Improving Biomedical Information Retrieval with Neural Retrievers

Man Luo,[1] **Arindam Mitra,**[2] **Tejas Gokhale,**[1] **Chitta Baral**[1]

[1] Arizona State University
[2] Microsoft
mluo26@asu.edu, arindam.mitra2@gmail.com, tgokhale@asu.edu, chitta@asu.edu

Abstract

Information retrieval (IR) is essential in search engines and dialogue systems as well as natural language processing tasks such as open-domain question answering. IR serve an important function in the biomedical domain, where content and sources of scientific knowledge may evolve rapidly. Although neural retrievers have surpassed traditional IR approaches such as TF-IDF and BM25 in standard open-domain question answering tasks, they are still found lacking in the biomedical domain. In this paper, we seek to improve information retrieval (IR) using neural retrievers (NR) in the biomedical domain, and achieve this goal using a three-pronged approach. First, to tackle the relative lack of data in the biomedical domain, we propose a template-based question generation method that can be leveraged to train neural retriever models. Second, we develop two novel pre-training tasks that are closely aligned to the downstream task of information retrieval. Third, we introduce the "Poly-DPR" model which encodes each context into multiple context vectors. Extensive experiments and analysis on the BioASQ challenge suggest that our proposed method leads to large gains over existing neural approaches and beats BM25 in the small-corpus setting. We show that BM25 and our method can complement each other, and a simple hybrid model leads to further gains in the large corpus setting.

1 Introduction

Information retrieval (IR) is widely used in commercial search engines and is an active area of research for natural language processing tasks such as open-domain question answering (ODQA). IR has also become important in the biomedical domain due to the explosion of information available in electronic form (Shortliffe et al. 2014). Biomedical IR has traditionally relied upon term-matching algorithms (such as TF-IDF and BM25 (Robertson and Zaragoza 2009)), which search for documents that contain terms mentioned in the query. For instance, the first example in Table 1 shows that BM25 retrieves a sentence that contains the word *"Soluvia"* from the question. However, term-matching suffers from failure modes, especially for terms which have different meanings in different contexts (example 2), or when crucial semantics from the question are not considered during retrieval (for instance, in the third example when the term "how large" is not reflected in the answer retrieved by BM25).

Since these failure modes can have a direct impact on downstream NLP tasks such as open-domain question answering (ODQA), there has been interest in developing neural retrievers (NR) (Karpukhin et al. 2020). NRs which represent query and context as vectors and utilize similarity scores for retrieval, have led to state-of-the-art performance on ODQA benchmarks such as Natural Questions (Kwiatkowski et al. 2019) and TriviaQA (Joshi et al. 2017). Unfortunately, these improvements on standard NLP datasets are not observed in the biomedical domain with neural retrievers.

Recent work provides useful insights to understand a few shortcomings of NRs. Thakur et al. (2021) find NRs to be lacking at exact word matching, which affects performance in datasets such as BioASQ (Tsatsaronis et al. 2015) where exact matches are highly correlated with the correct answer. Lewis, Stenetorp, and Riedel (2021) find that in the Natural Questions dataset, answers for 63.6% of the test data overlap with the training data and DPR performs much worse on the non-overlapped set than the test-train overlapped set. In this work, we found this overlap to be only 2% in the BioASQ dataset, which could be a potential reason for lower performance of NR methods. We also discovered that NRs produce better representations for short contexts that for long contexts – when the long context is broken down into multiple shorter contexts, performance of NR models improves significantly.

In this paper, we seek to address these issues and improve the performance of neural retrieval beyond traditional methods for biomedical IR. While existing systems have made advances by improving neural re-ranking of retrieved candidates (Almeida and Matos 2020; Pappas, Stavropoulos, and Androutsopoulos 2020), our focus is solely on the retrieval step, and therefore we compare our neural retriever with other retrieval methods. Our method makes contributions to three aspects of the retrieval pipeline – question generation, pre-training, and model architecture.

Our first contribution is the **"Poly-DPR"** model architecture for neural retrieval. Poly-DPR builds upon two recent developments: Poly-Encoder (Humeau et al. 2020) and Dense Passage Retriever (Karpukhin et al. 2020). In DPR, a question and a candidate context are encoded by two models separately into a contextual vector for each, and a score for each context can be computed using vector similarity. On the other hand, Poly-Encoder represents the query by K vectors and produces context-specific vectors for each query. Instead,

Question	Answer	Retrieved Context (BM25)	Retrieved Context(DPR)
What is Soluvia?	Soluvia by Becton Dickinson is a microinjection system for intradermal delivery of vaccines.	The US FDA approved Sanofi Pasteur's Fluzone Intradermal influenza vaccine that uses a new microinjection system for intradermal delivery of vaccines (Soluvia, Becton Dickinson).	Internet-ordered viagra (sildenafil citrate) is rarely genuine.
Is BNN20 involved in Parkinson's disease?	BNN-20 could be proposed for treatment of PD	Rare causes of dystonia parkinsonism.	BNN-20 could be proposed for treatment of PD
How large is a lncRNAs?	lncRNAs are defined as RNA transcripts longer than 200 nucleotides that are not transcribed into proteins.	lncRNAs are closely related with the occurrence and development of some diseases.	An increasing number of long noncoding RNAs (lncRNAs) have been identified recently.

Table 1: Illustrative examples from the BioASQ challenge along with the context retrieved by two methods BM25 and DPR.

our approach Poly-DPR represents each *context* by K vectors and produces *query-specific vectors* for each context. We further design a simple inference method that allows us to employ MIPS (Shrivastava and Li 2014) during inference.

Next, we develop **"Temp-QG"**, a template-based question generation method which helps us in generating a large number of domain-relevant questions to mitigate the train-test overlap issue. TempQG involves extraction of templates from in-domain questions, and using a sequence-to-sequence model (Sutskever, Vinyals, and Le 2014) to generate questions conditioned on this template and a text passage.

Finally, we design two new pre-training strategies: **"ETM"** and **"RSM"** that leverage our generated dataset to pre-train Poly-DPR. These tasks are designed to mimic domain-specific aspects of IR for biomedical documents which contain titles and abstracts, as opposed to passage retrieval from web pages (Chang et al. 2020). Our pre-training tasks are designed to be used for long contexts and short contexts. In both tasks, we utilize keywords in either query or context, such that the capacity of neural retrievers to match important keywords can be improved during training.

Armed with these three modules, we conduct a comprehensive study of document retrieval for biomedical texts in the BioASQ challenge. Our analysis demonstrates the efficacy of each component of our approach. Poly-DPR outperforms BM25 and previous neural retrievers for the BioASQ challenge, in the small-corpus setting. A hybrid method, which is a simple combination of BM25 and NR predictions, leads to further improvements. We perform a post-hoc error analysis to understand the failures of BM25 and our Poly-DPR model. Our experiments and analysis reveal aspects of biomedical information retrieval that are not shared by generic open-domain retrieval tasks. Findings and insights from this work could benefit future improvements in both term-based as well as neural-network based retrieval methods.

2 Related Work

Neural Retrievers aim to retrieve relevant context from a large corpus given a query. NRs can be clubbed into two architectural families – cross-attention models (Nogueira and Cho 2019; MacAvaney et al. 2019; Yang et al. 2019), and dual-encoder models which employ separate encoders to encode the query and context (Karpukhin et al. 2020; Chang et al. 2020). The cross-attention model requires heavy computation and can not be directly used in a large corpus setting, while dual-models can allow pre-computation of context representations and the application of efficient search methods such as MIPS (Shrivastava and Li 2014) during inference. To take advantage of both models, Poly-Encoder (Humeau et al. 2020) uses K representations for each query and an attention mechanism to get context-specific query representations. ColBERT (Khattab and Zaharia 2020) extends the dual-encoder architecture by performing a token-level interaction step over the query and context representations, but requires significant memory for large corpora (Thakur et al. 2021).

Pre-training Tasks for NR. Masked language modeling (MLM) and next-sentence prediction introduced in BERT (Devlin et al. 2019) have led to a paradigm shift in the training of neural network models for multiple NLP tasks. For text retrieval, pre-training tasks that are more aligned with the retrieval task have been developed. Chang et al. (2020) propose Body First Selection (BFS), and Wiki Link Prediction (WLP) for document retrieval. Lee, Chang, and Toutanova (2019) propose an Inverse Cloze Task (ICT) task in which a random sentence drawn from a passage acts as a query and the remaining passage as a relevant answer. Guu et al. (2020) show that ICT effectively avoids the cold-start problem.

Question Generation (QG) methods have become sophisticated due to the advances in sequence-to-sequence modeling (Sutskever, Vinyals, and Le 2014); QG is considered an auxiliary pre-training task for question answering models (Alberti et al. 2019). One set of QG methods can be categorized as 'Answer-Aware' QG (Du and Cardie 2018; Zhao et al. 2018; Dong et al. 2019), in which an answer extraction model first produces potential answers, followed by a question generator which generates a question given the context and a potential answer. Alberti et al. (2019) utilizes cycle consistency to verify whether a question-answering model predicts the same answer to the generated question. A second set of QG methods generate questions without conditioning the generator using the answer – for instance, Lopez et al. (2020) propose end-to-end question generation

based on the GPT-2 model, while Lewis, Denoyer, and Riedel (2019); Fabbri et al. (2020); Banerjee, Gokhale, and Baral (2021) generate questions using linguistic and semantic templates. Question paraphrasing (Hosking and Lapata 2021) is a related approach for creating augmented training samples. Question generation has also been explored in visual question answering, with end-to-end methods (Li et al. 2018; Krishna, Bernstein, and Fei-Fei 2019) and template-based methods (Banerjee et al. 2021). While our proposed question generation method is also template-based, instead of using a pre-defined list of templates designed by humans, our template extraction process is automated.

3 Poly-Dense Passage Retriever

3.1 Preliminaries

Dense Passage Representation (DPR) (Karpukhin et al. 2020) is a neural retriever model belonging to the dual-model family. DPR encodes the query q and the context c into dense vector representations:

$$v_q = E_q(q)\texttt{[CLS]}, \quad v_c = E_c(c)\texttt{[CLS]}. \tag{1}$$

where E_q and E_c are BERT (Devlin et al. 2019) models which output a list of dense vectors $(h_1, \ldots, h_n)$ for each token of the input, and the final representation is the vector representation of special token `[CLS]`. E_q and E_c are initialized identically and are updated independently while being trained with the objective of minimizing the negative log likelihood of a positive (relevant) context. A similarity score between q and each context c is calculated as the inner product between their vector representations:

$$\text{sim}(q, c) = v_q^T v_c. \tag{2}$$

Poly-Encoder (Humeau et al. 2020) also uses two encoders to encode query and context, but the query is represented by K vectors instead of a single vector as in DPR. Poly-Encoder assumes that the query is much longer than context, which is in contrast to information retrieval and open-domain QA tasks in the biomedical domain, where contexts are long documents and queries are short and specific.

3.2 Poly-DPR: Poly-Dense Passage Retriever

We integrate Poly-Encoder and DPR to use K vectors to represent context rather than query. In particular, the context encoder includes K global features $(m_1, m_2, \cdots, m_k)$, which are used to extract representation v_c^i, $\forall i \in \{1 \cdots k\}$ by attending over all context tokens vectors.

$$v_c^i = \sum_n w_n^{m_i} h_n, \quad \text{where} \tag{3}$$

$$(w_1^{m_i} \ldots, w_n^{m_i}) = \text{softmax}(m_i^T \cdot h_1, \ldots, m_i^T \cdot h_n). \tag{4}$$

After extracting K representations, a query-specific context representation $v_{c,q}$ is computed by using the attention mechanism:

$$v_{c,q} = \sum_k w_k v_c^k, \quad \text{where} \tag{5}$$

$$(w_1, \ldots, w_k) = \text{softmax}(v_q^T \cdot v_c^1, \ldots, v_q^T \cdot v_c^k). \tag{6}$$

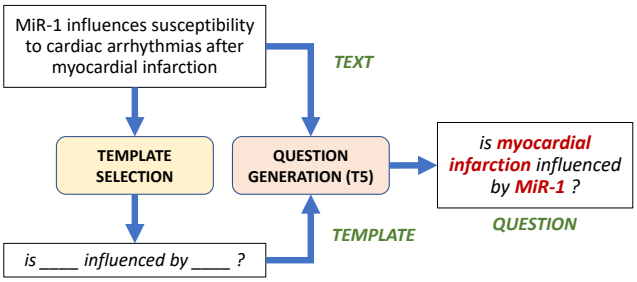

Figure 1: Overview of Template-Based Question Generation.

Although we can pre-compute K representations for each context in the corpus, during inference, a ranking of the context needs to be computed after obtaining all query-specific context representations. As such, we can not directly use efficient algorithms such as MIPS (Shrivastava and Li 2014). To address this challenge, we use an alternative similarity function for inference – the score $\text{sim}_{\text{infer}}$ is computed by obtaining K similarity scores for the query and each of the K representations, and take the maximum as the similarity score between context and query:

$$\text{sim}_{\text{infer}}(q, c) = \max(v_q^T \cdot v_c^1, \ldots, v_q^T \cdot v_c^k). \tag{7}$$

Using this similarity score, we can take advantage of MIPS to find the most relevant context to a query.

In sum, Poly-DPR differs from Poly-Encoder in two major aspects: (1) K pre-computed representations of context as opposed to K representations computed during inference, and (2) a faster similarity computation during inference.

3.3 Hybrid Model

In this paper, we also explore a hybrid model that combines the traditional approach of BM25 and neural retrievers. We first retrieve the top-100 candidate articles using BM25 and a neural retriever (Poly-DPR) separately. The scores produced by these two methods for each candidate are denoted by S_{BM25} and S_{NR} respectively and normalized to the $[0, 1]$ range to obtain S'_{BM25} and S'_{NR}. If a candidate article is not retrieved by a particular method, then its score for that method is 0. For each article, we get a new score:

$$S_{\text{hybrid}} = S'_{\text{BM25}} + S'_{\text{NR}}. \tag{8}$$

Finally, we re-rank candidates based on S_{hybrid} and pick the top candidates – for BioASQ performance is evaluated on the top-10 retrieved candidates.

4 Template Based Question Generation

We propose a template-based question generation approach – *TempQG*, that captures the style of the questions in the target domain. Our method consists of three modules: template extraction, template selection, and question generation.

Template Extraction aims to extract unique templates from which the questions in the training set can be generated. We first use bio-entity taggers from Spacy (https://spacy.io/) to obtain a set of entities from the question. We replace non-verb entities having a document frequency less than k with an

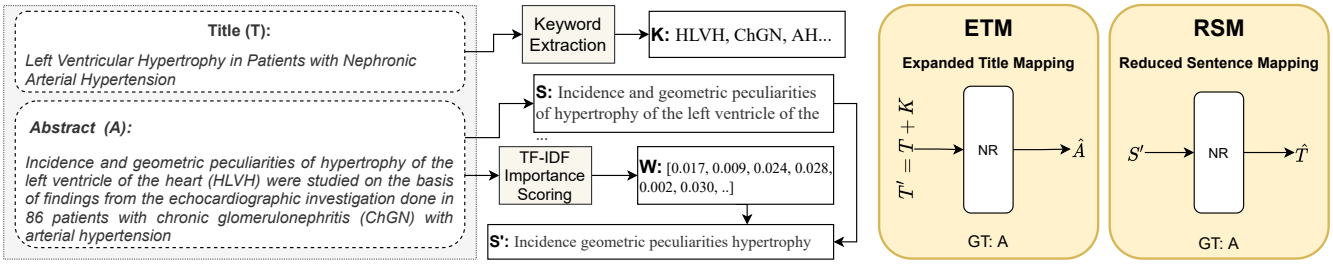

Figure 2: Poly-DPR is pre-trained on two novel tasks designed specifically for information retrieval applications. This figure illustrates the sample generation pipeline using the title and abstract from each sample in BioASQ.

underscore (_) – this prevents common entities such as "disease", "gene" from being replaced. For e.g., given the question *"Borden classification is used for which disease?"*, the entity tagger returns *["Borden classification", "disease"]*, but only the first entity clears our frequency-based criteria. As a result, the generated template is *"_ is used for which disease?"*. This process gives us a preliminary list of templates. We then use a question similarity model (which returns a score between $[0, 1]$) to compute the pairwise score between all templates. Templates are assigned to a cluster if they have a minimum similarity of 0.75 with existing templates of a cluster. Once clusters have been formed, we choose either the sentence with the smallest or second-smallest length as the representative template. These representative templates are used for question generation.

Template Selection. Given a text passage, we create a text-template dataset and train the PolyDPR architecture to retrieve a relevant template. After the model is trained, we feed new text inputs to the model, obtain query encoding and compute the inner product with each template. Templates with maximum inner product are selected to be used for QG.

Question Generation (QG). We use a T5 (Raffel et al. 2020) model for generating questions, by using text and template as conditional inputs. To distinguish between these two inputs, we prepend each with the word "template" or "context", resulting in an input of the form: $\{$"$template$" : $template$, "$context$" : $text\}$. Figure 1 shows an illustrative example for the template-based question generation method abbreviated as *TempQG*. The context used for generating the questions are any two consecutive sentences in the abstract. Given such a context, we first select 10 unique templates and concatenate each template with the context independently. These are used by the question generation model to produce 10 initial questions; duplicate questions are filtered out.

5 Pre-training for Neural Retrieval

Our aim is to design pre-training tasks specifically for the biomedical domain since documents in this domain bear the <*title, abstract, main text*> structure of scientific literature. This structure is not commonly found in documents such as news articles, novels, and text-books. Domain-specific pre-training tasks have been designed by Chang et al. (2020) for Wikipedia documents which contains hyperlinks to other Wikipedia documents. However most biomedical documents do not contain such hyperlinks, and a such, pre-training strategies recommended by Chang et al. (2020) are incompatible with structure of biomedical documents.

Therefore, we propose Expanded Title Mapping (ETM) and Reduced Sentence Mapping (RSM), designed specifically for biomedical IR, to mimic the functionality required for open-domain question answering. An overview is shown in Figure 2. The proposed tasks work for both short as well as long contexts. In biomedical documents, each document has a title (T) and an abstract (A). We pre-train our models on ETM or RSM and then finetune them for retrieval.

Expanded Title Mapping (ETM). For ETM, the model is trained to retrieve an abstract, given an extended title T' as a query. T' is obtained by extracting top-m keywords from the abstract based on the TF-IDF score, denoted as $K = \{k_1, k_2, \cdots, k_m\}$, and concatenating them with the title as: $T' = \{T, k_1, k_2, \cdots, k_m\}$. The intuition behind ETM is to train the model to match the main topic of a document (keywords and title) with the entire abstract.

Reduced Sentence Mapping (RSM). RSM is designed to train the model to map a sentence from an abstract with the extended title T'. For a sentence S from the abstract, we first get the weight of each word $W = \{w_1, w_2, \cdots, w_n\}$ by the normalization of TF-IDF scores of each word. We then reduce S to S' by selecting the words with the top-m corresponding weights. The intuition behind a reduced sentence is to simulate a real query which usually is shorter than a sentence in a PubMed abstract. Furthermore, S' includes important words based on the TF-IDF score, which is similar to a question including keywords.

6 Experiments

Dataset. We focus on the document retrieval task in BioASQ8 (Tsatsaronis et al. 2015) with a goal of retrieving a list of relevant documents to a question. This dataset contains 3234 questions in the training set and five test sets *(B1, B2, B3, B4, B5)* with 100 questions each. Each question is equipped with a list of relevant documents and a list of relevant snippets of the documents.

Baselines. We compare our work with BM25 and DPR in the short corpus setting, and BM25 and GenQ (Ma et al.

2021) in the large corpus setting. Note that the same test sets (B1-5) are used for evaluating both settings. We also compare an alternative question generation method AnsQG (Chan and Fan 2019) in which an answer extraction model first extracts an answer from a context and a question generation model uses the answer as well as the text to generate a question. Similarly, we compare our method with an existing pre-training task ICT (Lee, Chang, and Toutanova 2019). Retrieval systems for the BioASQ task typically follow a two-step process: retrieval of candidates and re-ranking. The focus of this paper is on improving the former, and thus we use different retrieval methods as baselines and do not compare with state-of-the-art systems that use various re-ranking methods. We use Mean Average Precision (MAP) as our evaluation metric.

6.1 Experimental Settings

Size of Corpus. PubMed is a large corpus containing 19 million articles, each with a title and an abstract. Due to this large corpus size, indexing the entire corpus takes a significantly long time. To conduct comprehensive experiments and to efficiently evaluate the impacts of each proposed method, we construct a small corpus with $133,084$ articles in total: $33,084$ articles belonging to the training and test sets of BioASQ8, and an additional $100K$ articles that are randomly sampled from the entire corpus.

Length of Context. We use two context lengths for training neural retrievers and indexing the corpus: 128 (short) and 256 (long). We use RSM as the pre-training task for short contexts and either ETM or ICT with long contexts.

Training Setup. We use BioBERT (Lee et al. 2020) as the initial model for both query and context encoders in all experiments. For BM25, we use an implementation from Pyserini (Lin et al. 2021) with default hyperparameters k=0.9 and b=0.4. We also try k=1.2 and b=0.75 as used by Ma et al. (2021) and find the default setting to be slightly better. For Poly-DPR, the number of representations K is set as 6 after a hyper-parameter search. While larger values of K improve results, it makes indexing slower [1].

6.2 Results

Effect of Pre-Training Tasks and Fine-Tuning Datasets. Table 2 shows results when Poly-DPR is trained with different methods of pre-training and different fine-tuning datasets. Both RSM and ETM lead to improvements even when the finetuning task has only a limited amount of supervised data, i.e. BioASQ. When compared to Poly-DPR trained without any pre-training, RSM improves by $\sim$9% and ETM by $\sim$18%. ETM is better than the existing pre-training method ICT (Lee, Chang, and Toutanova 2019) by $\sim$2%. When the size of fine-tuning set is large, i.e. with our question generation method (TempQG), the gains due to pretraining are higher with short contexts than with large contexts. We believe this to be a result of the finetuning dataset in the long-context setting being significantly larger than the pre-training dataset, thereby having a larger effect on the training process[1].

[1] see Appendix of https://arxiv.org/abs/2201.07745 for details.

CL	PT	FT	B1	B2	B3	B4	B5	Avg.
Short (128)	-	B	54.48	50.51	53.8	59.06	48.71	53.31
	-	T	62.92	58.79	62.94	70.30	63.39	63.67
	RSM	B	**65.94**	57.43	61.89	69.01	58.23	62.50
	RSM	A	56.84	55.79	57.52	58.68	55.15	56.80
	RSM	T	64.71	**64.92**	**64.28**	**73.11**	**66.29**	**66.66**
Long (256)	-	B	35.69	32.66	32.26	38.28	30.87	33.95
	-	T	63.95	**59.51**	62.98	66.71	62.80	63.19
	ICT	B	54.44	47.37	52.61	53.69	44.38	50.50
	ETM	B	56.63	46.63	52.79	56.97	49.61	52.53
	ETM	T	64.57	58.51	**64.02**	68.44	62.60	**63.62**
	ETM	A	54.44	49.95	48.42	58.15	52.60	52.71
	ICT+ETM	B	51.33	49.43	49.36	53.19	43.58	49.38
	ICT+ETM	T	**64.93**	58.49	60.18	**69.42**	**64.87**	63.58

Table 2: Effect of pre-training tasks (PT) and fine-tuning datasets (B: BioASQ, T: TempQG and A: AnsQG) on the performance of Poly-DPR with two context lengths (CL) on the BioASQ small corpus test set. Bi stands for the i^{th} batch in the testing sets.

We also see that when Poly-DPR is only trained on BioASQ, the performance with small contexts is much better than with long contexts (53.31% vs 33.95%). This suggests that Poly-DPR trained on the small corpus finds it difficult to produce robust representations for long contexts. On the other hand, the performance of Poly-DPR variants trained on TempQG is close for short and long contexts, which suggests that large-scale relevant training data improves representations.

Comparison with Baselines. Table 3 shows a comparison between baselines and our best model (Poly-DPR with short context (128) pre-trained with RSM and finetuned on TempQG). Note that our model is only trained on datasets acquired from the small corpus. However, we evaluate the same model on the large corpus test set.

In the **small corpus setting**, it can be seen that our model outperforms all existing methods in the small corpus setting, and is better than DPR by 13.3% and 20.8% in short (128) and long (256) context lengths respectfully. In the **large corpus setting**, our method is better than GenQ (Ma et al. 2021) on all five test sets. This shows that our method, which uses 10 million generated samples is better than GenQ which uses 83 million samples for training, thus showing the effectiveness of our template-based question generation method. Although our method performs better than BM25 on B1, B2, B5, the average performance is slightly worse (-1.17%). For the hybrid method, we apply our best Poly-DPR model to index the entire corpus, and use the procedure as described in Sec 3.3. Our hybrid method which combines BM25 and Poly-DPR, is better than all existing methods.

We also report state-of-the-art (SOTA) results reported on the BioASQ8 leader-board. These approaches are a combination of retrieval and improved re-ranking methods. Since this paper is concerned with improving retrieval and does not study re-ranking, we do not compare our methods directly with these approaches, but report them for completeness.

Model	B1	B2	B3	B4	B5	Avg.
Small Corpus						
BM25 (2009)	62.15	61.30	66.62	74.14	61.30	65.10
DPR_{128} (2020)	54.48	50.51	53.80	59.06	48.71	53.31
DPR_{256}	44.86	41.18	40.25	47.78	40.42	42.89
P-DPR_{128} (Ours)	64.71	**64.92**	64.28	73.11	**66.29**	66.66
P-DPR_{256} (Ours)	64.57	58.51	64.02	68.44	62.60	63.62
Hybrid (DPR_{128})	**66.55**	61.29	68.08	72.91	60.30	65.83
Hybrid (P-DPR_{128})	66.30	64.90	**69.54**	**75.71**	64.82	**68.25**
Large Corpus						
BM25	28.50	27.82	37.97	41.91	35.42	34.32
GenQ (2021)	28.90	20.30	30.70	29.00	33.10	28.40
P-DPR_{128} (Ours)	**35.10**	29.07	32.74	33.31	35.54	33.15
Hybrid (P-DPR_{128})	30.02	**31.31**	**39.79**	**42.18**	**37.99**	**36.26**
Large Corpus SOTA (Re-ranking)						
PA(2020)	35.91	**39.45**	52.73	41.15	**52.02**	44.25
bioinfo-4 (2020)	**38.23**	36.86	51.08	46.77	50.98	**44.78**
AUEB-4 (2020)	5.47	7.23	**53.29**	**49.92**	49.53	33.09

Table 3: Comparison between our Poly-DPR (P-DPR) with baseline methods in the small corpus and large corpus settings. The bottom section shows performance of existing methods that make improvements in the re-ranking method.

Index Unit	Mem.	Time	B1	B2	B3	B4	B5	Avg.
2-sents	21.0 G	321	64.71	**64.92**	**64.28**	**73.11**	**66.29**	**66.66**
128-chunk	8.1 G	206	**65.16**	63.24	63.72	72.13	65.29	65.91
256-chunk	4.5 G	192	63.76	59.71	62.70	67.21	64.17	63.51
Full	2.8 G	101	61.92	57.84	60.01	61.11	62.66	60.71
2-sents	21.0 G	321	**64.65**	**59.21**	63.65	**70.90**	**65.97**	**64.88**
128-chunk	8.1 G	206	64.11	58.08	**64.15**	69.90	63.16	63.88
256-chunk	4.5 G	192	64.57	58.51	64.02	68.44	62.6	63.62
Full	2.8 G	101	60.06	56.38	61.99	65.01	59.63	60.61

Table 4: Two best NR models in short and long context: the first block is Poly-DPR pretrained with RSM and fine-tuned on TempQG (short); the second block is Poly-DPR pretrained with ETM and fine-tuned on TempQG (long).

6.3 Ablation Study

We provide ablation studies of different hyper-parameters on model performance. Results are reported on the small corpus.

Granularity of Indexing. Here we examine the impact of indexing units. We conjecture that the representation produced with a shorter indexing unit is better than the one with a longer indexing unit, and thus an NR should perform better if the indexing unit is short. To verify this, we use our best Poly-DPR models that are trained in short and long context settings. We compare four indexing units, **2-sents:** two consecutive sentences, **128 chunk:** a chunk with maximum length of 128 tokens that includes multiple consecutive sentences, **256 chunk:** a chunk with maximum length of 256 tokens that includes multiple consecutive sentences, and **512 chunk:** the entire article including title and abstract, and we use 512 tokens to encode each article. The results are shown in Table 4; we see that the smaller indexing units yield better performance, even for the model that is trained in long context setting. We also present the memory (*Mem.*) and inference time (*Time*) which depend upon the choice of indexing unit. The inference time refers to the number of seconds taken to retrieve 10 documents for 100 questions. Table 4 shows that a smaller indexing unit requires more memory and longer inference time. Thus, there is a trade-off between retrieval quality and memory as well as inference time. Future work could explore ways to improve the efficiency of neural retrievers to mitigate this trade-off.

NT	B1	B2	B3	B4	B5	Avg.
1	**67.21**	62.43	**66.49**	**72.15**	61.55	65.96
5	66.76	62.19	66.41	71.55	**64.33**	66.25
10	64.71	**64.92**	64.28	73.11	62.29	**66.66**

Table 5: Effect of number of templates (NT) on performance.

K	B1	B2	B3	B4	B5	Avg.
0	62.06	**61.81**	61.85	66.69	61.30	62.74
6	62.92	58.79	**62.94**	70.30	63.39	63.67
12	**65.22**	60.86	62.59	**70.50**	**66.21**	**65.08**
0	61.70	58.28	58.62	67.33	61.48	61.48
6	**63.95**	**59.51**	**62.98**	66.71	62.80	63.19
12	63.83	57.81	62.72	**70.00**	**63.64**	**63.60**

Table 6: Comparison among different values of K for Poly-DPR in both short and long context settings.

Number of Templates for Generating Questions We study three values for the number of templates, 1, 5, and 10, and report the results for Poly-DPR in Table 5. We see that training Poly-DPR on questions generated from one template is already better than BM25. While increasing the number of templates yields better performance, the improvement is relatively small, and we conjecture that this could be due to lower-quality or redundant templates. A question filtering module can be used to control the quality of the questions as shown in previous work (Alberti et al. 2019).

Number of Context Representations Poly-DPR encodes a context into K vector representations. We study the effect of three values of K (0, 6, and 12) on model performance, both with short (128) and long (256) contexts. All models are trained directly on the TempQG without pretraining. Table 6 shows that a larger K value yields better performance. This observation is aligned with Humeau et al. (2020).

6.4 Question Generation Analysis

Table 7 shows examples of selected templates and generated questions. Our template-based generation approach can produce diverse and domain-style questions using three strategies. **Fill in the blank**: the generator fills the blank in the template by key entities mentioned in the context without changing the template, as shown by Example 1. **Changing partially**: the generator produces questions by using part of the template and ignores some irrelevant part as shown by

#	Context	Template	Generated Question
1	The lysosomal-membrane protein type 2A (LAMP-2A) acts as the receptor for the substrates of chaperone-mediated autophagy (CMA), which should undergo unfolding before crossing the lysosomal membrane and reaching the lumen for degradation.	which receptor is targeted by _	Which receptor is targeted by LAMP-2A?
2	Is Tokuhashi score suitable for evaluation of life expectancy before surgery in Iranian patients with spinal metastases? One of the most important selection criteria for spinal metastases surgery is life expectancy and the most important system for this prediction has been proposed by Tokuhashi.	what is evaluated with _	What is the Tokuhashi score?
3	Lambert-Eaton myasthenic syndrome (LEMS) is a pre-synaptic disorder of the neuromuscular and autonomic transmission mediated by antibodies to voltage-gated calcium channels at the motor nerve terminal.	_ is diagnosed in which _	Lambert-Eaton myasthenic syndrome is diagnosed in which neuromuscular and autonomic pathways?

Table 7: Ilustrative examples for templates and questions generated by TempQG[1].

	Question	Explanation
B1	What is minodixil approved for?	minodixil is a typo, the correct one is minoxidil
B2	List 5 proteins with antioxidant properties?	BM25 fails to connect proteins and antioxidant properties, and retrieves documents all related to antioxidant, however, they are not about proteins nor antioxidant proteins.
B3	How large is a lncRNAs?	BM25 retrieves document about lncRNAs but not about how large it is.
P1	What is Xanamem?	NR fails to retrieve any document related to Xanamem, rather, it retrieves documents that lexical similar to Xanamem such as Ximenia, Xadago, and Xenopus.
P2	Does an interferon (IFN) signature exist for SLE patients?	NR ranks documents about interferon higher than documents of SLE patients and documents of both. In the retrieved documents, interferon appears rather frequently.

Table 8: Examples of the common failure modes of BM25 and Poly-DPR[1].

Example 2. **Ignoring entirely**: the generator ignores the template entirely and generates questions that are not relevant to the given context as shown by Example 3.

6.5 Error Analysis

To better understand the differences between BM25 and NR, we study their failure modes. From the BioASQ test set, we select questions on which either BM25 or Poly-DPR perform poorly, and categorize these failure cases (see Table 8).

Failures Cases of BM25. We found 91 failure cases on which the MAP score of BM25 is 0 for 41 cases, and the performance of BM25 is at least 0.5 less than Poly-DPR for 50 cases. Upon manual inspection, we identify three common categories of these failures. **B1:** questions contain keywords with typographical errors. **B2:** questions mention multiple entities related to each other. BM25 may fail to retrieve documents that connect these entities. **B3:** questions mention conceptual properties of entities and answers are values. For example, "how large" is a conceptual property and "200" is the answer value. BM25 retrieves documents related to the entities in questions but not contain the answer.

Failure cases of Poly-DPR. There we 55 failure cases of Poly-DPR, including 23 cases with 0 MAP score and 32 case where the score for BM25 is at least 0.5 better than Poly-DPR. There are two common failure modes of Poly-DPR. **P1:** questions are simple but focused on rare entities which Poly-DPR fails to retrieve. This conforms with the finding that NR performs significantly worse than BM25 on entity-questions (Sciavolino et al. 2021). We find that for such questions, retrieved entities and entities in the question are lexical similar or have overlapping substrings, which in turn could be due to the WordPiece embeddings (Wu et al. 2016) used in BERT. **P2:** Questions mention multiple entities. Articles that contain frequent entities are ranked higher than articles that include *all* entities in the question.

7 Discussion and Conclusion

In this work, we show that DPR, a neural retriever, is unable to surpass BM25 on biomedical benchmarks such as BioASQ. We address this drawback of NRs with a three-pronged approach with Poly-DPR: a new model architecture, TempQG: a template-based question generation method, and two new pre-training tasks designed for biomedical documents. TempQG can generate high quality domain-relevant questions which positively impact downstream performance. While in this paper, we apply TempQG to a small corpus of 100,000 PubMed articles, we show that this method can surpass neural retrievers when trained on small or large corpora. Our model achieves better performance than BM25 in the small corpus setting, but it falls short by $\sim 1\%$ in the large corpus setting. However, we show that a hybrid model combining our approach and BM25 is better than all previous baselines on the entire corpus. In the future, applying our question generation methods to the entire PubMed corpus, and combining our approach with improved re-ranking techniques could potentially result in further improvement.

Acknowledgements

This work was funded in part by National Science Foundation grants 2132724, 1816039 and 1750082, DARPA SAIL-ON program (W911NF2020006), and DARPA CHESS program (FA875019C0003). The views and opinions of the authors expressed herein do not necessarily state or reflect those of the funding agencies and employers. Code is available at https://github.com/luomancs/neural_retrieval_for_biomedical_domain.git

References

Alberti, C.; Andor, D.; Pitler, E.; Devlin, J.; and Collins, M. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6168–6173. Florence, Italy: Association for Computational Linguistics.

Almeida, T.; and Matos, S. 2020. BIT. UA at BioASQ 8: Lightweight Neural Document Ranking with Zero-shot Snippet Retrieval. In *CLEF (Working Notes)*.

Banerjee, P.; Gokhale, T.; and Baral, C. 2021. Self-Supervised Test-Time Learning for Reading Comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1200–1211.

Banerjee, P.; Gokhale, T.; Yang, Y.; and Baral, C. 2021. WeaQA: Weak supervision via captions for visual question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3420–3435.

Chan, Y.-H.; and Fan, Y.-C. 2019. A Recurrent BERT-based Model for Question Generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 154–162. Association for Computational Linguistics.

Chang, W.; Yu, F. X.; Chang, Y.; Yang, Y.; and Kumar, S. 2020. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 13042–13054.

Du, X.; and Cardie, C. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Fabbri, A.; Ng, P.; Wang, Z.; Nallapati, R.; and Xiang, B. 2020. Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4508–4513. Online: Association for Computational Linguistics.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M.-W. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *ArXiv*, abs/2002.08909.

Hosking, T.; and Lapata, M. 2021. Factorising Meaning and Form for Intent-Preserving Paraphrasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics.

Humeau, S.; Shuster, K.; Lachaux, M.; and Weston, J. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781. Online: Association for Computational Linguistics.

Kazaryan, A.; Sazanovich, U.; and Belyaev, V. 2020. Transformer-Based Open Domain Biomedical Question Answering at BioASQ8 Challenge. In *CLEF (Working Notes)*.

Khattab, O.; and Zaharia, M. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In Huang, J.; Chang, Y.; Cheng, X.; Kamps, J.; Murdock, V.; Wen, J.; and Liu, Y., eds., *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, 39–48. ACM.

Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2019. Information Maximizing Visual Question Generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2008–2018. Computer Vision Foundation / IEEE.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36: 1234 – 1240.

Lee, K.; Chang, M.-W.; and Toutanova, K. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6086–6096. Florence, Italy: Association for Computational Linguistics.

Lewis, P.; Denoyer, L.; and Riedel, S. 2019. Unsupervised Question Answering by Cloze Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4896–4910. Florence, Italy: Association for Computational Linguistics.

Lewis, P.; Stenetorp, P.; and Riedel, S. 2021. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. In *EACL*.

Li, Y.; Duan, N.; Zhou, B.; Chu, X.; Ouyang, W.; Wang, X.; and Zhou, M. 2018. Visual Question Generation as Dual Task of Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society.

Lin, J.; Ma, X.; Lin, S.-C.; Yang, J.-H.; Pradeep, R.; and Nogueira, R. 2021. Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations. *ArXiv*, abs/2102.10073.

Lopez, L. E.; Cruz, D. K.; Cruz, J. C. B.; and Cheng, C. 2020. Transformer-based End-to-End Question Generation. *ArXiv*, abs/2005.01107.

Ma, J.; Korotkov, I.; Yang, Y.; Hall, K.; and McDonald, R. T. 2021. Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation. In *EACL*.

MacAvaney, S.; Yates, A.; Cohan, A.; and Goharian, N. 2019. CEDR: Contextualized Embeddings for Document Ranking. In Piwowarski, B.; Chevalier, M.; Gaussier, É.; Maarek, Y.; Nie, J.; and Scholer, F., eds., *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 1101–1104. ACM.

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

Pappas, D.; Stavropoulos, P.; and Androutsopoulos, I. 2020. AUEB-NLP at BioASQ 8: Biomedical Document and Snippet Retrieval. In *CLEF (Working Notes)*.

Raffel, C.; Shazeer, N. M.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *ArXiv*, abs/1910.10683.

Robertson, S.; and Zaragoza, H. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3: 333–389.

Sciavolino, C.; Zhong, Z.; Lee, J.; and Chen, D. 2021. Simple Entity-centric Questions Challenge Dense Retrievers. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Shortliffe, E. H.; Shortliffe, E. H.; Cimino, J. J.; and Cimino, J. J. 2014. *Biomedical informatics: computer applications in health care and biomedicine*. Springer.

Shrivastava, A.; and Li, P. 2014. Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS). In *Advances in Neural Information Processing Systems*, 2321–2329.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 3104–3112.

Thakur, N.; Reimers, N.; Ruckl'e, A.; Srivastava, A.; and Gurevych, I. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *ArXiv*, abs/2104.08663.

Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; Almirantis, Y.; Pavlopoulos, J.; Baskiotis, N.; Gallinari, P.; Artières, T.; Ngomo, A. N.; Heino, N.; Gaussier, É.; Barrio-Alvers, L.; Schroeder, M.; Androutsopoulos, I.; and Paliouras, G. 2015. An overview of the large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yang, W.; Xie, Y.; Lin, A.; Li, X.; Tan, L.; Xiong, K.; Li, M.; and Lin, J. 2019. End-to-End Open-Domain Question Answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Zhao, Y.; Ni, X.; Ding, Y.; and Ke, Q. 2018. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3901–3910. Brussels, Belgium: Association for Computational Linguistics.