Query Generation leveraging Large Language Models for Data Augmentation

Artur Guimarães 86389
João Daniel Silva 86445
João Coelho 86448

Abstract

Data generation resorting to Large Language Models has gained attention given the zero-shot generalization ability of recently proposed models. In this project, we worked on query generation given an input document. We compared the performance of prompting methods and fine-tuning of LLMs under limited resources, also considering a query filtering step. We concluded that the tuned methods were superior to zero/few-shot prompting. Furthermore, we were able to generate sets of queries that improved re-rankers when used for training, and also show the usefulness of synthetic data for other tasks, such as query expansion and question generation from image captions. The developed code is publicly available here.

1 Introduction

In this practical project we explored methods for query generation given an input document. Applications of this technique include the creation of synthetic data to train models, and document expansion for lexical retrieval systems.

We consider few-shot prompting with fixed examples as previously proposed for this task [5], and extend it by considering in-context dynamic examples [36], and chain-of-thought enhanced prompts [15]. Furthermore, we instruct fine-tune [74] a LLM using low-rank adaptation [21].

Other authors [18] have argued that since generation models are prone to hallucination, sometimes the generated queries will not be related to the input document, hindering the performance on downstream tasks. As such, besides aiming to minimize hallucination through our proposed methods, we compared two methods for query filtering, (i) through semantic similarity, and (ii) through question-answering (QA) models.

Regarding evaluation, we compared the quality of the queries generated by the multiple approaches by using them for three different downstream tasks, (i) document re-ranking on the MARCO document dataset [42], (ii) first stage retrieval with query expansion for the TREC Clinical Trials task [60], and (iii) VQA model pretraining by generating queries for captions from NWPU-Captions [8].

Results show that synthetic data is indeed useful for the tasks in hand, with fine-tuned generation models outperforming zero and few-shotting pre-trained LLMs. Query filtering shows to be important but the QA-based version is not appropriate on long input documents.

2 Related Work

In this section, we will introduce large language models, discuss relevant literature on query generation and its applications, and highlight previous work related to each of the downstream tasks.

2.1 Large Language Models

Currently, the paradigm of machine learning for natural language processing tasks has shifted tremendously in a short time span, favoring transfer learning from pre-trained language models to downstream tasks over creating specialized models for each task. This has resulted in groundbreaking

state-of-the-art performance across multiple fields, with foundational models like LLaMa [71], PALM 2 [3] and GPT-4 [48] leading the way.

Some fine-tuning approaches have demonstrated impressive zero-shot generalization to different tasks, for instance the usage reinforcement learning with human feedback [17], and instruct fine-tuning [74]. This, however, still requires training models with a very high number of parameters, which hinders end-to-end tuning especially under limited hardware conditions. Solutions like low-rank adaptation (LoRA) [21] aim to overcome this issue by selecting only a small percentage of trainable parameters, by decomposing weight matrices into low-rank factors, and quantizing the model to 16-bit.

Prompt-based methods [37], are particularly useful when training is not possible, since instead of fine-tuning the model, the generation capabilities of LLMs are directly exploited. Previous work has shown very convincing evidence on how prompt usage substantially improves results [85], specially the few-shot prompting technique [6], which conditions the output generation on a small set of provided examples from the task in hand.

Regarding model architectures, in this work we consider the T5 [56] as a baseline, and the recently proposed LLaMA [70]. The T5 is a multitask model that addresses problems as sequence-to-sequence tasks. T5's architecture is based on the Transformer [72], considering both the encoder and the decoder. Differences from the original architecture include the usage of relative scalar embeddings [63]. It was pre-trained using a masked language model objective similar to BERT [14], and other approaches such as next-word prediction and de-shuffling. LLaMA is a causal language model, and also considers a different position encoding strategy by using rotary position embeddings [65]. It was trained on over one trillion tokens from multiple languages, and the version of the model with 13 billion parameters outperforms the much larger GPT-3 (175B billion parameters) in multiple tasks.

2.2 Query Generation

Regarding data augmentation, the main motive for its usage is the necessity of data for tasks where data is lackluster [78]. This can be done by applying transformations to pre-existing data [20, 11, 76], by employing small-scale [16, 31] or large-scale [54, 2, 62, 41] language models, greatly enhancing how scalable and efficient these techniques can be over manually creating human generated datasets.

For the specific case of query generation, Doc2Query [45] is, to the best of our knowledge, the first work that used transformer-based models to generate queries given an input document. Later, DocT5Query [44] was proposed as an extension that replaced the transformer with a T5 model. In Doc2Query [18], the authors argue that since generation models can hallucinate, some output queries may not be useful, showing that properly filtering the queries improves their results in downstream tasks. More recently, Large Language Models have been used for query generation. For instance, InPars [5] proposes to few-shot prompt GPT-3 [6] with context examples, and also shows that query filtering, by removing queries with low generation probability, improves results on downstream Information Retrieval tasks. This model has been extended in InPars-v2 [29], where the model was changed to an open-source alternative, GPT-J [73], and a neural model is used to assign relevancy scores to the generated query and respective document, removing pairs below a threshold.

2.3 Document Retrieval

Within the field of information retrieval, neural methods currently achieve state-of-the-art results, particularly on tasks that include ranking short text passages according to relevance towards user questions [81]. Most approaches rely on Transformer-based neural language models, either following a bi-encoder or a cross-encoder architecture.

Bi-encoders encode queries and passages independently [59, 53]. This allows the offline indexing of individual passage representations through methods that support the fast execution of maximum inner product searches [30]. Conversely, cross-encoders generate a representation for the concatenation of a query and a passage, directly modelling the interactions between these two components [43, 50]. The representation can then be used to predict a relevance score for the passage to the query, for example by using a feed-forward layer with a Sigmoid activation.

Besides the generation-based methods for augmentation mentioned in Section 2.2, previous data augmentation methods to generate training data for retrieval models would leverage cross-encoders to score sampled unlabeled query-document pairs [53, 69].

Instead of generating more training data, augmentation models can also be considered to extend either queries or documents with new terms, which can help lexical retrievals deal with vocabulary mismatch [28, 45, 44]. In retrieval tasks with long queries, a similar approach named NQS [52] can be applied, where multiple smaller queries are generated from the large query. Then, independent ranks are computed for each generated query, and ensembled with a method such as RRF [9].

In this work, we will train models (with generated queries) for document re-ranking, i.e, the task of re-ranking the top-N documents for a query, retrieved initially through an efficient first-stage method (i.e., a bi-encoder or lexical baselines such as BM25 [61] or RM3 [28]). For that, we will use the MonoT5 [46] architecture, which casts document re-ranking as a sequence-to-sequence task, leveraging the T5 model. Alike a cross-encoder, it receives a concatenation of a query q and a document d, but adds task-specific tokens: "**Query:** q **Document:** d **Relevant:**". During training, the model learns to generate *true* or *false* tokens, depending on the relevancy of the document to the query. During inference, documents are ranked based on the probability of generating the *true* token. Extensions to this architecture have been proposed, for instance the DuoT5 [51], which receives as input two documents and a query as in a pairwise ranking task, and RankT5 [87], which shows that the classification loss used in MonoT5 can be replaced by pairwise or list-wise ranking losses to achieve better results in ranking tasks.

2.4 Visual Question Answering for the Remote Sensing Domain

In Visual Question Answering (VQA), a system is given an image and a question and is tasked to obtain an answer according to the image contents. Recent methods have increasingly used the Transformer architecture to obtain multi-modal representations of image and text to obtain an answer either by classification [32, 67, 34] or generating text word-by-word [75, 83, 82].

Remote Sensing Visual Question Answering (RSVQA) consists of the application of VQA methods to the remote sensing domain, constituting a useful framework to extract Earth observation data. Such questions can be regarding the land cover usage, or the presence/number of objects, among others. Until recently, methods for RSVQA [39, 86, 84] would use shallow approaches based on the use of Convolution Neural Networks to obtain representations of the image, a Recurrent Neural Network based model to obtain representations of the question, and finally a fusion and classification component that combines both types of information and selects an answer from a set of candidates, thus treating the VQA problem as a multi-label classification task. However, the usage of more recent deep learning methods such as the Transformer architecture has been recently introduced [64, 7, 4, 57], together with new efforts to address the RSVQA as open-ended instead of classification [57].

Although recent methods show promise for exploration and development, the low amount of available data remains as a significant problem. While there is a very high quantity of satellite imagery, there is a much lower quantity of pairs of image and text. The RSVQA datasets that exist also lack diversity in geographical areas and question types, as they were assembled with template generation methods [39]. To address this, we intend to create questions leveraging the information of larger remote sensing image captioning datasets [8] that can then be used as pretraining for the RSVQA task.

3 Proposed methods

In this section we present our methods, including the baselines, our extensions to previously existing models, and the query filtering models.

3.1 Query Generation Baselines

T5 Baselines: Before starting to use LLMs for the generation task, we considered T5 models as baselines. Namely, a fine-tuned T5 model [56] that considers the first 512 tokens of the input (FirstP), and a fine-tuned LongT5 model [19], that considers the first 1536 tokens of the input (LongP). Both models have approximately 220 million parameters. While it can be considered large, there is still a considerable gap between these models and the smallest LLaMA model available, which contains 7 billion parameters.

The training data was the same for both models: document-query pairs from the MARCO train splits, mixed with context-query pairs from SQuAD [58] train splits, which combined yielded a

total of 252059 training pairs. The models were validated by computing BLEU and ROUGE over development splits of MARCO. More details can be found in the appendix (A.1.1), and on the model pages we made publicly available: FirstP [22], LongP [24].

LLaMA Baselines: We used the two few-shot prompts proposed by InPars [5] as baselines, but using LLaMA as the language model. In our initial tests, we used the 7B parameter version of the model.

In the first method proposed by InPars, hereby referenced to as "vanilla", three fixed few-shot examples are used for all input documents. The second method, hereby referenced to as GBQ, maintains the usage of three fixed few-shot examples, but bad queries are also used to guide the prompt. Explicit examples of these prompts are shown at the appendix (A.2.1,A.2.2).

We argue and verify empirically through our results that these prompts may induce the model into hallucination, since by having fixed examples that have little to do with the input document, the context may not be properly built. As such, the following sections describe our extensions to deal with this problem, (i) through prompting, (ii) through model-fine-tuning, and (iii) through filtering.

3.2 LLMs for Query Generation

Few-Shot Prompting LLaMA We used two new prompts to extend previous work, both on top of the LLaMA-7B model.

Similarly to the "vanilla" InPars approach, we consider the usage of few-shot prompting. However, instead of using fixed query-document examples, we sampled similar documents, since previous research has shown that its usage improves results in multiple tasks [36, 1, 33]. In this work, we used a simple BM25 retriever for the example selection, with the index from which the sampled documents are chosen depending on the task in hand, and considered three examples. This method will hereby be referred to as "Sampled" throughout this document.

The second strategy, also uses the in-context examples, but considers chain-of-thought methods [15, 77]. For each sampled example, the model was asked to generate an explanation as to why the document is relevant to the query. This information is fed to the final prompt. Three examples were considered. This method will hereby be referred to as "Sampled + CoT" throughout this document. Explicit examples of both these prompts are shown at the appendix (A.2.3,A.2.4).

Fine-tuning LLaMA We fine-tuned LLaMA-7B using LoRA, which made only 0.06% of the model's parameters trainable (i.e., 4 million). While this number may seem small, it still takes about 15 hours to train the model on an single A100 (40GB) GPU. The data was the same that was used to train the baselines, but we followed an instruct-based prompt approach, although all instructions were the same (i.e., instruct the model to generate a query). More information on model training, including the full prompt, is available at the appendix (A.1.2), and at the model's page we made publicly available [23]. We will refer to this model as LLaMA-LoRA-Qgen.

3.3 Query Filtering

Despite the usage of the above methods in an attempt to minimize hallucination, the output queries may still be of low quality. As such, we consider two approaches to filter queries, (i) through semantic similarity, i.e., the query should have high semantic similarity to the document from which it was generated, and (ii) through question-answering (QA) models, i.e., if a QA model can answer the generated query given the document, the query should be a good candidate. For the semantic similarity model we consider a MonoT5 trained by us on MARCO. The model receives an input pair and outputs a relevancy score. Pairs with a relevancy above a defined threshold are kept. For the QA task we consider a RoBERTa [38] model trained on SQuAD, which is openly available [27]. The model receives a pair and returns the answer (if possible) together with an answer score. Pairs where an answer is extracted with score above a defined threshold are kept.

4 Experimental Evaluation

In this section we present our full experimental evaluation. We start by discussing the downstream tasks, specifying how the setup was built for each one to account for generated queries. We then present and discuss the obtained results.

4.1 Experimental Methodology and Setup

As previously stated, we evaluated the quality of generated queries by applying them in three different downstream tasks. First, we used the models to generate training data for MARCO, in order to train document re-ranking models. Then, we used the models for query expansion, by generating multiple queries for each long patient description in the TREC Clinical Trial data. Finally, we studied how well the models can zero-shot in a totally different domain, by generating queries for NWPU-Captions, to be used to pre-train a RSVQA model.

Document Re-ranking

On the MARCO dataset, we performed top-100 re-ranking using models that were trained with generated queries. The top-100 documents to be re-ranked for each query in the dev split (5193) were sampled using ANCE-MaxP [80]. The model to be fine-tuned is the `monot5-base-msmarco-10k` [26]. The MARCO metric was used, the MRR@10 [10].

Regarding re-ranker training, we used a setup of 30000 real queries from MARCO training splits, and compared it to a setup with the same 30000 real queries, combined with 10000 generated ones for documents of the collection that are not associated with real queries. For each query, 3 negative documents are sampled using the ANCE-MaxP model. The single positive pair available is repeated 3 times to balance the dataset. All proposed methods were used to generate independent sets of queries, and models with and without filtering were compared. Then, a larger experiment was conducted, by considering 200000 real queries with 10 negatives per query, further trained with 40000 generated filtered queries. More details on model training can be found in the appendix (A.1.3).

Query Expansion

For query expansion, we considered the TREC Clinical Trials (CT) data and the NQS [52] approach, aiming to improve it by changing the generation models. This task evaluates the effectiveness of systems in retrieving relevant clinical trial documents for a specific patient description. The task uses a common document collection of 375,381 clinical trial descriptions, 75 queries for the 2021 track, and 50 for the 2022 track. The official TREC-CT evaluation metrics were used, namely the P@10, NDCG@10, MRR, R-Precision, and R@1000.

The goal is to match the most relevant clinical trials to a given patient description, returning the top-1000 pairs obtained for each patient description. Ground-truth relevance judgments are available for matching, assigning a three-point scale to each relation.

Alike NQS, a first-stage retrieval pipeline was devised that uses a Pyserini [35] index in order to rank documents in accordance to each query with BM25 and RM3. Following the first-stage retrieval, generation models were used to generate synthetic queries from the original long patient description. An independent top-k ranking was computed for each generated query. The results of those rankings were ensembled to a final rank through RRF.

Regarding the proposed generation models that require training, this dataset does not have enough data to fine-tune a model for the task. As such, the models trained on MARCO were used. This is not totally out-of-domain, since approximately 10% of MARCO query-document pairs are considered to be of the medical domain [40]. Results were also compared with medAlpaca [25], which fine-tunes LLaMA-7B in the same setup as ours, but with medical data.

RSVQA Model Pretraining

The datasets for RSVQA are constituted by triplets of (image, question, answer). RSVQA benchmarks [39] were used to evaluate how the creation of synthetic data to pre-train a model can help to further the performance. Specifically, we considered a VisualBERT-like model from [64], which is a strong baseline in the domain. An EfficientNetV2 [68] is used to extract image features at different depths, resulting in 5 visual tokens. As for the questions, BERT [13] is used to tokenize the text and obtain their embeddings. These features are concatenated and passed to a Transformer encoder to obtain a multi-modal representation that is then passed to a classifier to obtain an answer.

Questions were generated from NWPU-Captions, which is a remote sensing image captioning dataset with 31,500 images and 5 captions per image, for a total 157,500 sentences. The query generators were used to produce K questions for each caption. For the baseline, the only filtering criteria for the questions were if they are answerable by a question-answering (QA) model [27], and if so that answer is saved to construct a question-answer pair associated with an image.

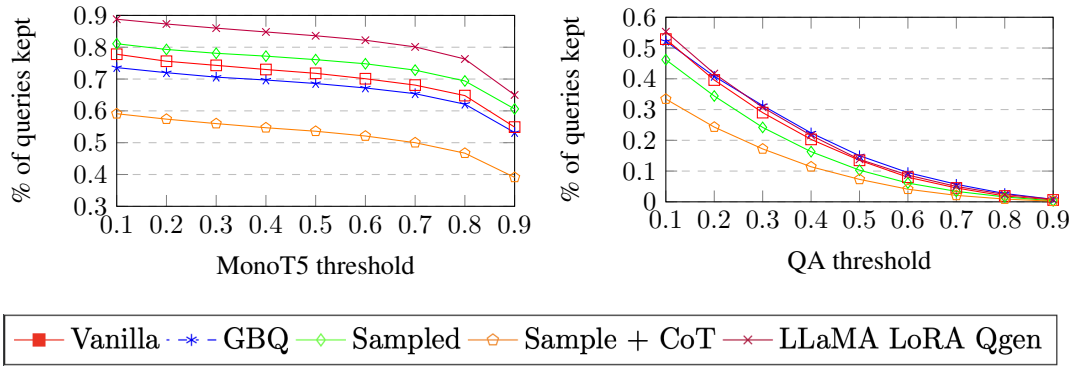

Figure 1: Percentage of queries kept with varying thresholds for MonoT5 filtering (left) and QA filtering (right), considering independent sets of 10000 generated queries for all generation models.

4.2 Experimental Results

Query Filtering

We first show insights on the query filtering process to motivate our method of choice for each task. For this, consider the line-charts in Figure 1, which shows the percentage of kept queries when generating 10000 queries with each method for MARCO documents. The low yield from the QA approach hinders its usage given the computational costs of generation. We argue that the low yield on this method is due to the underlying training on SQuAD of the QA model, since SQuAD's contexts are usually much smaller than MARCO documents. As such, we resort to the MonoT5 filtering method for tasks with longer inputs, and the QA model for tasks with smaller inputs where the answer is necessary. It is also observable that both Sampled and LLaMA-LoRA-Qgen achieve an overall higher query yield than both InPars baselines. The low yield of the Sample+CoT method was surprising, and we further address it when discussing the results for downstream tasks.

Document Re-ranking

We start by establishing our baseline results, as depicted in the first group of models in Table 1. The base model (without further fine-tuning) was improved by training with real queries only, from 0.3865 to 0.4016. This result was further improved by training with the same real queries, and 10000 queries generated by the T5-FirstP model. However, when extending the real queries with queries generated by the LLaMA baselines, Vanilla and GBQ, the results worsen to 0.4005 and 0.3991, respectively. These last two results are improved when using query MonoT5 filtering with a threshold of 70%. As such, from the baselines, we confirmed the hypotheses of hallucination in the baseline prompts, and also see that filtering is beneficial.

The second group of Table 1 shows our proposed extensions. The usage of in-context examples for LLaMA few-shot prompting led to an improvement even without filtering, when compared to the baselines. Surprisingly, the model trained with chain-of-though enhanced prompts dropped performance. We hypothesize that this may be due to the poor explanations provided by the LLaMA-7B model in a zero-shot fashion, as exemplified in the appendix **??**, where we compare the low quality explaination returned by LLaMA, to one generated by ChatGPT [47] for two random samples. Finally, our fine-tuned model achieves similar results to the few-shot with dynamic examples method, but achieves the highest overall result of 0,4077 when used with query filtering. It is also noticeable that the models with the best results are the ones with highest query yield from Figure 1.

The impact and usefulness of generated queries is further exemplified by the models in the last group of the Table 1. The model trained with real queries only was trained in the same conditions as the previous models, but with approximately 16 times more data. The performance is comparable with the previous models trained with much less data, and we saw small improvements when further training it with 40000 filtered queries generated by the two best methods.

Overall, we can conclude that our methods generate queries which quality is comparable to real queries, since performance does not drop when training with them. The appendix A.3.1 shows examples of queries generated by the multiple models.

Query Expansion

For the query expansion methods, results are available for the TREC 2021 task in Table 2 and for the TREC 2022 task in Table 3. Observing the results, we can state that RRF is a successful method of ensembling results for this task, albeit requiring a quality underlying model to obtain good results. Our baselines models, T5-FirstP and T5-LongP, are only beneficial for the R@1000 measurement (which is useful for first stage retrieval), but detrimental otherwise when performing RRF, attesting that out-of-domain training might not be useful for this task. LLaMA, LLaMA-LoRA-Qgen and medAlpaca-7B yielded substantial improvements in all metrics, justifying the usage of RRF.

Analyzing the LLaMA and LLaMA-LoRA-Qgen results, fine-tuning LLaMA actually obtained negative results, begging to question that a general training on the MARCO dataset might not be that well-suited for the task at hand, even if containing a subset of medical documents. As such, we resorted to medAlpaca-7B, which was trained following the same setup as LLaMA-LoRA-Qgen, but considering data with medical information from multiple sources, and all metrics improved.

RSVQA Model Pretraining

The results for the task of RSVQA-HR are available in Table 4. The VQA task was tackled under a classification task, with each label corresponding to a possible answer, and using accuracy as metric. The results from the original paper [39] and of a SOTA work [4], that uses two Transformer decoders with cross-attention from CLIP features [55], are also displayed.

First, we train only on the target dataset ("from scratch"), without pretraining. Then, the T5-FirstP model was used to generate 10 questions per caption for a total of 50 questions per image. Afterward, we choose to apply the QA model for filtering out unanswerable questions, given the captions are small in size. This leaves a total of 20 ± 5 questions per image. From these, only 10 questions were selected in a way that maximizes the number of possible different answers. Finally, due to the high value of unique possible answers (28,000), the top 6k possible answers were selected which constitutes 95% of the original answers. Pretraining on these generated questions achieved a final performance of 85.01% in RSVQA-HR compared to the original performance of 84.85%.

Due to the high computational cost of LLaMa, only 2 questions were generated per caption for a total of 10 questions per image. Corresponding answers were obtained by the QA model but they were not used for filtering, given the already small sample. Only the top 95% of the answer space was kept, in the same manner. The final performance obtained was 85.06% which achieves the highest performance of all models. It should also be noted that the baseline was highly optimized, with a complex hyperparameter search, which wasn't performed in the new models.

5 Conclusion and Future Work

In this work we studied query generation, comparing prompting techniques and fine-tuning of LLMs. We surpass baseline generation methods in the re-ranking tasks, showing that the quality of generated queries is comparable to real queries. Furthermore, we conclude that for the RSVQA task we achieve comparable results to much more complex state-of-the-art models. Finally, we show the effectiveness of the RRF technique, and conclude that in-domain training is necessary for the generation models.

As for future work, we select a few possible directions. First, we can simply use larger versions of LLaMA to get better results, for instance, preliminary tests show that LLaMA-65B generates much better explanations than the 7B version, which could potentially improve results overall, especially for the "Sample + CoT" method. Moreover, we can work on efficiency and study the novel QLoRA [12] approach, a method similar to LoRA which quantizes the model to 4-bit. Other filtering methods can also be consider, i.e., prompt LLaMA to say whether or not a document is relevant to a query. This is in line with recent work that considers a single LLM to perform every task within a pipeline [66].

The MARCO dataset has a large set of relevance judgments. However, recent retrieval datasets such as ClueWeb2022 [49] still do not have relevancy judgments. Hence, generating a query collection can be useful, and models trained with such queries can be compared with current anchor-text versions [79].

For the RSVQA task, tackling the VQA as a classification task might hinder the benefits of using synthetic text data, which motivates the use of generative models. The filtering mechanism seems to improve performance which could help improve the results on LLaMa zero-shot question generation, together with the usage of few-shot prompting.

Finally, for the TREC-CT task, we hypothesize that LLaMA-7B would have to be trained on a subset of the MARCO dataset containing only medical documents, and further prompt engineering research would have to be conducted, as the TREC downstream task is quite unique, intending to generate queries from queries (patient descriptions) and not documents which is the most common scenario.

Table 1: Results on MARCO for the multiple approaches (MRR), with and without filtering.

			MARCO Document Dev	
Retriever	Reranker	Fine-tune Method	MRR@100 (All Queries)	MRR@100 (Filtered Queries)
ANCE-MaxP	MonoT5	-	0.3865	-
ANCE-MaxP	MonoT5	RQ (30k, 3 negs)	0.4016	-
ANCE-MaxP	MonoT5	RQ + T5-FirstP	0.4027	-
ANCE-MaxP	MonoT5	RQ + LLaMA Vanilla	0.4005	0.4034
Ance-MaxP	MonoT5	RQ + LLaMA GBQ	0.3991	0.4010
ANCE-MaxP	MonoT5	RQ + LLaMA Sampled	0.4054	0.4049
ANCE-MaxP	MonoT5	RQ + LLaMA Sampled + CoT	0.3925	0.3980
ANCE-MaxP	MonoT5	RQ + QG-LLaMA-LoRA	0.4035	**0.4077**
ANCE-MaxP	MonoT5	RQ (200k, 10 negs)	0.4176	-
ANCE-MaxP	MonoT5	RQ + LLaMA Sampled (40k)	-	0.4180
ANCE-MaxP	MonoT5	RQ + LLaMA Sampled + CoT (40k)	-	**0.4183**

Table 2: Results on TREC 2021.

Model	P@10	NCDG@10	MRR	R-Precision	R@1000
BM_{25}	0.1640	0.2901	0.3125	0.0927	0.261
BM_{25} + RM_3	0.2053	0.3498	0.3537	0.1338	0.4494
"" + "" + $RRF_{\mathrm{T5-FirstP}}$	0.1307	0.2431	0.2604	0.1008	0.4683
"" + "" + $RRF_{\mathrm{T5-LongP}}$	0.1400	0.2449	0.2677	0.1040	0.5026
"" + "" + RRF_{LLaMA}	0.1960	0.3624	0.4253	0.1447	0.5189
"" + "" + $RRF_{\mathrm{LLaMA-LoRA-Ggen}}$	0.1871	0.3344	0.3922	0.1317	0.5199
"" + "" + $RRF_{\mathrm{medAlpaca-7B}}$	**0.1963**	**0.3645**	**0.4330**	**0.1581**	**0.5427**

Table 3: Results on TREC 2022.

Model	P@10	NCDG@10	MRR	R-Precision	R@1000
BM_{25}	0.1900	0.2700	0.3209	0.0971	0.3049
BM_{25} + RM_3	0.2240	0.3306	0.3404	0.1574	0.4201
"" + "" + $RRF_{\mathrm{T5-FirstP}}$	0.1140	0.1801	0.2083	0.0971	0.4191
"" + "" + $RRF_{\mathrm{T5-LongP}}$	0.1240	0.1939	0.2143	0.0973	0.4918
"" + "" + RRF_{LLaMA}	0.2580	0.3725	0.4302	0.1888	0.5383
"" + "" + $RRF_{\mathrm{LLaMA-LoRA-Ggen}}$	0.2480	0.3662	0.3783	0.1780	0.5601
"" + "" + $RRF_{\mathrm{medAlpaca-7B}}$	**0.3221**	**0.4542**	**0.4869**	**0.2227**	**0.6368**

Table 4: Results on RSVQA-HR dataset.

Model	Accuracy
ResNet + LSTM[39]	79.08
CLIP+Cross-Attention [4]	**85.30**
From scratch	84.85
T5-FirstP	85.01
LLaMa-Alpaca-Zero-Shot	**85.06**

Author Contribution

- Artur Guimarães: Report Writting, Code-base contributions, TREC-CT experiments.
- João Silva: Report Writting, Code-base contributions, RSVQA experiments.
- João Coelho: Report Writting, Code-base contributions, MARCO experiments.

References

[1] S. Agrawal, C. Zhou, M. Lewis, L. Zettlemoyer, and M. Ghazvininejad. In-context Examples Selection for Machine Translation. *ArXiv*, abs/2212.02437, 2022.

[2] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. Do not have enough data? deep learning to the rescue! In *Conference on Artificial Intelligence*, 2020.

[3] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al. Palm 2 technical report. *ArXiv*, abs/2305.10403, 2023.

[4] Y. Bazi, M. M. Al Rahhal, M. L. Mekhalfi, M. A. Al Zuair, and F. Melgani. Bi-modal transformer-based approach for visual question answering in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

[5] L. H. Bonifacio, H. Abonizio, M. Fadaee, and R. F. Nogueira. InPars: Data Augmentation for Information Retrieval using Large Language Models. *ArXiv*, abs/2202.05144, 2022.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Conference on Neural Information Processing Systems*, 2020.

[7] C. Chappuis, V. Zermatten, S. Lobry, B. Le Saux, and D. Tuia. Prompt-RSVQA: Prompting visual context to a language model for remote sensing visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1372–1381, 2022.

[8] Q. Cheng, H. Huang, Y. Xu, Y. Zhou, H. Li, and Z. Wang. NWPU-Captions Dataset and MLCA-Net for Remote Sensing Image Captioning. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–19, 2022.

[9] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.

[10] N. Craswell. Mean reciprocal rank. In *Encyclopedia of Database Systems*. 2009.

[11] X. Cui, V. Goel, and B. Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477, 2015.

[12] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. *ArXiv*, abs/2305.14314, 2023.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Human Language Technologies, NAACL-HLT*, 2019.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[15] S. Diao, P. Wang, Y. Lin, and T. Zhang. Active prompting with chain-of-thought for large language models. *ArXiv*, abs/2302.12246, 2023.

[16] M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. *ArXiv*, abs/1705.00440, 2017.

[17] P. Fernandes, A. Madaan, E. Liu, A. Farinhas, P. H. Martins, A. Bertsch, J. G. C. de Souza, S. Zhou, T. Wu, G. Neubig, and A. F. T. Martins. Bridging the gap: A survey on integrating (human) feedback for natural language generation, 2023.

[18] M. Gospodinov, S. MacAvaney, and C. Macdonald. Doc2query-: When less is more. In *Advances in Information Retrieval - European Conference on Information*, 2023.

[19] M. Guo, J. Ainslie, D. C. Uthus, S. Ontañón, J. Ni, Y. Sung, and Y. Yang. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL*, pages 724–736, 2022.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv*, abs/1207.0580, 2012.

[21] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[22] HuggingFace Transformers. FirstP-T5 for Query Generation. `https://huggingface.co/jmvcoelho/t5-base-msmarco-squad-query-generation-firstp-v2`, . Accessed: 2023-04-27.

[23] HuggingFace Transformers. LLaMA for Query Generation. `https://huggingface.co/jmvcoelho/llama-lora-msmarco-squad-query-generation`, . Accessed: 2023-05-24.

[24] HuggingFace Transformers. LongP-T5 for Query Generation. `https://huggingface.co/jmvcoelho/t5-base-msmarco-squad-query-generation-longp-v2`, . Accessed: 2023-04-27.

[25] HuggingFace Transformers. Medalpaca for Query Generation. `https://huggingface.co/medalpaca`, . Accessed: 2023-05-09.

[26] HuggingFace Transformers. MonoT5. `https://https://huggingface.co/castorini/monot5-base-msmarco-10k`, . Accessed: 2023-04-27.

[27] HuggingFace Transformers. RoBERTA-base-squad2. `https://huggingface.co/deepset/roberta-base-squad2`, . Accessed: 2023-04-27.

[28] N. A. Jaleel, J. Allan, W. B. Croft, F. Diaz, L. S. Larkey, X. Li, M. D. Smucker, and C. Wade. Umass at TREC 2004: Novelty and HARD. In *Thirteenth Text REtrieval Conference, TREC 2004*, 2004.

[29] V. Jeronymo, L. H. Bonifacio, H. Abonizio, M. Fadaee, R. de Alencar Lotufo, J. Zavrel, and R. F. Nogueira. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *ArXiv*, abs/2301.01820, 2023.

[30] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *ArXiv*, abs/1702.08734, 2017.

[31] S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *ArXiv*, abs/1805.06201, 2018.

[32] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019.

[33] X. Li and X. Qiu. Finding Supporting Examples for In-Context Learning. *ArXiv*, abs/2302.13539, 2023.

[34] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137, 2020.

[35] J. Lin, X. Ma, S. Lin, J. Yang, R. Pradeep, and R. Nogueira. Pyserini: An easy-to-use python toolkit to support replicable IR research with sparse and dense representations. *ArXiv*, abs/2102.10073, 2021.

[36] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What Makes Good In-Context Examples for GPT-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL*, 2022.

[37] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9), 2023.

[38] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692, 2019.

[39] S. Lobry, D. Marcos, J. Murray, and D. Tuia. RSVQA: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8555–8566, 2020.

[40] S. MacAvaney, A. Cohan, and N. Goharian. SLEDGE: a simple yet effective baseline for COVID-19 scientific knowledge search. *ArXiv*, abs/2005.02365, 2020.

[41] Y. Meng, J. Huang, Y. Zhang, and J. Han. Generating training data with language models: Towards zero-shot language understanding. *ArXiv*, abs/2202.04538, 2022.

[42] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches*, 2016.

[43] R. Nogueira and K. Cho. Passage Re-ranking with BERT. *ArXiv*, abs/1901.04085, 2019.

[44] R. Nogueira and J. Lin. From doc2query to doctttttquery. *Technical Report*, 6, 2019.

[45] R. F. Nogueira, W. Yang, J. Lin, and K. Cho. Document Expansion by Query Prediction. *ArXiv*, abs/1904.08375, 2019.

[46] R. F. Nogueira, Z. Jiang, R. Pradeep, and J. Lin. Document ranking with a pretrained sequence-to-sequence model. In *Empirical Methods on Natural Language Processing*, 2020.

[47] OpenAI. ChatGPT. `https://openai.com/research/chatgpt`. Accessed: 2023-05-25.

[48] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[49] A. Overwijk, C. Xiong, X. Liu, C. VandenBerg, and J. Callan. Clueweb22: 10 billion web documents with visual and semantic information. *ArXiv*, abs/2211.15848, 2022.

[50] R. K. Pasumarthi, S. Bruch, X. Wang, C. Li, M. Bendersky, M. Najork, J. Pfeifer, N. Golbandi, R. Anil, and S. Wolf. TF-Ranking: Scalable TensorFlow Library for Learning-to-Rank. In *International Conference on Knowledge Discovery & Data Mining*, 2019.

[51] R. Pradeep, R. Nogueira, and J. Lin. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. *ArXiv*, abs/2101.05667, 2021.

[52] R. Pradeep, Y. Li, Y. Wang, and J. Lin. Neural Query Synthesis and Domain-Specific Ranking Templates for Multi-Stage Clinical Trial Matching. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2325–2330, 2022.

[53] Y. Qu, Y. Ding, J. Liu, K. Liu, R. Ren, W. X. Zhao, D. Dong, H. Wu, and H. Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, 2021.

[54] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, Proceedings of Machine Learning Research, 2021.

[56] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140), 2020.

[57] M. M. A. Rahhal, Y. Bazi, S. O. Alsaleh, M. Al-Razgan, M. L. Mekhalfi, M. A. Zuair, and N. Alajlan. Open-ended remote sensing visual question answering with transformers. *International Journal of Remote Sensing*, 43(18):6809–6823, 2022.

[58] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *Empirical Methods in Natural Language Processing*, 2016.

[59] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Empirical Methods in Natural Language Processing*, 2019.

[60] K. Roberts, D. Demner-Fushman, E. M. Voorhees, S. Bedrick, and W. R. Hersh. Overview of the trec 2021 clinical trials track. In *Proceedings of the Thirtieth Text REtrieval Conference*, 2021.

[61] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 2009.

[62] T. Schick and H. Schütze. Few-shot text generation with natural language instructions. In *Empirical Methods in Natural Language Processing*, pages 390–402, 2021.

[63] P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 2018.

[64] J. D. Silva, J. Magalhães, D. Tuia, and B. Martins. Remote sensing visual question answering with a self-attention multi-modal encoder. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, page 40–49, 2022.

[65] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *ArXiv*, abs/2104.09864, 2021.

[66] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agent. *ArXiv*, abs/2304.09542, 2023.

[67] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *ArXiv*, abs/1908.07490, 2019.

[68] M. Tan and Q. Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106, 2021.

[69] N. Thakur, N. Reimers, J. Daxenberger, and I. Gurevych. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[70] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971, 2023.

[71] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. In *Conference on Neural Information Processing Systems*, 2017.

[73] B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, 2021.

[74] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-Instruct: Aligning Language Model with Self Generated Instructions. *ArXiv*, abs/2212.10560, 2022.

[75] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao. Simvlm: Simple visual language model pretraining with weak supervision. *ArXiv*, abs/2108.10904, 2021.

[76] J. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *ArXiv*, abs/1901.11196, 2019.

[77] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, and D. Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv*, abs/2201.11903, 2022.

[78] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: when to warp? In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pages 1–6, 2016.

[79] Y. Xie, X. Liu, and C. Xiong. Unsupervised dense retrieval training with web anchors. *ArXiv*, abs/2211.15848, 2023.

[80] L. Xiong, C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.

[81] A. Yates, R. Nogueira, and J. Lin. Pretrained transformers for text ranking: BERT and beyond. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[82] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. *ArXiv*, abs/2205.01917, 2022.

[83] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al. Florence: A new foundation model for computer vision. *ArXiv*, abs/2111.11432, 2021.

[84] Z. Yuan, L. Mou, Q. Wang, and X. X. Zhu. From easy to hard: Learning language-guided curriculum for visual question answering on remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

[85] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen. A survey of large language models, 2023.

[86] X. Zheng, B. Wang, X. Du, and X. Lu. Mutual attention inception network for remote sensing visual question answering. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.

[87] H. Zhuang, Z. Qin, R. Jagerman, K. Hui, J. Ma, J. Lu, J. Ni, X. Wang, and M. Bendersky. RankT5: Fine-Tuning T5 for Text Ranking with Ranking Losses. *ArXiv*, abs/2210.10634, 2022.

A Appendix

A.1 Model Training Setups

A.1.1 Baseline T5s

The baseline T5s follow a standard T5 seq2seq training. The models were trained with a batch size of 8, 8 gradient accumulation steps, and a learning rate of 3e-4 with a linear scheduler and 400 warm-up steps. The following prompt was used, where d is the input document:

T5 Prompt

"**Generate query**: d. **Query**:"

A.1.2 LLaMA-LoRA-Qgen

For the fine-tuned LLaMA model, we resort to low-rank adaptation, considering only 006% of the parameters as trainable (approximately 4 million of the 7 billion of the original model). We use a batch size of 32, 5 steps of gradient accumulation, and a learning rate of 3e-4. LoRA parameters, r, α and dropout, were set to 8, 16 and 5%, respectively. We used a variation of instruction-based prompt used by AlpacaLoRA, where d is an input document:

LLaMA-LoRA-QGen Prompt

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: Consider a query that can be posed to a search engine. Generate such a query, given that the following document should be returned as very relevant.

Input: d

Response:

A.1.3 MonoT5 Re-rankers

The re-ranking models follow the described MonoT5 architecture. We use a learning rate of 3e-4, linear schedule with 200 warm-up steps, 16 batch size, and 8 steps of gradient accumulation. Since

we are using the MonoT5 architecture, we considered a cross-entropy loss between the logits of the first generated token, and the target token, either *true* or *false*, depending on the label of the input query-document pair. The input prompt is the original MonoT5 prompt, where q is the input query and q the input document:

MonoT5 Prompt

Query: q **Document:** d **Relevant:**

A.2 Prompts

A.2.1 InPars Vanilla Prompt

The InPars Vanilla prompt considers fixed examples for few-shot context learning. For an input document d, a prompt with a single example is as follows:

Vanilla Prompt

Document: This a fixed example, the same for all input docs.
Query: This is the corresponding query.
Document: d
Query:

A.2.2 InPars GBQ Prompt

The InPars GBQ prompt also considers fixed examples for few-shot context learning, but adds a bad query to guide generation. For an input document d, a prompt with a single example is as follows:

GBQ Prompt

Document: This a fixed example, the same for all input docs.
Bad Query: This is a fixed bad query.
Good Query: This is the corresponding query.
Document: This is the input document.
Good Query:

A.2.3 Dynamic In-Context Example Prompt

Instead of using fixed examples for all input documents, this approach samples a similar document to guide generation.

Dynamic In-Context Example Prompt

Document: This is an example document that was sampled.
Query: This is the corresponding query.
Document: This is the input document.
Query:

A.2.4 Chain-of-Though Enhanced Prompt

This prompt also considers the usage of sampled examples, but adds previously generated explanations as to why the example document is relevant to the query:

Chain-of-Though Enhanced Prompt

Document: This is an example document that was sampled.
Query: This is the corresponding query.
The document is relevant to the query because (previously generated context)

Document: This is the input document.
Query:

In order to generate the explanation given a document d and a query q, the following prompt was used:

Explanation Prompt

Given the following query: q, a search engine returned the document: d. The document is relevant to the query because

A.2.5 AlpacaLoRA Prompt to Generate Questions from Image Captions

Alpaca was trained with a specific prompt over instruction-following data, which considers an introductory explanation of the task, and the instruction itself to generate the questions from a caption c.

AlpacaLoRA Prompt to Generate Questions from Image Captions

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: You are a powerful language model. You are generating questions that could be asked about an image. Don't forget this question is about an image. Generate a question for the following sentence and make it concise and short.
Input: c
Response:

A.3 Generation Examples

A.3.1 Generated Queries

Question Generation Example 1 (MS-MARCO)

Document: Acclimatization "From Wikipedia, the free encyclopedia (Redirected from Acclimatize)navigation search Not to be confused with Acclamation. Acclimatization or acclimatisation (also called acclimation or acclimatation) is the process in which an individual organism adjusts to a change in its environment (such as a change in altitude, temperature, humidity, photoperiod, or p H), allowing it to maintain performance across a range of environmental conditions. Acclimatization occurs in a short period of time (hours to weeks), and within the organismś lifetime (compared to adaptation, which is a development that takes place over many generations). This may be a discrete occurrence (for example, when mountaineers acclimate to high altitude over hours or days) or may instead represent part of a periodic cycle, such as a mammal shedding heavy winter fur in favor of a lighter summer coat. Organisms can adjust their morphological, behavioral, physical, and/or biochemical traits in response to changes in their environment. (...)
Vanilla Generation: How large is the canadian military?
GBQ Generation: How does acclimation work?
Sampled Generation: acclimatization definition
Sampled + CoT Generation: acclimatization adaptation definition
LLaMA-LoRA-Qgen Generation: what is acclimatization

Question Generation Example 2 (MS-MARCO)

Document: Obstructive Sleep Apnea in Children Obstructive Sleep Apnea in Children PDF PRINT COMMENTSSHAREJAMES CHAN, M. D. Cleveland Clinic Foundation, Cleveland, Ohio JENNIFER C. EDMAN, M. D. Fairview Hospital, Cleveland, Ohio PETER J. KOLTAI, M. D. Cleveland Clinic Foundation, Cleveland, Ohio. Am Fam Physician. 2004 Mar 1;69 (5):1147-1155. Patient Information Handout Obstructive sleep-disordered breathing is common in children. From 3 percent to 12 percent of children snore, while obstructive sleep apnea syndrome affects 1 percent to 10 percent of children. The majority of these children have mild symptoms, and many outgrow the condition. Consequences of untreated obstructive sleep apnea include failure to thrive, enuresis, attention-deficit disorder, behavior problems, poor academic performance, and cardiopulmonary disease. The most common etiology of obstructive sleep apnea is adenotonsillar hypertrophy. (...)
Vanilla Generation: what is obstructive sleep apnea?
GBQ Generation: what is obstructive sleep apnea?
Sampled Generation: what is obstructive sleep apnea in children?
Sampled + CoT Generation: what is a sleep disorder?
LLaMA-LoRA-Qgen Generation: what is the most common cause of obstructive sleep apnea in children?

Question Generation Example (RSVQA)

Caption: Two airplanes and one smaller plane were parked in the open space .
Question: What are the three aircraft in the open space ?
Answer: two airplanes and one smaller plane

Explaination Example 1

Query: does california have a casino?
Document: California Casinos Gambling in California California Casinos Gambling in California California Casinos Average Rating 3.7/5 California Casino Map Top Rated Online Casinos for Californiaup toVisit CasinoRead Review up to test TCs Apply Visit Casino Read Review Loading... There are no available casinos for requested criteria.Something went wrong. Please try again later.Show as List Grid San Manuel Casino24 /7 Operated by: San Manuel Band of Mission Indians San Manuel Casino is located just about an hour from Downtown LA at 777 San Manuel Boulevard in Highland, California. (...)
LLaMA Generation: The document is relevant to the query because it is a casino in California.
ChatGPT Generation: The document is relevant to the query because it mentions San Manuel Casino, which is located in Highland, California. This indicates that there is at least one casino in California.

Explaination Example 2

Query: meaning of context
Document: What does context mean? Vocabulary - What does context mean?Definitions for context kan-tekst context Here are all the possible meanings and translations of the word context. Princetons WordNet (3.40 / 5 votes) Rate this definition:context, linguistic context, context of use (noun)discourse that surrounds a language unit and helps to determine its interpretation context, circumstance, setting (noun)the set of facts or circumstances that surround a situation or event"the historical context Wiktionary (4.00 / 4 votes) Rate this definition:context (Noun) The surroundings, circumstances, environment, background or settings that determine, specify, or clarify the meaning of an event or other occurrence.In what context did your attack on him happen? - We had a pretty tense relationship at the time, and when he insulted me I snapped.Etymology: From contextus. context (Noun)The text in which a word or passage appears and which helps ascertain its meaning.Etymology: From contextus. context (Noun)The surroundings and environment in which an artifact is found and which may provide important

clues about the artifactś function and/or cultural meaning.Etymology: From contextus. context (Noun)The trama or flesh of a mushroom.Etymology: From contextus. Webster Dictionary (2.00 / 2 votes) Rate this definition:Context (adj)knit or woven together; close; firm Etymology: [L. contextus; cf. F. contexte .]Context (noun)the part or parts of something written or printed, as of Scripture, which precede or follow a text. (...)
LLaMA Generation: The document is relevant to the query because it is in the context of the query.
ChatGPT Generation: The document is relevant to the query because the document defines "context" as the surrounding circumstances that determine the meaning of an event or the interpretation of language. It also includes examples and discusses the etymology of the word.