BIT.UA at BioASQ 11B: Two-Stage IR with Synthetic Training and Zero-Shot Answer Generation

Tiago Almeida[1], Richard A. A. Jonker[1], Roshan Poudel[1], Jorge M. Silva[1] and Sérgio Matos[1,*,†]

[1]IEETA/DETI, LASI, University of Aveiro, Portugal

Abstract

This paper presents the efforts of the Biomedical Informatics and Technologies (BIT) group at the University of Aveiro in the eleventh edition of the BioASQ challenge. This paper presents our efforts in the eleventh edition of the BioASQ challenge. We addressed Task B in its two phases: document retrieval (phase A) and question answering (phase B). In phase A, we utilized a sparse retrieval method for initial document retrieval, implemented using Anserini, followed by a re-ranking step using transformer models, including monoT5 and PubMedBERT. Phase B featured the application of large language models (LLMs) to generate answers to questions based on a relevant article, with models such as Alpaca-LoRA, OA-Pythia, and OA-LLaMA. We also explored a variety of prompts and question types, as well as different generation strategies to optimize our system's performance. Our systems, in phase A, achieved competitive results scoring at the top and close to the top for all the batches, and achieving the best results in terms of F1 for all the batches. Regarding the phase B, our systems underperformed according to the automatic measures. Code to reproduce our submissions is available at https://github.com/ieeta-pt/BioASQ_11B.

Keywords

Information Retrieval, Dense Retrieval, Language model, Answer Generation

1. Introduction

The realm of biomedical literature has been experiencing an exponential increase, predominantly driven by the rise in open-access and peer-reviewed publications. This rapid expansion results in an information overload, posing a significant challenge to researchers, physicians, and other healthcare practitioners [1]. As delineated by Klerings et al. [1], the primary concern stems not from the abundance of information but the scarcity of sophisticated information retrieval systems proficient in managing this growing body of literature. To mitigate this, the BioASQ challenge is a yearly competition that stimulates the creation of intelligent retrieval systems. In its eleventh year, the BioASQ challenge [2, 3] comprises several tasks targeting unique facets of information retrieval and text mining within the biomedical domain.

Task B and the Synergy task emphasises information retrieval and question-answering. Task B bifurcates into phases A and B. Phase A involves identifying relevant documents or snippets

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

tiagomeloalmeida@ua.pt (T. Almeida); richard.jonker@ua.pt (R. A. A. Jonker); proshan@ua.pt (R. Poudel); jorge.miguel.ferreira.silva@ua.pt (J. M. Silva); aleixomatos@ua.pt (S. Matos)
https://t-almeida.github.io/online-cv/ (T. Almeida); https://jorgeMFS.com/ (J. M. Silva)
0000-0002-4258-3350 (T. Almeida); 0000-0002-3806-6940 (R. A. A. Jonker); 0000-0001-6450-023X (R. Poudel); 0000-0002-6331-6091 (J. M. Silva); 0000-0003-1941-3983 (S. Matos)

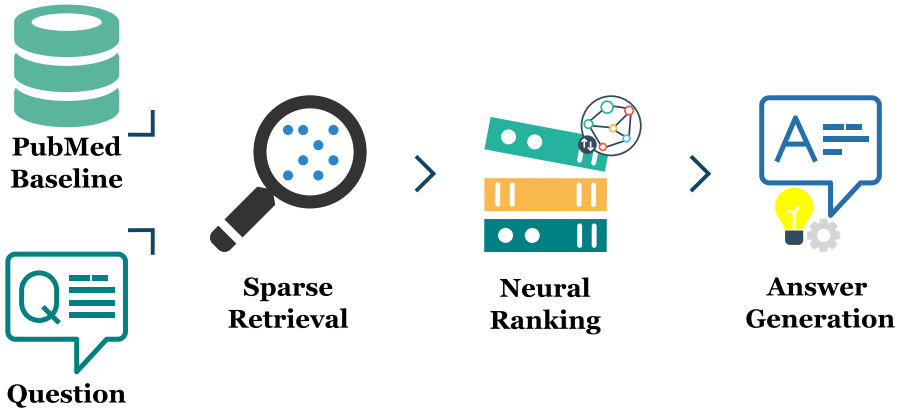

Figure 1: High level overview of the entire system pipeline.

that answer a biomedical question, while phase B addresses the extraction and generation of responses. These tasks collectively aim at advancing systems that provide evidence or answers to open-ended biomedical queries. In contrast, the Synergy task seeks to resolve open-ended questions about COVID-19 by leveraging IR and QA systems.

This paper describes our participation in Task B phase A and ideal answer in phase B of the BioASQ challenge. During phase A, we utilized the traditional BM25 [4] for base document retrieval, followed by document re-ranking executed via a variety of transformer models, including monoT5 [5] and PubMedBERT [6]. These models were fine-tuned on prior years' data, and synthetic data generation was employed to mitigate the constraints of a small dataset size. During phase B, we adopted a naive unsupervised approach where language models were prompted to generate answers to a question provided with a article as context. The approach involved exploring various models and prompts along with differing context selections. Figure 1 shows an illustration of an end-to-end pipeline for information retrieval and answering system.

Following this introduction, Section 2 explains the related work. Section 3 is the methodological section, where we explore the used datasets and corpora and thoroughly illustrates the employed methodologies. Section 4 shows our results and section 5 discusses them. The paper concludes in Section 6, summarising the key findings of our participation, with a brief discussion of future work in Section 7.

2. Related Work

The BioASQ challenge has consistently catalyzed significant advancements in biomedical information retrieval and question-answering. Task B, in particular, encapsulates the essence of these complex processes, focusing on two fundamental fields: Information Retrieval (IR) and Question Answering (QA).

Fundamentally, IR (phase A) aims to identify and retrieve relevant documents or snippets that align with a posed biomedical question, thereby addressing the issue of locating pertinent information within the vast corpus of biomedical literature [2]. QA (phase B), on the other hand, is concerned with extracting and generating comprehensive answers from the retrieved

information. This intricate process requires understanding the question at hand and determining the most suitable answer by leveraging the context provided by the retrieved documents.

In the latest competition, the state-of-the-art performances were achieved by systems that utilized a two-step process: an initial sparse retrieval system followed by a Transformer-based re-ranking model [7]. This approach was not unique to a single submission but was rather a common thread among various entries. Our previous work Almeida et al. [8] also employed a similar pipeline, that used BM25 as first-stage, and in the second stage, employing powerful models such as PubMedBERT [6] and UPWM [9]. These models have shown remarkable proficiency in interpreting intricate biomedical queries and matching it to a relevant article.

2.1. Information Retrieval

Information Retrieval (IR) involves identifying relevant documents that match a specific query. IR can be broadly categorized into sparse retrieval and dense retrieval. Sparse retrieval, usually associated with more traditional approaches, involves converting text into an inverted index to enable fast searching. An inverted index stores a mapping of terms to documents. Sparse retrieval has the advantage that it is fast and explainable. The simpler approach of sparse retrieval includes Bag-of-Words and term frequency-inverse document frequency (tf-idf). There are also sparse retrieval techniques that are enhanced by transformer-based models such as DeepCT [10] and HDCT [11] which produces contextualized term weights that can be stored in traditional inverted indexes. Nevertheless, one of the most relevant and well-known algorithms used in sparse retrieval is BM25 [4].

$$\text{BM25} = \sum \left(\frac{\text{tf}(t, D) \cdot (k_1 + 1)}{\text{tf}(t, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \cdot \ln \left(\frac{N - \text{df}(t) + 0.5}{\text{df}(t) + 0.5} \right) \right).$$

Where $\text{tf}(t, D)$ represents the term frequency of term (t) in the document (D), $|D|$ represents the length of the documents, avgdl is the average length of a document in the collection, N is the number of documents in the collection and $\text{tf}(t)$ is the number of documents containing term t. k_1 and b are hyperparameters that can be tuned.

On the other hand, a more recent approach called dense retrieval has emerged, utilizing transformer models to convert both documents and queries into the same dimensional space [12]. In this approach, the query is transformed into a vector representation by the dense retrieval model. The search process involves comparing the similarity of the query vector against all the document vectors that have been previously encoded. Prominent approaches in this domain include DPR [13] and ANCE [14], which employ transformer-based models to learn a joint dimensional space for projecting queries and documents in a meaningful way. This enables queries to be closer in dimensional space to their relevant documents. To facilitate efficient execution of this type of search, libraries like Facebook's FAISS [15] offer a comprehensive framework designed specifically for this purpose.

Both the dense retrieval and sparse retrieval techniques can be broadly classified as representation-based approaches. In this approach, the document and query are encoded separately, and the search is performed based on either similarity measures (dense retrieval) or cumulative scores (sparse retrieval). In contrast, interaction-based approaches jointly score

the query and document, allowing for the extraction of more intricate matching patterns and potentially improving retrieval results. However, due to the need to score the query against every document in the collection, interaction-based approaches are not practical for searching the entire document corpus. Therefore, representation-based approaches are commonly adopted as first-stage retrieval techniques to reduce the search space. Subsequently, more powerful interaction-based techniques can be employed to further refine the ranking order, a process known as re-ranking in the literature. These models are typically trained using pointwise and pairwise techniques [16]. Pointwise learning involves assigning a score to each document, and the ranking is then performed by sorting these scores. On the other hand, pairwise learning involves comparing pairs of documents and enforcing a margin between positive and negative document pairs, leading to a more discriminative learning process.

2.2. Question Answering

Question Answering(QA) aims to provide accurate and relevant answers to various questions. QA tasks can be generally divided into two main categories:

- **Extractive QA** involves identifying and extracting an answer from the given context.
- **Generative QA** requires the model to generate an answer freely, sometimes requiring a context.

Generative QA can further be divided into open and closed generative QA. In open generative QA, the text is generated using a context provided. This is not to be confused with open-domain QA. Closed generative QA has no context, and the model entirely generates the answer.

More recent generative QA approaches leverage large language models (LLMs) for zero-shot answer generation. In this setup, the model is provided with a query containing the context and asked to generate an answer. This approach is relatively new in the literature. GPT-3 [17] is a powerful autoregressive language model that uses deep learning to produce human-like text. It has 175 billion parameters and has been applied successfully in zero-shot tasks that require a deep understanding of context, making it a suitable choice for generative QA tasks.

Other recent LLMs have surfaced, such as LLaMA [18], Alpaca [19] and Pythia [20]. LLaMA is a foundation LLM that is based on various transformer-based architectures, namely GPT-3 [17], PaLM [21], and GPTNeo [22]. Alpaca is a LLM based on LLaMA that was fine-tuned on the text generated by OpenAi's GPT-3.5. Using this technique of knowledge distillation, LLMs can be made much smaller without sacrificing too much performance. Alpaca-LoRA[1] employs an approach known as Low-Rank Adaptation [23], which keeps the pre-trained model weights constant and introduces trainable rank decomposition matrices at each layer of the Transformer architecture. This significantly reduces the number of trainable parameters for downstream tasks. Pythia is a library for Transformers, providing various pre-trained models, which are also GPT based. OpenAssistant [24] fine-tuned Pythia and LLaMA models on human-labelled datasets to boost the models' performance and create an open-source competitor to ChatGPT.

[1] https://github.com/tloen/alpaca-lora

3. Methodology

The methodology section commences with a comprehensive overview of the corpora and the dataset used in each task. Subsequently, it details the methods employed for each task we participated in.

3.1. Corpora and Dataset

For Task B, we were provided with a dataset containing data from the first ten editions of the challenge. The dataset included 4719 questions, categorized as 1417 'factoid', 1271 'yesno', 1130 'summaries', and 901 'lists'. Each question was accompanied by its relevant documents, snippets, concepts, RDF triples, and exact and ideal answers. To construct our corpus, we utilized the PubMed annual baseline document collections spanning from 2013 to 2023. This corpus consisted of the abstracts and titles of all documents. The most recent PubMed baseline collection (2023) contains approximately 35 million documents. However, we encountered a challenge due to the dynamic nature of the documents. Each year, documents are updated or removed, which means that the relevant documents for a question in the first edition may no longer be present in the document collection for the current edition. This posed a problem when relying solely on the latest baseline collection to extract the title and abstract for accurate querying. To address this issue, we augmented each question with the year it appeared in, enabling us to query the relevant documents more precisely.

Additionally, we encountered some documents that were missing titles, abstracts, or both. This could be due to licensing or linguistic issues. We addressed this by removing these incomplete documents from the collection. Afterwards, we created sparse Anserini [25] indexes for each year. Having yearly indexes proved advantageous as it allowed us to search for relevant documents specific to the year in which a question appeared. This approach enhanced the accuracy of retrieving pertinent information for each question.

Regarding the question dataset, there were cases where questions were repeated or were very similar although having a different set of relevant documents. Due to this fact, we decided to merge similar questions by merging the set of relevant articles to enrich the training data. To accomplish this, we leveraged the pre-trained SimCSE [26] model to compute the similarity between questions. Then, questions with a similarity score above 99% were automatically merged, while questions with a similarity score between 90% and 99% were manually reviewed. Another additional step was to remove the questions before BioASQ 4. During these years of the challenge, the systems were able to use the full-text article from PubMed Central (PMC) to make judgments. This will lead to situations where the model does not have the necessary content to make a correct prediction for these document pairs. At the end of this process, the number of resulting question were 3465 (-30%) totalling, 25781 question-documents positive pairs. In order to build a training dataset for neural relevance models, we need to also gather negative question-document pairs, such that the model can learn how to correctly score relevant and irrelevant documents. To accomplish this, we performed random sampling over the list of documents provided by the BM25 that were not positives for a given question. This should result in a list of strong negative documents.

3.1.1. Synthetic Question Generation

Data quality and quantity are crucial for developing strong, effective models in deep learning for information retrieval and relevance determination. In the previous section, we describe our pre-processing steps to increase the quality of the gold standard data. However, we are still missing in terms of data quantity. We propose generating questions by transformer-based language models to create a synthetic dataset that can be used to pre-train the relevance models to first learn basic retrieval patterns.

To synthetically generate a question for a given article, we fed an engineered prompt that tries to condition a language model to generate a question based on the information contained in the article. More formally, we empirically built the prompt, $p = \{p_1, ..., p_M\}$, such that a language model would maximize the probability of a question, $y = \{y_1, ..., y_N\}$, being sampled according to Equation 1.

$$y \sim \prod_{i=1}^{N} P(y_i | p_1, ..., p_M, y_1, ..., y_{i-1}) \tag{1}$$

In this work, we mainly used zero-shot question generation since we did not explore training the language models to generate questions based on the BioASQ data. To further guide the language model into generating useful questions, we also included a question starting word as part of the prompt, such that the model will be forced to pick the following words conditioned on that starting word. Some examples of words that start a question are {What, Which, Is, List, Are, Does}[2], Prompt 1 shows the prompt that we adopted for generating a question in a zero-shot fashion with OA-pythia 12B model.

```
<|prompter|>Given the following context
\"{article}\", generate a question that can be
answered by the information provided in the
context: <|endoftext|><|assistant|>What
```

Prompt 1: Example of the last prompt used to generate synthetic questions with OA-pythia model

Regarding the language models that we used, we tried with small language models like, GPT-Neo-125M [22] and also with the larger ones such as OA-pythia-12B [20, 24] model. The synthetic dataset contained 79855 questions that were generated from 15971 randomly sampled articles.

3.2. Phase A

Our approach for the phase A of the challenge involved the development of a two-stage retrieval pipeline designed to handle the large volume of biomedical literature with efficiency. Figure 1 presents the overview of our two-stage retrieval pipeline.

[2]These words follow the distribution of the starting words that appear in the BioASQ dataset.

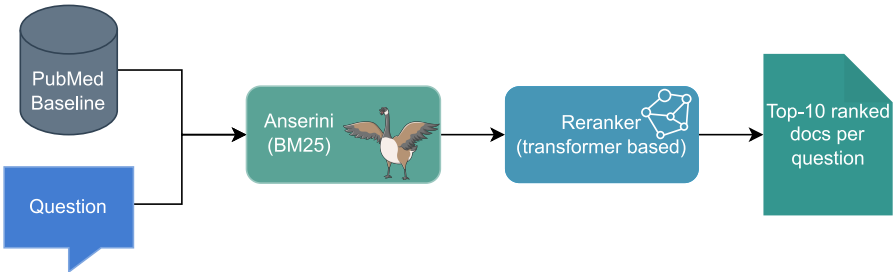

Figure 2: Overview of the proposed two-stage retrieval pipeline for participating in phase A.

At first, we utilized a sparse retrieval method. To accomplish this, we constructed an inverted index, a commonly used data structure in information retrieval that maps terms to the documents that contain them, using Anserini, a powerful retrieval toolkit built on Lucene [27]. For compatibility with our Python-based pipeline, we used Pyserini, Anserini's Python wrapper [28].

For document retrieval, we adopted the BM25 ranking function, which is widely recognized for its effectiveness [29]. We selected the top 100 documents based on BM25 scores as the initial retrieval result and occasionally extended them to the top 1000 for broader coverage. Figure 3 illustrates that extending from the top 100 to the top 1000 documents increases the number of expected documents in the set by 20% (from 71% to 91% recall). This extension provides a higher chance of retrieving more relevant documents. However, it comes with a trade-off in speed, as the neural retrieval system needs to process ten times more documents. It is worth noting that if the top 100 documents already contain a sufficient number of positive documents, using the top 1000 may not yield significant gains in metrics. This observation will be later addressed in the discussion section.

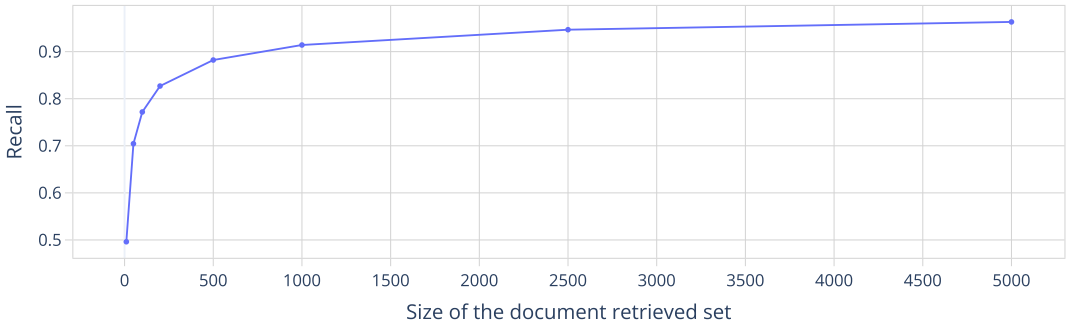

Figure 3: Recall value of the retrieved documents at different result set sizes.

The parameters for the BM25, specifically b and k_1, were selected through a preliminary hyperparameter tuning process. Figure 4 shows a summary of all the runs and their respective parameters. Based on this we adopted the parameters $k_1 = 0.5$ and $b = 0.3$.

In the second-stage, we utilized re-ranking models, which includes state-of-the-art

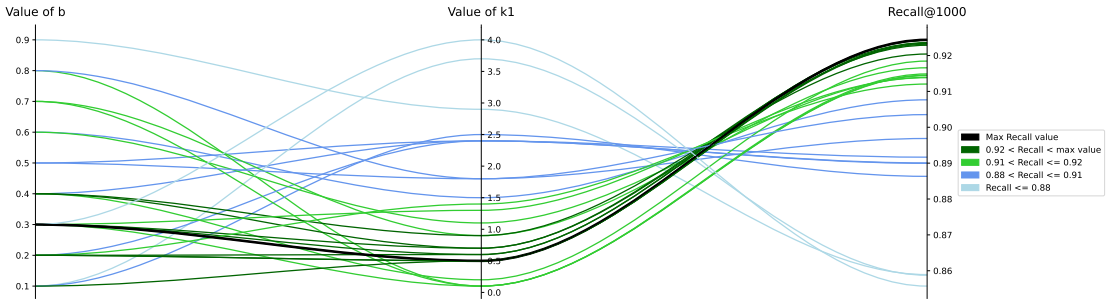

Figure 4: Parallel coordinate plot showing impact of b and k_1 hyperparameters in determining recall score during hyperparameter tuning process

transformer-based models such as PubMedbert [6] and monoT5 [5] (both base and large variantes). We also considered the BioGPT [30] and Pegasus [31] models, but due to their higher computational cost, they were discarded. These models were trained using both pointwise and pairwise approaches to evaluate their effectiveness in differing scenarios. To expand upon the limited availability of training data, we also experimented with including synthetic data in our training regimen as a pretraining mechanism.

Finally, to consolidate the output from several models, we used the reciprocal rank fusion (RRF) [32]. This approach acts as an ensemble technique to improve the overall ranking order of the relevant documents by considering the judgment of multiple models.

3.2.1. Submissions

The runs submitted to the phase A challenge were ensembles of various trained models with different checkpoints. The various systems submitted are briefly described in Table 1.

Table 1
Summary table of the configuration of the submitted system for each evaluation batch. The 'x' means that the system was used in that specific batch.

System	Synthetic	Training	# Models			Submissions			
			BERT	T5-L	T5-B	B1	B2	B3	B4
System-0	False	Pointwise	3	-	-	x	x	x	x
System-1	False	Pointwise	5	-	-	x	x	x	x
System-2	False	Pointwise	-	2	2	x	-	-	-
	True	Pointwise	7	-	-	-	x	x	x
System-3	False	Pointwise	2	2	-	x	-	-	-
	True	Pairwise	7	-	-	-	x	x	-
	Mixed	Pointwise	5	-	-	-	-	-	x
System-4	False	Pointwise	-	4	-	x	-	-	-
	Mixed	Mixed	22	3	-	-	x	x	x

With more detail:

- **System-0**: This system contained 3 PubMedBERT models that re-ranked 1000 documents fetched from BM25.
- **System-1**: This system contained 5 PubMedBERT models that re-ranked 100 documents fetched from BM25.
- **System-2**: For the first batch, the system contained 2 T5-base and 2 T5-Large models that re-ranked 100 documents from BM25. For the rest of the batches, the system used an ensemble of models trained on synthetic data and then fine-tuned on the challenge data. The models ensembled were 7 PubMedBERT models, 5 of which re-ranked 1000 documents, and the remaining 2 re-ranked 200 documents.
- **System-3**: In the first batch, an ensemble of 2 T5-base models and 2 PubMedBERT models were used to re-rank 100 documents. The following 2 batches investigated pairwise training with synthetic data, where 7 PubMedBERT models were trained using a pairwise loss function. Among them, 4 models re-ranked 100 documents, and 3 re-ranked 500 documents. In the final batch, some models were removed and replaced with models from the first system.
- **System-4**: In the first batch, the system contained 2 T5-Large models and 2 T5-Base models. The Large models re-ranked 1000 documents, and the Base models re-ranked 100 documents. In the remaining submissions, we ensembled most of our trained models, reaching a total of 25 models. However, it should be noted that only 24 models were used in the last batch.

3.3. Phase B

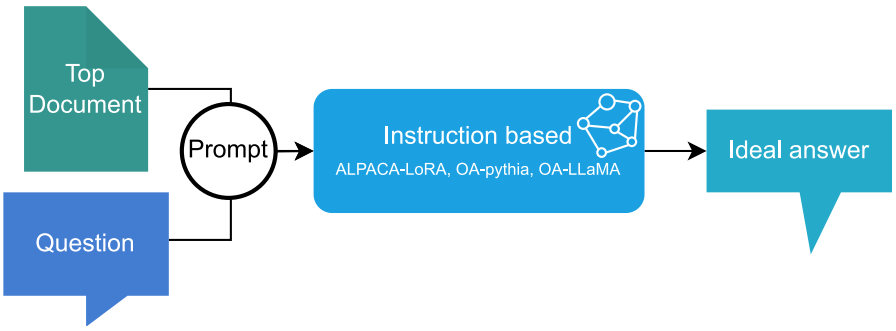

Figure 5: Overview of our methodology for participating in the phase B of the BioASQ challenge.

To provide a natural language answer to a question, we adopted an exploratory approach, testing various prompts and models to gauge their effectiveness in generating precise and meaningful answers. Recognizing the varying complexities inherent to different question types, we also experimented with per-question type prompting. This approach considers the nature of the question—be it factoid, list, or summary—and tailors the model's prompt accordingly, enabling more accurate and contextually relevant responses, see Prompt 2 as reference.

```
Below is an instruction that describes a task,
paired with an input that provides further
context. Write a response that appropriately
completes the request.
### Instruction:
{instruction}
### Input:
ABSTRACT: {text}

QUESTION: {question}
### Response:
```

Prompt 2: Example of a zero-shot prompt for generation answers with the Alpaca-LoRa model. The text within brackets correspond to placehorders for the instruction, question and article text. The default instruction was "Given the ABSTRACT, answer the QUESTION". For yesno type of question we used the following "Given the input ABSTRACT produce a yes or no answer to QUESTION", while for the summary type we used "Given the input ABSTRACT produce a short and concise answer to QUESTION".

Context selection should play a large part in the quality of the text generation. We tested this using our top retrieved article from phase A, the top gold standard article, or a combination of both as context for the model. The latter was accomplished by selection the gold standard article that was ranked higher according to our model. A key focus of our experimentation was the application of various advanced language models such as ALPACA-LoRA (13 billion), OA-Pythia (12 billion) and OA-LLaMA (30 billion) models. Furthermore, we dabbled with different answer-generation strategies, including random sampling, beam search and contrastive search. In random search, a random token is selected for the next token following the probability distribution of the model. In beam search, multiple possible continuations at each step are explored based on a predefined beam width, aiming to find the most probable and coherent sequence of words. Contrastive search involves searching for alternative continuations or completions by contrasting different options and selecting the most distinctive or interesting one. We also extensively tested different hyperparameters for model generation, including temperature and the maximum token length. This experimentation allowed us to fine-tune our models' performance, leading to more precise and informative answers.

3.3.1. Submissions

For Phase B, our submissions consisted of various instruction transformer-based models, each described concisely in Table 2. The "Document Source" column specifies the origin of the article used as context for answer generation. Specifically, "System-0"and "System-4" correspond to the highest scoring documents outputted by the respective phase A system. On the other hand, "Gold" indicates that the document was obtained from the provided gold standard.

More precisely, due to time constraints, for the first batch we selected only a single model

Table 2
Summary table of the configuration of the submitted system for each evaluation batch for phase B.

Batch	System	Model			Document Source		
		LoRA-Alpaca	OA-Pythia	OA-LLaMA	System-0	System-4	Gold
1	System-0	7b	-	-	x	-	-
2	System-0	7b	-	-	-	x	-
	System-1	7b	-	-	-	x	x
	System-2	30b	-	-	-	x	-
	System-3	30b	-	-	-	x	x
	System-4	30b	-	-	-	x	x
3	System-0	-	12b	-	x	-	x
	System-2	-	12b	-	-	x	x
	System-3	-	12b	-	-	x	-
4	System-0	-	-	30b	x	-	-
	System-1	-	-	30b	-	x	x
	System-2	-	12b	-	-	x	x
	System-3	-	-	30b	-	x	-
	System-4	-	12b	-	-	x	-

for submission using the top ranked document from our best performing model in phase A. This was seen as a naive approach, which is why in further batches we tested with both our models and the gold standard document to answer a question. In the second batch we used the same Alpaca-LoRA model, with the addition of the 30 billion parameter version, tested on the best performing model of the batch and also using the gold standard documents. For the third batch, we were unable to submit 5 submissions due to technical problems, however in this submission we changed the model to OpenAssistant's Pythia 12 billion parameter model. In the final batch, we additionally tested the OpenAssistant LLaMA model with 30 billion parameters. Regarding the generation strategies, we adopted contrastive search for the Alpaca-LoRA and random sample with high confidence for the OpenAssistant variantes.

4. Results

This section starts by addressing our validation results measured over a subset of the training data. Then we show the official preliminary results of the BioASQ challenge for phase A and B. Note that the preliminary results are the results available at time of writing and are due to changes after the reevaluation period. To see the official results, use the BioASQ 11B official leaderboard[3].

[3]Phase A: http://participants-area.bioasq.org/results/11b/phaseA/, Phase B: http://participants-area.bioasq.org/results/11b/phaseB/

4.1. Validation results

The validation of our models was conducted to assess their performance and gain valuable insights in their configuration. In this section, we summarize the validation results obtained over a subset of the training data. More precisely, we performed a stratified train/test split of 95/5 of the dataset, which corresponds to 3292 questions for training and 173 for validation. Taking into consideration that the official evaluation batch only contains 90 questions, we believe that our split was representative.

Table 3 summarizes the best validation results of various neural relevance models trained on different subsets of data and also the BM25 baseline. The models were evaluated based on their Mean Average Precision at 10 (MAP@10) score, which measures the average precision of the top 10 retrieved documents for each query. Each neural model was trained using different combinations of training data, including synthetic and gold standard datasets.

Table 3
Summary of the best validation results for various relevance models and different combinations of training data.

Model type	Training data		MAP@10
	Synthetic	Gold standard	
PubMedBERT	x	x	**58.06**
PubMedBERT		x	57.75
PubMedBERT	x		51.74
monoT5-base	x		49.89
monoT5-base		x	51.90
monoT5-large		x	57.36
BM25			43.58

Overall, when training with the gold standard data, all the reranking methods are capable to improve upon the baseline, reinforcing the idea that it is beneficial to adopt a reranking method as a second-stage mechanism of a retrieval pipeline. Regarding the architectures, the PubMedBERT and monoT5-large architectures managed to achieve comparable performances, whilst the monoT5-base architecture achieved considerably poor results. This disparity may be attributed to the fact that monoT5 is a sequence-to-sequence model that directly learns the retrieval task using natural language, placing greater emphasis on the quality of the underlying language model, and, therefore, their size.

Notably, the best configuration we obtained involved training the PubMedBERT model with synthetically generated data and subsequently fine-tuning it with the gold data. This outcome highlights the beneficial impact of incorporating synthetic data.

Furthermore, an unexpected result emerged when comparing the performance of models that were only trained with synthetic data against the BM25 baseline. It was surprising to observe that using only synthetic data yielded improvements over the performance of BM25. This suggests that it is indeed possible to train models without relying on gold data and still achieve superior performance compared to traditional baselines such as BM25. This finding opens up new possibilities for model training and highlights the potential of synthetic data as a

valuable resource for retrieval tasks where no labelled data is available.

4.2. Phase A

The preliminary results of our submissions are displayed in Table 4, showcasing the rankings based on Mean Average Precision at 10 (MAP@10). Additionally, we provide the results regarding F1-score at 10, offering insights into the trade-off between precision and recall across the systems. The Top Competitor represents the most successful system among all competitor systems. Overall, we achieved highly competitive results, achieving the best-performing system in the first and second batches in MAP@10 and the best-performing system in all the batches in the F1-score. Significantly, the systems that attained these high F1-scores were relevance models, designed to discard documents if the likelihood of their relevance fell below 1%. Consequently, for questions with less than 10 positive documents, these systems were capable of outputting fewer than 10 documents, thus increasing precision compared to a ranking model that consistently outputs 10 documents regardless of their scores.

Table 4
Preliminary results made available by the BioASQ team for all the batches for phase A.

System	Batch 1			Batch 2			Batch 3			Batch 4		
	F1	MAP	Rank	F1	MAP	Rank	F1	MAP	Rank	F1	MAP	Rank
System-0	**27.74**	**45.90**	**1**	20.21	35.80	9	**19.11**	28.39	13	18.09	25.80	10
System-1	27.69	45.31	2	**20.88**	38.40	2	19.09	28.94	12	**19.25**	27.03	8
System-2	21.92	45.22	3	16.32	35.35	11	17.83	28.96	11	18.13	27.01	9
System-3	24.18	44.99	4	14.97	34.61	12	12.64	27.20	14	12.81	27.63	6
System-4	21.83	42.75	10	16.18	**38.52**	**1**	12.77	30.42	4	12.73	27.70	5
Baseline	12.00	34.50	-	5.10	29.96	-	5.91	22.42	-	5.35	19.96	-
Top Competitor	18.23	44.62	4	15.98	37.42	3	13.20	**31.85**	**1**	14.25	**32.24**	**1**
Median	16.78	37.32	17	13.72	27.81	17	11.10	23.26	18	10.42	21.47	14

Comparing now the performance between the systems, the initial two, namely System-0 and System-1, were employed to study the difference between re-ranking 1,000 and 100 documents. An analysis of these models' results across various batches revealed that the performance was not significantly affected by the increase in re-ranked documents. This observation will be further revisited in the subsequent discussion section.

Upon evaluating the remaining systems for the first batch, it was discerned that the utilization of T5 models did not significantly enhance performance compared to the BERT models. This observation carries substantial importance, especially given that the inference time for T5 models exceeded that of the BERT-based models. Consequently, the decision was taken to cease the deployment of T5 models in subsequent submissions, favouring instead the more efficient BERT models, which delivered adequate performance. Furthermore, System-4 for the initial batch demonstrated an unexpectedly lower Mean Average Precision (MAP) compared to the outcomes of other ensemble methods. This indicates that the specific configuration or ensemble of models in System-4 did not yield the anticipated results.

Furthermore, upon comparing System-2 and System-3, the distinctive variance can be traced back to the training technique deployed. It was deduced that pairwise training slightly under-

performed compared to pointwise training methods. As a consequence, only pointwise training was used in the final batch.

Turning to the final system, System-4, it was observed that ensembling more models consistently outperformed the other systems in all instances. Again, this is an anticipated result corroborating existing literature [33].

4.3. Phase B

The preliminary, automatically generated results regarding the phase B are displayed in Table 5. Before analysis of the results, it is important to note that the metrics used to evaluate the systems in the competition is a manual evaluation of the ideal answers, rather than these automatic metrics. Overall, our systems showed a reasonable performance on the automatic metrics, at best placing 9th, and the remainder of the submission are mostly below the median position of the submissions. The metrics used in the competition Rougue-2 and Rogue-SU4, in the results presented, we show Rogue-2(F1). Given our approach used to generate the text was from an unsupervised nature, this is not surprising, as our system is not guided to generate expected BioASQ answers. Nevertheless, the answers can be correct and therefore missed by the automatic metrics. In Appendix A we showcase some examples of answerers that were generated by the OA-LLAMA-30B model and the OA-Pythia-12B model.

Table 5
Automatic evaluation made available by the BioASQ team for all the batches for phase B.

System	Batch 1		Batch 2		Batch 3		Batch 4	
	R-2(F1)	Rank	R-2(F1)	Rank	R-2(F1)	Rank	R-2(F1)	Rank
System-0	29.07	12	17.84	21	12.908	26	15.86	26
System-1	-	-	22.57	18	-	-	24.33	24
System-2	-	-	28.53	16	12.14	25	10.76	33
System-3	-	-	32.42	9	08.91	30	15.82	27
System-4	-	-	31.69	10	-	-	08.92	35
Median	31.03	10	28.53	16	20.54	18	32.12	19
Best	**40.63**	**1**	**32.90**	**1**	**37.41**	**1**	**40.84**	**1**

5. Discussion

Throughout phase A, we observed that our reranking methods consistently enhanced the baseline ranking order, which is known to be a challenging task, as mentioned in [9]. To provide a more tangible visualization of these improvements, we present in Figure 6 the ratio of improvement achieved by our reranking models in comparison to the BM25 baseline. Remarkably, across all batches, our reranking models achieved an average improvement of 30%, and in some cases, even nearing 40%. We attribute these notable gains to two primary factors. Firstly, the

quality of our training data played a crucial role, as we focused on meticulous cleaning of the gold standard data prior to training our models. Additionally, the availability of more advanced training algorithms enabled efficient fine-tuning of entire transformer-based models, further contributing to the model's performance.

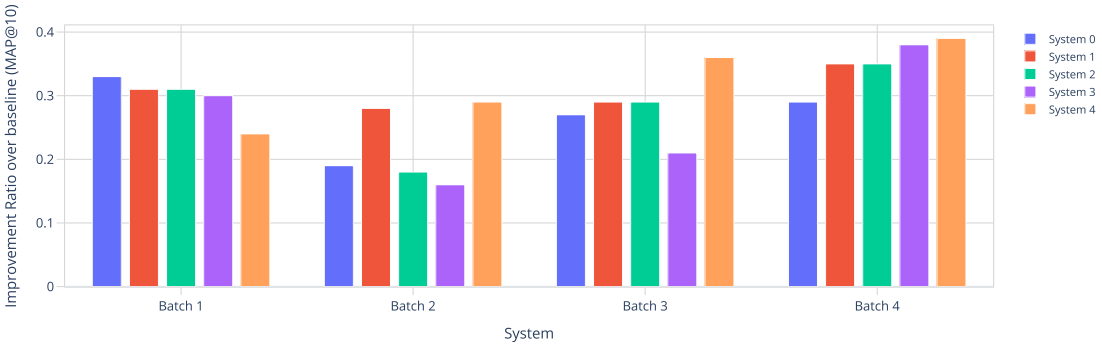

Figure 6: Improvement ratio over the baseline BM25 in terms of MAP@10 of all the submitted systems in all the batches.

Next, we delve into a detailed discussion of various factors that impact the performance of our systems, namely model architecture, loss function, the number of reranked documents, and the utilization of synthetic data during pretraining. To facilitate this analysis, we present parallel plots in Figures 7 and 8, showcasing these variables for the models used in the second and fourth batches, respectively. Although we focus on these two batches for clarity, it is worth noting that the first and third batches follow similar patterns.

Upon examining both figures, it becomes evident that the preferred architecture and loss function for optimal performance are PubMedBERT and pointwise, respectively, as these models achieved the highest MAP@10 scores according to the plots. Furthermore, in terms of the number of documents used for reranking, it appears that increasing the count does not lead to improved metrics. This observation may be attributed to the fact that the evaluation metrics only consider the top ten documents. This consideration arises from the fact that the BioASQ team evaluates the system's performance based on the top 10 documents only. Therefore, when there are already enough positive documents among the top 100, reranking a larger number of documents may not result in noticeable improvements.

Finally, the impact of synthetic data yields contradictory results. In the case of the second batch (Figure 7), incorporating synthetic data did not contribute to an overall performance improvement. However, for the fourth batch, it did exhibit a positive effect. We speculate that this discrepancy may be attributed to the quantity and coverage of the synthetic questions generated. Specifically, for the fourth batch, the test set questions may have been closer to those synthetically generated, particularly in terms of the documents used for their generation. Further investigation is needed to validate this hypothesis.

The generation phase (Phase B) of our system presented several insightful findings. Notably, we observed a positive correlation between the size of the language model and the quality of generated answers, which aligns with previous findings that larger models generally tend to

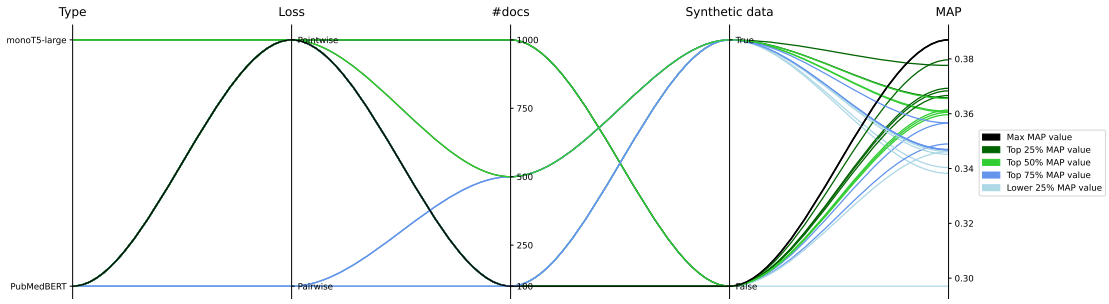

Figure 7: Parallel plot showing the impact of different hyperparameters in the MAP score of the neural retrieval models used in batch 2.

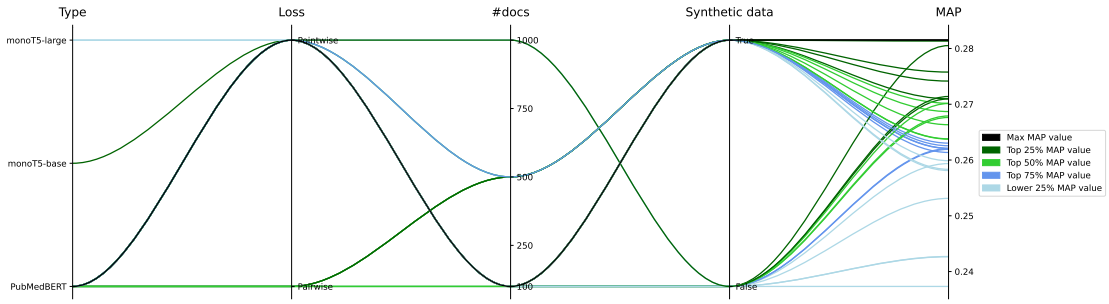

Figure 8: Parallel plot showing the impact of different hyperparameters in the MAP score of the neural retrieval models used in batch 4.

perform better [34, 35].

Additionally, we found that small modifications to the prompt significantly impacted the system's output, suggesting that the models may struggle with generalization. This effect was more pronounced in smaller models, indicating that fine-tuning may be necessary to achieve optimal results [36]. In contrast, for larger models, the quality of generation was less affected by the prompt variation, showcasing their robustness.

Overall, the text generation quality was satisfactory, demonstrating coherence and relevance to the biomedical questions. The employment of different prompts for various question types particularly enhanced the performance of smaller models, aligning them more closely with the inherent intricacies of each question category.

We also explored ensembling multiple contexts to improve answer diversity and depth. Unfortunately, our attempts were not fruitful, suggesting that this method might require further refinement for it to be effective in this specific task.

Finally, we hypothesize that with some pre-training or domain-specific training, the models might perform even better. Such training could enhance their ability to generate precise and contextually accurate answers for biomedical questions, further increasing their utility in real-world applications [37].

6. Conclusion

In this paper, we detailed our participation on tasks B phase A and B of the eleventh edition of the BioASQ challenge. For phase A, we adopted a two-stage retrieval pipeline comprising the Anserini BM25 as the initial stage, followed by reranker models based on PubMedBERT and monoT5 transformer-based models. In order to effectively train the reranker models effectively, we enhanced the quality of the gold standard data through careful cleaning and also explored synthetic data augmentation techniques through question generation. By using these methods, we achieved significant improvements over the baseline ranking order. Our systems, were able to place first in various batches of the competition.

For phase B, our approach involved leveraging instruction transformer-based models to generate answers conditioned on the articles retrieved during phase A in a zero-shot setting. We observed a positive correlation between the size of the language model and the quality of the generated answers. Smaller models were more sensitive to prompt variations, indicating the need for fine-tuning to enhance their performance. Larger models, on the other hand, exhibited greater robustness and generated coherent and relevant answers. The employment of different prompts for various question types improved the performance of smaller models, aligning them more closely with the specific intricacies of each question category. Overall our performance on the phase B, according to the automatic metrics, was mediocre. However, we believe that further manual analysis is required for a fair evaluation given the unsupervised nature of our generation method.

7. Future Work

In terms of the direction for future work, several promising avenues appear worthy of exploration, particularly for Phase B of our system.

First, while our initial attempts to join multiple contexts (ensembling) did not yield the anticipated results, we believe this approach still holds considerable potential. Therefore, refining our ensembling techniques to effectively integrate different contexts into the question-answering process will be an area of interest. This could potentially enhance both the diversity and depth of our generated answers.

Second, the incorporation of snippet extraction as an intermediary step in our approach might serve to enhance the precision of our answer generation. Extracting relevant snippets from the retrieved documents could refine the context that is fed into our models, potentially leading to more accurate and relevant answers. Several recent works have reported success using such techniques [38].

Lastly, fine-tuning the models specifically for the ideal answers in Phase B of Task B could further boost performance. As we observed that prompt changes significantly impacted the system's output, especially for smaller models, task-specific fine-tuning might increase the models' robustness against these changes and enhance their overall performance. In fact, recent studies have shown that fine-tuning large-scale pre-trained models on downstream tasks can lead to substantial improvements in task performance [35, 36].

Acknowledgments

This work was partially supported by national funds through the Foundation for Science and Technology (FCT) in the context of the project UIDB/00127/2020. Tiago Almeida is funded by FCT under the grant 2020.05784.BD. Jorge Miguel Silva has received funding from the EC under grant agreement 101081813, Genomic Data Infrastructure.

References

[1] I. Klerings, A. S. Weinhandl, K. J. Thaler, Information overload in healthcare: too much of a good thing?, Zeitschrift für Evidenz, Fortbildung und Qualität im Gesundheitswesen 109 (2015) 285–290.

[2] G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, et al., BioASQ: A challenge on large-scale biomedical semantic indexing and question answering., in: AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text, Arlington, VA: Citeseer, 2012.

[3] A. Nentidis, G. Katsimpras, A. Krithara, S. Lima-López, E. Farré-Maduell, L. Gasco, M. Krallinger, G. Paliouras, Overview of BioASQ 2023: The eleventh BioASQ challenge on Large-Scale Biomedical Semantic Indexing and Question Answering, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), 2023.

[4] S. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. URL: https://www.nowpublishers.com/article/Details/INR-019. doi:10.1561/1500000019.

[5] R. Nogueira, Z. Jiang, R. Pradeep, J. Lin, Document Ranking with a Pretrained Sequence-to-Sequence Model, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 708–718. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.63. doi:10.18653/v1/2020.findings-emnlp.63.

[6] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, ACM Transactions on Computing for Healthcare (HEALTH) 3 (2021) 1–23.

[7] A. Nentidis, G. Katsimpras, E. Vandorou, A. Krithara, G. Paliouras, Overview of BioASQ tasks 10a, 10b and Synergy10 in CLEF2022, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 171–178. URL: https://ceur-ws.org/Vol-3180/paper-10.pdf.

[8] T. Almeida, A. Pinho, R. Pereira, S. Matos, Deep Learning solutions based on fixed contextualized embeddings from PubMedBERT on BioASQ 10b and traditional IR in Synergy, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September

5th - to - 8th, 2022, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 204–221. URL: https://ceur-ws.org/Vol-3180/paper-12.pdf.

[9] T. Almeida, S. Matos, Universal passage weighting mecanism (UPWM) in BioASQ 9b, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 196–212. URL: https://ceur-ws.org/Vol-2936/paper-13.pdf.

[10] Z. Dai, J. Callan, Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval, 2019. URL: http://arxiv.org/abs/1910.10687. doi:10.48550/arXiv.1910.10687, arXiv:1910.10687 [cs].

[11] Z. Dai, J. Callan, Context-Aware Document Term Weighting for Ad-Hoc Search, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1897–1907. URL: https://dl.acm.org/doi/10.1145/3366423.3380258. doi:10.1145/3366423.3380258.

[12] L. Gao, J. Callan, Condenser: a pre-training architecture for dense retrieval, arXiv preprint arXiv:2104.08253 (2021).

[13] V. Karpukhin, B. Oguz, S. Min, L. Wu, S. Edunov, D. Chen, W. Yih, Dense passage retrieval for open-domain question answering, CoRR abs/2004.04906 (2020). URL: https://arxiv.org/abs/2004.04906. arXiv:2004.04906.

[14] L. Xiong, C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, A. Overwijk, Approximate nearest neighbor negative contrastive learning for dense text retrieval, CoRR abs/2007.00808 (2020). URL: https://arxiv.org/abs/2007.00808. arXiv:2007.00808.

[15] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, IEEE Transactions on Big Data 7 (2019) 535–547.

[16] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, S. Ma, Optimizing dense retrieval model training with hard negatives, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1503–1512.

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[18] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[19] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Stanford alpaca: An instruction-following LLaMA model, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[20] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, O. van der Wal, Pythia: A suite for analyzing large language models across training and scaling, 2023. arXiv:2304.01373.

[21] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat,

S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, PaLM: Scaling language modeling with pathways, 2022. arXiv:2204.02311.

[22] S. Black, L. Gao, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large scale autoregressive language modeling with mesh-tensorflow, 2021. URL: https://doi.org/10.5281/zenodo.5297715. doi:10.5281/zenodo.5297715, If you use this software, please cite it using these metadata.

[23] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[24] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.-R. Tam, K. Stevens, A. Barhoum, N. M. Duc, O. Stanley, R. Nagyfi, S. ES, S. Suri, D. Glushkov, A. Dantuluri, A. Maguire, C. Schuhmann, H. Nguyen, A. Mattick, OpenAssistant conversations – democratizing large language model alignment, 2023. arXiv:2304.07327.

[25] P. Yang, H. Fang, J. Lin, Anserini: Enabling the use of Lucene for information retrieval research, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 1253–1256. URL: https://doi.org/10.1145/3077136.3080721. doi:10.1145/3077136.3080721.

[26] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 6894–6910. URL: https://aclanthology.org/2021.emnlp-main.552. doi:10.18653/v1/2021.emnlp-main.552.

[27] P. Yang, H. Fang, J. Lin, Anserini: enabling the use of Lucene for information retrieval research, in: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 1253–1256.

[28] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: An easy-to-use python toolkit to support replicable ir research with sparse and dense representations, arXiv preprint arXiv:2102.10073 (2021).

[29] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: BM25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.

[30] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, T.-Y. Liu, BioGPT: generative pre-trained transformer for biomedical text generation and mining, Briefings in Bioinformatics 23 (2022). URL: https://doi.org/10.1093/bib/bbac409. doi:10.1093/bib/bbac409.

[31] J. Zhang, Y. Zhao, M. Saleh, P. J. Liu, PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[32] G. V. Cormack, C. L. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, 2009, pp. 758–759.

[33] G. V. Cormack, C. L. A. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 758–759. URL: https://doi.org/10.1145/1571941.1572114. doi:10.1145/1571941.1572114.

[34] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[36] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).

[37] U. Khandelwal, K. Clark, D. Jurafsky, L. Kaiser, Sample efficient text summarization using a single pre-trained transformer, arXiv preprint arXiv:1905.08836 (2019).

[38] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, arXiv preprint arXiv:1905.07129 (2019).

A. Examples of answers generation.

Question: Which are the types of Charcot-Marie-Tooth Disease (CMT)?

OA-LLaMA-30B: There are four main types of Charcot-Marie-Tooth Disease (CMT): CMT1, CMT2, CMT3, and CMT4.

Pythia-12B: The four types of CHARCOT-MARIE-TOOTH DISEASE (OMIM #301500) are caused by mutations in genes encoding components of paranodal axohemisegmental junctions (AHJs). They include demyelinating forms due to loss-of-function mutations in NEURL4 (Type 1; Oculomotor Apraxia Lateralis 4); Dejerine-Sottas Dysarthria (DSD; Type 2A; MOCS2A); Kugelberg-Welander Spastic Paraplegia (KWSPRAP; Type 2B; SPG11) and Hereditary Diffuse Neuronal Degeneration (HDND; Type 3; VPS13A). Inherited Axon Guarinoni Juntos (IAGJ; Type 4; UNC5H1) is characterized by onion bulb formation at AHJs due to increased expression of Caspr and contactin-associated protein (CTDPNNP1) resulting from gain-of-function mutations in their promoter regions.[6]

Answer generated 1: Question id 6446caac57b1c7a315000074

Question: What are Luminopsins?

OA-LLaMA-30B: Luminopsins are chimeric proteins consisting of a luciferase fused to an opsin that allow for control of neuronal activity through optogenetics.

Pythia-12B: Luminopsins (abbreviated as LMOs or Lumisomes) are genetically encoded photoswitches that respond to blue light by changing their absorption spectrum from 500 nm to 600-700nm.[1][2] They consist of a fusion protein betweenluciferases such as firefly Lucina pellucida[b] (...)

Answer generated 2: Question id 6419cd02690f196b5100003a

Question: Is SARS-CoV-2 transmitted through breast milk?

OA-LLaMA-30B: No, according to this study, there is no evidence that breast milk contains infectious SARS-COVID-2.

Pythia-12B: No, according to this study which analyzed over 100 women's breast milk.

Answer generated 3: Question id 641791ed690f196b51000032

Question: Which amino acid in implicated in the Blue diaper syndrome?

OA-LLaMA-30B: The Blue diaper syndrome is caused by a frameshift mutation in the Proprotein Convertase Subtilisin/Kexin Type 1 (PCSK1) gene which results in the substitution of Valine for Leucine at position 227 (p.[Val227Leufs*12])

Pythia-12B: The patient's symptoms are consistent with those seen in individuals with Proprotein Convertase Subtilisin/Kexin Type 1 (PCSK1) mutations, which result in decreased activity or lack of expression of the protein. Specifically, these mutations lead to a decrease in proteolytic processing of peptides involved in nutrient transport across epithelia such as jejunum and ileum, resulting in malabsorption of carbohydrates, fats, and proteins. This may explain why the patient experienced episodes of fasting hypokalemia secondary to carbohydrate and fat maldigestion while also experiencing diarrhea due to poor digestion and malassimilation of macronutrients. It should be noted that other factors such as imbalances in electrolytes and bicarbonate production could contribute to the patient's phenotype as well. Additionally, given the rarity of the condition, it is possible that there are undiscovered modifier genes that influence penetrance and expressivity of PCSK1 mutations, further complicating diagnosis and management of affected individuals (...)

Answer generated 4: Question id 63f57d9b33942b094c000004