011**, *44*, 277–288. [CrossRef]
35. Hristovski, D.; Dinevski, D.; Kastrin, A.; Rindflesch, T.C. Biomedical question answering using semantic relations. *BMC Bioinform.* **2015**, *16*, 6. [CrossRef]
36. Liu, T.Y. Learning to rank for information retrieval. *Found. Trends® Inf. Retr.* **2009**, *3*, 225–331. [CrossRef]
37. Nogueira, R.; Yang, W.; Cho, K.; Lin, J. Multi-stage document ranking with BERT. *arXiv* **2019**, arXiv:1910.14424.
38. Pradeep, R.; Liu, Y.; Zhang, X.; Li, Y.; Yates, A.; Lin, J. Squeezing water from a stone: A bag of tricks for further improving cross-encoder effectiveness for reranking. In Proceedings of the European Conference on Information Retrieval, Stavanger, Norway, 10–14 April 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 655–670.
39. Zhou, Y.; Shen, T.; Geng, X.; Tao, C.; Xu, C.; Long, G.; Jiao, B.; Jiang, D. Towards robust ranker for text retrieval. *arXiv* **2022**, arXiv:2206.08063.
40. Ponte, J.M.; Croft, W.B. A language modeling approach to information retrieval. In Proceedings of the ACM SIGIR Forum, New York, NY, USA, 2 August 2017; Volume 51, pp. 202–208.
41. Zheng, G.; Callan, J. Learning to reweight terms with distributed representations. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 575–584.
42. Dai, Z.; Callan, J. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv* **2019**, arXiv:1910.10687.
43. Nogueira, R.; Lin, J.; Epistemic, A. From doc2query to docTTTTTquery. *Online Prepr.* **2019**, *6*, 2. Available online: https://cs.uwaterloo.ca/~jimmylin/publications/Nogueira_Lin_2019_docTTTTTquery.pdf (accessed on 15 August 2024).
44. Bai, Y.; Li, X.; Wang, G.; Zhang, C.; Shang, L.; Xu, J.; Wang, Z.; Wang, F.; Liu, Q. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv* **2020**, arXiv:2010.00768.
45. Mallia, A.; Khattab, O.; Suel, T.; Tonellotto, N. Learning passage impacts for inverted indexes. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 1723–1727.
46. Jang, K.R.; Kang, J.; Hong, G.; Myaeng, S.H.; Park, J.; Yoon, T.; Seo, H. Ultra-high dimensional sparse representations with binarization for efficient text retrieval. *arXiv* **2021**, arXiv:2104.07198.
47. Chen, Q.; Wang, H.; Li, M.; Ren, G.; Li, S.; Zhu, J.; Li, J.; Liu, C.; Zhang, L.; Wang, J. SPTAG: A Library for Fast Approximate Nearest Neighbor Search. GitHub. 2018. Available online: https://github.com/microsoft/SPTAG (accessed on 15 August 2024).
48. Mitra, B.; Nalisnick, E.; Craswell, N.; Caruana, R. A dual embedding space model for document ranking. *arXiv* **2016**, arXiv:1602.01137.
49. Gao, L.; Dai, Z.; Callan, J. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv* **2021**, arXiv:2104.07186.
50. Huang, P.S.; He, X.; Gao, J.; Deng, L.; Acero, A.; Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international Conference on Information & Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013; pp. 2333–2338.
51. Hu, B.; Lu, Z.; Li, H.; Chen, Q. Convolutional neural network architectures for matching natural language sentences. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2042–2050.
52. Li, C.; Liu, Z.; Xiao, S.; Shao, Y. Making Large Language Models A Better Foundation For Dense Retrieval. *arXiv* **2023**, arXiv:2312.15503.
53. Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; Liu, Z. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings through Self-Knowledge Distillation. *arXiv* **2024**, arXiv:2402.03216.

54. Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M.S.; Love, J.; et al. Gemma: Open models based on gemini research and technology. *arXiv* **2024**, arXiv:2403.08295.
55. Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.E.; Lomeli, M.; Hosseini, L.; Jégou, H. The Faiss library. *arXiv* **2024**, arXiv:2401.08281.
56. Doms, A.; Schroeder, M. GoPubMed: Exploring PubMed with the gene ontology. *Nucleic Acids Res.* **2005**, *33*, W783–W786. [CrossRef]
57. Wang, L.L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; Yang, J.; Burdick, D.; Eide, D.; Funk, K.; Katsis, Y.; Kinney, R.M.; et al. CORD-19: The COVID-19 Open Research Dataset. In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Online, 5–10 July 2020.
58. Lin, J.; Ma, X.; Lin, S.C.; Yang, J.H.; Pradeep, R.; Nogueira, R. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 11–15 July 2021; pp. 2356–2362.
59. Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. The Llama 3 Herd of Models. *arXiv* **2024**, arXiv:2407.21783.
60. Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Fan, Z.; et al. Qwen2 Technical Report. a*rXiv* **2024**, arXiv:2407.10671.
61. Shin, A.; Jin, Q.; Lu, Z. Multi-stage literature retrieval system trained by PubMed search logs for biomedical question answering. In Proceedings of the Conference and Labs of the Evaluation Forum (CLEF), Thessaloniki, Greece, 18–21 September 2023.
62. Rosso-Mateus, A.; Muñoz-Serna, L.A.; Montes-y Gómez, M.; González, F.A. Deep Metric Learning for Effective Passage Retrieval in the BioASQ Challenge. In Proceedings of the CLEF 2023: Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, 18–21 September 2023.
63. Almeida, T.; Jonker, R.A.A.; Poudel, R.; Silva, J.M.; Matos, S. BIT.UA at BioASQ 11B: Two-Stage IR with Synthetic Training and Zero-Shot Answer Generation. In Proceedings of the Conference and Labs of the Evaluation Forum, Bologna, Italy, 5–8 September 2022.
64. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
65. Sun, W.; Yan, L.; Ma, X.; Ren, P.; Yin, D.; Ren, Z. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *arXiv* **2023**, arXiv:2304.09542.
66. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
67. Yoon, S.; Choi, E.; Kim, J.; Yun, H.; Kim, Y.; won Hwang, S. ListT5: Listwise Reranking with Fusion-in-Decoder Improves Zero-shot Retrieval. *arXiv* **2024**, arXiv:2402.15838.
68. Yu, Y.; Xiong, C.; Sun, S.; Zhang, C.; Overwijk, A. Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning. *arXiv* **2022**, arXiv:2210.15212.
69. Kim, J.; Kim, M.; Park, J.; Hwang, S.w. Relevance-assisted Generation for Robust Zero-shot Retrieval. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, Singapore, 6–10 December 2023; pp. 723–731.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.