Zero-shot Neural Passage Retrieval via Domain-targeted Synthetic Question Generation

Ji Ma **Ivan Korotkov** **Yinfei Yang** **Keith Hall** **Ryan McDonald**
Google Research
{maji,ivankr,yinfeiy,kbhall,ryanmcd}@google.com

Abstract

A major obstacle to the wide-spread adoption of neural retrieval models is that they require large supervised training sets to surpass traditional term-based techniques, which are constructed from raw corpora. In this paper, we propose an approach to zero-shot learning for passage retrieval that uses synthetic question generation to close this gap. The question generation system is trained on general domain data, but is applied to documents in the targeted domain. This allows us to create arbitrarily large, yet noisy, question-passage relevance pairs that are domain specific. Furthermore, when this is coupled with a simple hybrid term-neural model, first-stage retrieval performance can be improved further. Empirically, we show that this is an effective strategy for building neural passage retrieval models in the absence of large training corpora. Depending on the domain, this technique can even approach the accuracy of supervised models.

1 Introduction

Recent advances in neural retrieval have led to advancements on several document, passage and knowledge-base benchmarks (Guo et al., 2016; Pang et al., 2016; Hui et al., 2017; Dai et al., 2018; Gillick et al., 2018; Nogueira and Cho, 2019a; MacAvaney et al., 2019; Yang et al., 2019a,b,c). Most neural passage retrieval systems are, in fact, two stages (Zamani et al., 2018; Yilmaz et al., 2019), illustrated in Figure 1. The first is a true retrieval model (aka first-stage retrieval[1]) that takes a question and retrieves a set of candidate passages from a large collection of documents. This stage itself is rarely a neural model and most commonly is an term-based retrieval model such as BM25 (Robertson et al., 2004; Yang et al., 2017), though there is recent work on neural models (Zamani et al., 2018; Dai and Callan, 2019; Chang et al., 2020; Karpukhin et al., 2020; Luan et al., 2020). This is usually due to the computational costs required to dynamically score large-scale collections. Another consideration is that BM25 is often high quality (Lin, 2019). After first-stage retrieval, the second stage uses a neural model to rescore the filtered set of passages. Since the size of the filtered set is small, this is feasible.

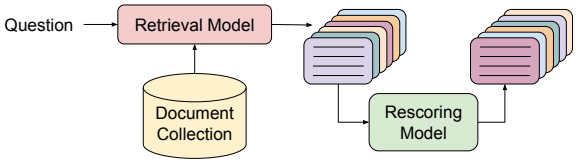

Figure 1: End-to-end neural retrieval. A first-stage model over a large collection returns a smaller set of relevant passages which are reranked by a rescorer.

The focus of the present work is methods for building neural models for *first-stage passage retrieval* for large collections of documents. While rescoring models are key components to any retrieval system, they are out of the scope of this study. Specifically, we study the *zero-shot* setting where there is no target-domain supervised training data (Xian et al., 2018). This is a common situation, examples of which include enterprise or personal search environments (Hawking, 2004; Chirita et al., 2005), but generally any specialized domain.

The zero-shot setting is challenging as the most effective neural models have a large number of parameters, which makes them prone to overfitting. Thus, a key factor in training high quality neural models is the availability of large training sets. To address this, we propose two techniques to improve neural retrieval models in the zero-shot setting.

First, we observe that general-domain question-passage pairs can be acquired from community platforms (Shah and Pomerantz, 2010; Duan et al., 2017) or high quality academic datasets that are publicly available (Kwiatkowski et al., 2019; Bajaj et al., 2016). Such resources have been used to

[1] Also called *open domain* retrieval.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1075–1088
April 19 - 23, 2021.

create open domain QA passage retrieval models. However, as shown in Guo et al. (2020) and in our later experiments, neural retrieval models trained on the general domain data often do not transfer well, especially for specialized domains.

Towards zero-shot neural retrieval with improved domain adaptability, we propose a data augmentation approach (Wong et al., 2016) that leverages these naturally occurring question/answer pairs to train a generative model that synthesizes questions given a text (Zhou et al., 2017). We apply this model to passages in the target domain to generate unlimited pairs of synthetic questions and target-domain passages. This data can then be used for training. This technique is outlined in Figure 2.

A second contribution is a simple hybrid model that interpolates a traditional term-based model – BM25 (Robertson et al., 1995) – with our zero-shot neural model. BM25 is also zero-shot, as its parameters do not require supervised training. Instead of using inverted index which is commonly used in term-based search, we exploit the fact that BM25 and neural models can be cast as vector similarity (see Section 4.4) and thus nearest neighbour search can be used for retrieval (Liu et al., 2011; Johnson et al., 2017). The hybrid model takes the advantage of both the term matching and semantic matching.

We compare a number of baselines including other data augmentation and domain transfer techniques. We show on three specialized domains (scientific literature, travel and tech forums) and one general domain that the question generation approach is effective, especially when considering the hybrid model. Finally, for passage retrieval in the scientific domain, we compare with a number of recent supervised models from the BioASQ challenge, including many with rescoring stages. Interestingly, the quality of the zero-shot hybrid model approaches supervised alternatives.

2 Related Work

Neural Retrieval The retrieval vs. rescorer distinction (Figure 1) often dictates modelling choices for each task. For first-stage retrieval, as mentioned earlier, term-based models that compile document collections into inverted indexes are most common since they allow for efficient lookup (Robertson et al., 2004; Yang et al., 2017). However, there are studies that investigate neural first-stage retrieval. A common technique is to learn the term weights to be used in an inverted index (Zamani et al., 2018; Dai and Callan, 2019, 2020). Another technique is representation-based models that embed questions and passages into a common dense subspace (Palangi et al., 2016) and use nearest neighbour search for retrieval (Liu et al., 2011; Johnson et al., 2017). Recent work has shown this can be effective for passage scoring (Chang et al., 2020; Karpukhin et al., 2020; MacAvaney et al., 2020). Though all of the aforementioned first-stage neural models assume supervised data for fine-tuning. For rescoring, scoring a small set of passages permits computationally intense models. These are often called interaction-based, one-tower or cross-attention models and numerous techniques have been developed (Guo et al., 2016; Hui et al., 2017; Xiong et al., 2017; Dai et al., 2018; McDonald et al., 2018), many of which employ pre-trained contextualized models (Nogueira and Cho, 2019a; MacAvaney et al., 2019; Yang et al., 2019a,b). Khattab and Zaharia (2020) also showed that by delaying interaction to the last layer, one can build a first stage retrieval model which also leverages the modeling capacity of an interaction based models.

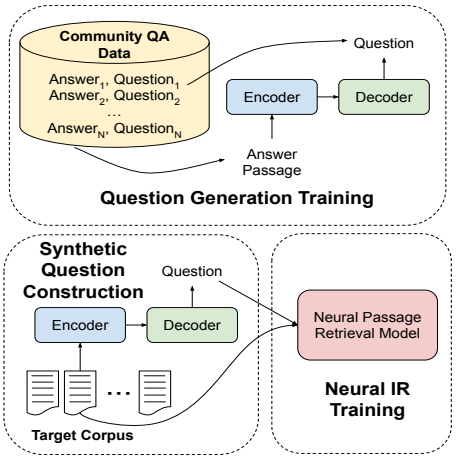

Figure 2: Synthetic query generation for neural IR.

Model Transfer Previous work has attempted to alleviate reliance on large supervised training sets by pre-training deep retrieval models on weakly supervised data such as click-logs (Borisov et al., 2016; Dehghani et al., 2017). Recently, Yilmaz et al. (2019) has shown that training models on general-domain corpora adapts well to new domains without targeted supervision. Another common technique for adaptation to specialized domains is to learn cross-domain representations (Cohen et al., 2018; Tran et al., 2019). Our work is more aligned with methods like Yilmaz et al. (2019) which use general domain resources to build neural models for new domains, though via different techniques – data augmentation vs. model transfer. Our experiments show that data augmentation

compares favorably a model transfer baseline. For specialized domains, recently, there have been a number of studies using cross-domain transfer and other techniques for biomedical passage retrieval via the TREC-COVID challenge[2,3] that uses the CORD-19 collection (Wang et al., 2020).

Question generation for data augmentation is a common tool, but has not been tested in the pure zero-shot setting nor for neural passage retrieval. Duan et al. (2017) use community QA as a data source, as we do, to train question generators. The generated question-passage pairs are not used to train a neural model, but QA is instead done via question-question similarity. Furthermore, they do not test on specialized domains. Alberti et al. (2019) show that augmenting supervised training resources with synthetic question-answer pairs can lead to improvements. Nogueira et al. (2019) employed query generation in the context of first-stage retrieval. In that study, the generated queries were used to augment documents to improve BM25 keyword search. Here we focus on using synthetic queries to train the neural retrieval models.

Hybrid Models Combining neural and term-based models have been studied, most commonly via linearly interpolating scores in an approximate re-ranking stage (Karpukhin et al., 2020; Luan et al., 2020) or through the final layer of a rescoring network (Severyn et al., 2015; McDonald et al., 2018). Since rescoring can be cast as classification, blending signals is straight-forward. However, this is approximate as it does not operate over the whole collection. For first-stage retrieval, the most common method is to learn term weights for a standard inverted index in order to make search efficient (Zamani et al., 2018; Dai and Callan, 2019). Here we propose a first-stage retrieval model that incorporates both term-based (sparse) and neural-based (dense) representations in a hybrid model that uses nearest neighbor search for exact inference (Liu et al., 2011; Johnson et al., 2017; Wu et al., 2019). Similar methods using approximate nearest neighbour search have been investigated by Seo et al. (2019).

3 Synthetic Question Generation

In this work, we are specifically investigating the zero-shot scenario where there exists neither user issued questions nor domain specific data except the passage collection itself. We propose to address the training data scarcity issue by generating synthetic questions (Zhou et al., 2017; Duan et al., 2017; Alberti et al., 2019; Nogueira et al., 2019). Leverage the fact that there are large question-answer data sources freely available from the web (Shah and Pomerantz, 2010; Duan et al., 2017). we first train a question generator using general domain question-answer pairs. The passage collection of a target domain is then fed into this generator to create pairs of noisy question-passage pairs, which are used to train a retrieval model (see Figure 2). In this work, we mine English question-answer pairs from community resources, primarily StackExchange[4] and Yahoo! Answers[5]. Note we use stackexchange as it covers a wide range of topics, and we focus on investigating the domain adaptability of using a question generation approach. We leave comparing question generator trained on different datasets or using different architectures to future work.

To ensure data quality, we further filter the data by only keeping question-answer pairs that were positively rated by at least one user on these sites. In total, the final dataset contains 2 millions pairs, and the average length of questions and answers are 12 tokens and 155 tokens respectively. This dataset is *general domain* in that it contains question-answer pairs from a wide variety of topics.

Our question generator is an encoder-decoder with Transformer (Vaswani et al., 2017) layers, which is a common for generation tasks such as translation and summarization (Vaswani et al., 2017; Rothe et al., 2019). The encoder is trained to build a representation for a text and the decoder generates a question for which that text is a plausible answer. Appendix B has model specifics.

Ubuntu Forums

Passage: Every time I get a notification about and begin updating when they become available, the process is interrupted by an error message: error in foomatic-filters. Then I get "error in linux generic package" and a bunch of numbers. This is replaced before I can write it all down with "error in Linux package" Everything seems to go OK except I don't know if the updates are really being installed. I tried un-installing and re-installing foomatic-filters . . .
Generated Question: How do I get rid of error in foomatic-filters?

Biomedical Literature

Passage: Electroencephalographic tracings of 50 patients who presented the classical features of Friedreich's ataxia were reviewed . . . Friedreich's ataxia is mainly a spinal disorder. Involvement of supraspinal and in particular brain stem or diencephalic structures may be more extensive in those patients who show electrographic abnormalities. This would require confirmation with comparative data based on pathological observations. Impaired function of brain stem inhibitory mechanism may be responsible for the slightly raised incidence of seizures in patients with Friedreich's ataxia and other cerebellar degenerations.
Generated Question: What is the significance of Friedreich's ataxia?

Table 1: Examples of domain-targeted synthetic generated questions used to train passage retrieval models.

[2] ir.nist.gov/covidSubmit/
[3] ir.nist.gov/covidSubmit/archive.html
[4] archive.org/details/stackexchange
[5] webscope.sandbox.yahoo.com/catalog.php?datatype=l

Our approach is robust to domain shift as the generator is trained to create questions based on a given text. As a result, generated questions stay close to the source passage material. Real examples are shown in Table 1 for technical and biomedical domains, highlighting the model's adaptability.

4 Neural First-stage Retrieval

In this section we describe our architecture for training a first-stage neural passage retriever. Our retrieval model belongs to the family of *relevance-based dense retrieval* [6] that encodes pairs of items in dense subspaces (Palangi et al., 2016). Let $Q = (q_1, \ldots q_n)$ and $P = (p_1, \ldots, p_m)$ be a question and passage of n and m tokens respectively. Our model consists of two encoders, $\{f_Q(), f_P()\}$ and a similarity function, sim(). An encoder is a function f that takes an item x as input and outputs a real valued vector as the encoding, The similarity function, sim(), takes two encodings, $\mathbf{q}, \mathbf{p} \in \mathbb{R}^N$ and calculates a real valued score, $s = \text{sim}(\mathbf{q}, \mathbf{p})$. For passage retrieval, the two encoders are responsible for computing dense vector representation of questions and passages.

4.1 BERT-based Encoder

In this work, both query and document encoders are based on BERT (Devlin et al., 2019), which has been shown to lead to large performance gains across a number of tasks, including document ranking (Nogueira and Cho, 2019a; MacAvaney et al., 2019; Yang et al., 2019b). In addition, we share parameters between the query and passage encoder – i.e., $f_Q = f_P$, so called Siamese networks – as we found this greatly increased performance while reducing parameters.

We encode P as (CLS, $p_1, \ldots, p_m$, SEP). For some datasets, a passage contains both a title $T = (t_1, ..., t_l)$ and content $C = (c_1, ..., c_o)$, in which case we encode the passage as (CLS, $t_1, ..., t_l$, SEP, $c_1, ..., c_o$, SEP). These sequences are fed to the BERT encoder. Let $h_{\text{CLS}} \in \mathbb{R}^N$ be the final representation of the "CLS" token. Passage encodings $\mathbf{p}$ are computed by applying a linear projection, i.e., $\mathbf{p} = \mathbf{W} * h_{\text{CLS}}$, where $\mathbf{W}$ is a $N \times N$ weight matrix (thus $N = 768$), which preserves the original size of h_{CLS}. This has been shown to perform better than down-projecting to a lower dimensional vector (Luan et al., 2020), especially for long passages.

We encode Q as (CLS, $q_1, q_2, ..., q_n$, SEP) which is then fed to the BERT encoder. Similarly, a linear projection on the corresponding "CLS" token, using the same weight matrix $\mathbf{W}$, is applied to generate $\mathbf{q}$. Following previous work (Luan et al., 2020; Lee et al., 2019b), we use dot product as the similarity function, i.e., $\text{sim}(\mathbf{q}, \mathbf{p}) = \langle \mathbf{q}, \mathbf{p} \rangle = \mathbf{q}^\intercal \mathbf{p}$.

The top half of Figure 3 illustrates the model.

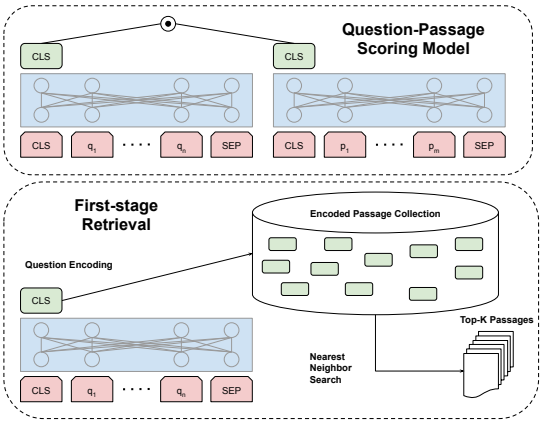

Figure 3: First-stage neural passage retrieval. Top: A BERT-based transformer encodes questions and passages and scores them via dot-product. Bottom: Passages from the collection are encoded and stored in a nearest neighbour search backend. At inference, the question is encoded and relevant passages retrieved.

4.2 Training

For training, we adopt softmax cross-entropy loss. Formally, given an instance $\{\mathbf{q}, \mathbf{p}^+, \mathbf{p}_1^-, ..., \mathbf{p}_k^-\}$ which comprises one query $\mathbf{q}$, one relevant passage $\mathbf{p}^+$ and k non-relevant passages $\mathbf{p}_i^-$. The objective is to minimize the negative log-likelihood:

$$L(\mathbf{q}, \mathbf{p}^+, \mathbf{p}_1^-, ..., \mathbf{p}_k^-) = \log(e^{\langle \mathbf{q}, \mathbf{q}^+ \rangle} + \sum_{i=1}^{k} e^{\langle \mathbf{q}, \mathbf{q}_i^- \rangle}) - \langle \mathbf{q}, \mathbf{q}^+ \rangle$$

This loss function is a special case of ListNet loss (Cao et al., 2007) where all relevance judgements are binary, and only one passage is marked relevant for each training example.

For the set $\{\mathbf{p}_1^-, ..., \mathbf{p}_k^-\}$, we use in-batch negatives. Given a batch of (query, relevant-passage) pairs, negative passages for a query are passages from different pairs in the batch. In-batch negatives has been widely adopted as it enables efficient training via computation sharing (Yih et al., 2011; Gillick et al., 2018; Karpukhin et al., 2020).

4.3 Inference

Since the relevance-based model encodes questions and passages independently, we run the encoder

[6] A.k.a. two-tower, dual encoder or dense retrieval.

over every passage in a collection offline to create a distributed lookup-table as a backend. At inference, we run the question encoder online and then perform nearest neighbor search to find relevant passages, as illustrated in the bottom half of Figure 3. While there has been extensive work in fast approximate nearest neighbour retrieval for dense representations (Liu et al., 2011; Johnson et al., 2017), we simply use distributed brute-force search as our passage collections are at most in the millions, resulting in exact retrieval.

4.4 Hybrid First-stage Retrieval

Traditional term-based methods like BM25 (Robertson et al., 1995) are powerful zero-shot models and can outperform supervised neural models in many cases (Lin, 2019). Rescoring systems have shown that integrating BM25 into a neural model improves performance (McDonald et al., 2018). However, for first-stage retrieval most work focuses on approximations via re-ranking (Karpukhin et al., 2020; Luan et al., 2020). Here we present a technique for exact hybrid first-stage retrieval without the need for a re-ranking stage. Our method is motivated by the work of Seo et al. (2019) for sparse-dense QA.

For a query Q and a passage P, BM25 is computed as the following similarity score,

$$\text{BM25}(Q,P) = \sum_{i=1}^{n} \frac{\text{IDF}(q_i) * \text{cnt}(q_i \in P) * (k+1)}{\text{cnt}(q_i \in P) + k * (1 - b + b * \frac{m}{m_{\text{avg}}})},$$

where k/b are BM25 hyperparameters, IDF is the term's inverse document frequency from the corpus, cnt is the term's frequency in a passage, n/m are the number of tokens in Q/P, and m_{avg} is the collection's average passage length.

Like most TF-IDF models, this can be written as a vector space model. Specifically, let $\mathbf{q}^{\text{bm25}} \in [0,1]^{|V|}$ be a sparse binary encoding of a query of dimension $|V|$, where V is the term vocabulary. Specifically this vector is 1 at position i if $v_i \in Q$, here v_i is the i-th entry in V. Furthermore, let $\mathbf{p}^{\text{bm25}} \in \mathbb{R}^{|V|}$ be a sparse real-valued vector where,

$$\mathbf{p}_i^{\text{bm25}} = \frac{\text{IDF}(v_i) * \text{cnt}(v_i \in P) * (k+1)}{\text{cnt}(v_i \in P) + k * (1 - b + b * \frac{m}{m_{\text{avg}}})}$$

We can see that,

$$\text{BM25}(Q,P) = \langle \mathbf{q}^{\text{bm25}}, \mathbf{p}^{\text{bm25}} \rangle$$

As BM25 score can be written as vector dot-product, this gives rise to a simple hybrid model,

$$\begin{aligned} \text{sim}(\mathbf{q}^{\text{hyb}}, \mathbf{p}^{\text{hyb}}) &= \langle \mathbf{q}^{\text{hyb}}, \mathbf{p}^{\text{hyb}} \rangle \\ &= \langle [\lambda \mathbf{q}^{\text{bm25}}, \mathbf{q}^{\text{nn}}], [\mathbf{p}^{\text{bm25}}, \mathbf{p}^{\text{nn}}] \rangle \\ &= \lambda \langle \mathbf{q}^{\text{bm25}}, \mathbf{p}^{\text{bm25}} \rangle + \langle \mathbf{q}^{\text{nn}}, \mathbf{p}^{\text{nn}} \rangle, \end{aligned}$$

where $\mathbf{q}^{\text{hyb}}$ and $\mathbf{p}^{\text{hyb}}$ are the hybrid encodings that concatenate the BM25 ($\mathbf{q}^{\text{bm25}}$/$\mathbf{p}^{\text{bm25}}$) and the neural encodings ($\mathbf{q}^{\text{nn}}$/$\mathbf{p}^{\text{nn}}$, from Sec 4.1); and λ is a interpolation hyperparameter that trades-off the relative weight of BM25 versus neural models.

Thus, we can implement BM25 and our hybrid model as nearest neighbor search with hybrid sparse-dense vector dot-product (Wu et al., 2019).

5 Experimental Setup

We outline data and experimental details. The Appendix has further information to aid replicability.

5.1 Evaluation Datasets

BioASQ Biomedical questions from Task B Phase A of BioASQ (Tsatsaronis et al., 2015). We use BioASQ 7 and 8 test data for evaluation. The collection contains all abstracts from MEDLINE articles. Given an article, we split its abstract into chunks with sentence boundaries preserved. A passage is constructed by concatenating the title and one chunk. Chunk size is set so that each passage has no more than 200 wordpiece tokens.

Forum Threads from two online user forum domains: Ubuntu technical help and TripAdvisor topics for New York City (Bhatia and Mitra, 2010). For each thread, we concatenate the title and initial post to generate passages. For BERT-based models we truncate at 350 wordpiece tokens. Unlike the BioASQ data, this data generally does not contain specialist knowledge queries. Thus, compared to the collection of question-answer pairs mined from the web, there is less of a domain shift.

NaturalQuestions Aggregated queries issued to Google Search (Kwiatkowski et al., 2019) with relevance judgements. We convert the original format to a passage retrieval task, where the goal is to retrieval the long answer among all wiki paragraphs (Ahmad et al., 2019). We discarded questions whose long answer is either a table or a list. We evaluate retrieval performance on the development set as the test set is not publicly available. The target collection contains all passages from the development set and is augmented with passages from 2016-12-21 dump of Wikipedia (Chen et al.,

2017). Each passage is also concatenated with title. For BERT-based models passages are truncated at 350 wordpiece tokens. This data is different from the previous data in two regards. First, there is a single annotated relevant paragraph per query. This is due to the nature in which the data was curated. Second, this data is entirely "general domain".

Dataset statistics are listed in Appendix A.

5.2 Zero-shot Systems

BM25 Term-matching systems such as BM25 (Robertson et al., 1995) are themselves zero-shot, since they require no training resources except the document collection itself. We train a standard BM25 retrieval model on the document collection for each target domain.

ICT The Inverse Cloze Task (ICT) (Lee et al., 2019b) is an unsupervised pre-training objective which randomly masks out a sentence from a passage and creates synthetic sentence-passage pairs representing membership of the sentence in the passage. These masked examples can then used to train or pre-train a retrieval model. Lee et al. (2019b) showed that masking a sentence with a certain probability, p, can both mimic the performance of lexical matching ($p = 0$) or semantic matching ($p > 0$). ICT is *domain-targeted* since training examples are created directly from the relevant collection. Chang et al. (2020) showed that ICT-based pre-training outperforms a number of alternatives such as Body First Selection (BFS) or Wiki Link Prediction (WLP) for large-scale retrieval.

Ngram Gysel et al. (2018) proposes to train unsupervised neural retrieval system by extracting ngrams and titles from each document as queries. Different from ICT, this approach does not mask the extract ngrams from the original document.

QA The dataset mined from community question-answer forums (Sec. 3) itself can be used directly to train a neural retrieval model since it comes of the form query and relevant text (passage) pair. This data is naturally occurring and not systematically noisy, which is an advantage. However, the data is not domain-targeted, in that it comes from general knowledge questions. We call models trained on this dataset as QA. Applying a model trained on general domain data to a specific domain with no adaptation is a strong baseline (Yilmaz et al., 2019).

QGen The QGen retrieval model trained on the domain-targeted synthetic question-passage pairs described in Section 3. While this model can contain noise from the generator, it is domain-targeted.

	QA	ICT	Ngram	ICT+Ngram	QGen
BioASQ	2.00M	90.50M	636.54M	727.05M	82.62M
NQ	2.00M	71.58M	356.15M	427.72M	84.33M
Forum Travel	2.00M	0.30M	1.25M	1.54M	0.26M
Forum Ubuntu	2.00M	0.42M	2.07M	2.49M	0.43M

Table 2: Number of (synthetic-question, passage) pairs used in zero-shot experiments.

QGenHyb This is identical to QGen, but instead of using the pure neural model, we train the hybrid model in Section 4.4 setting $\lambda = 1.0$ for all models to avoid any domain-targeted tuning. We train the term and neural components independently, combing them only at inference.

All **ICT**, **NGram**, **QA** and **QGen** models are trained using the neural architecture from Section 4. For BioASQ experiments, question and passage encoders are initialized with BioBERT base v-1.1 (Lee et al., 2019a). All other data uses uncased BERT base (Devlin et al., 2019).

We can categorize the neural zero-shot models along two dimensions *extractive* vs. *transfer*. ICT and Ngram are extractive, in that they extract exact substrings from a passage to create synthetic questions for model training. Note that extractive models are also unsupervised, since they do not rely on general domain resources. QA is a *direct* cross-domain transfer model, in that we train the model on data from one domain (or general domain) and directly apply it to the target domain for retrieval. QGen models are *in-direct* cross-domain transfer models, in that we use the out-of-domain data to generate resources for model training.

5.3 Generated Training Datasets

The nature of each zero-shot neural system requires different generated training sets. For ICT, we follow Lee et al. (2019b) and randomly select at most 5 sentences from a document, with a mask rate of 0.9. For Ngram models, Gysel et al. (2018) suggests that retrieval models trained with ngram-order of around 16 was consistently high in quality. Thus, in our experiment we also use 16 and move the ngram window with a stride of 8 to allow 8 token overlap between consecutive ngrams.

For QGen models, each passage is truncated to 512 sentence tokens and feed to the question generation system. We also run the question generator on individual sentences from each passage to promote questions that focus on different aspects of the same document. We select at most 5 salient sentences from a passage, where sentence saliency

is the max term IDF value in a sentence.

The size of the generated training set for each baseline is shown in Table 2.

6 Results and Discussion

Our main results are shown in Table 3. We compute Mean Average Precision over the first N[7] results (MAP), Precision@10 and nDCG@10 (Manning et al., 2008) with TREC evaluation script[8]. All numbers are in percentage.

Accuracy of pure neural models are shown in the upper group of Table 3. First, we see that both QA and QGen consistently outperform neural baselines such as ICT and Ngram that are based on sub-string masking or matching. Matching on sub-strings likely biases the model towards memorization instead of learning salient concepts of the passage. Furthermore, query encoders trained on sub-strings are not exposed to many questions, which leads to adaptation issues when applied to true retrieval tasks. Comparing QGen with QA, typically QGen performs better, especially for specialized target domains. This suggests that domain-targeted query generation is more effective for domain shift than direct cross-domain transfer (Yilmaz et al., 2019).

Performance of term-based models and hybrid models are shown in Table 3 (bottom). We can see that BM25 is a very strong baseline. However, this could be an artifact of the datasets as the queries are created by annotators who already have the relevant passage in mind. Queries created this way typically have large lexical overlapping with the passage, thus favoring term matching based approaches like BM25. This phenomenon has been observed by previous work Lee et al. (2019b). Nonetheless, the hybrid model outperforms BM25 on all domains, and the improvements are statistically significant on 9/12 metrics. This illustrate that term-based model and neural-based model return complementary results, and the proposed hybrid approach effectively combines their strengths.

For NaturalQuestions since there is a single relevant passage annotation, we report Precision@1 and Mean reciprocal rank (MRR)[9]. Results are show in Table 4. We can see here that while QGen still significantly outperform other baselines, the gap between QGen and QA is smaller. Unlike BioASQ and Forum datasets, NaturalQuestions contains general domain queries, which aligns well with the question-answer pairs for training the QA model. Another difference is that NaturalQuestions consists of real information seeking queries, in this case QGen performs better than BM25.

6.1 Zero-shot vs. Supervised

One question we can ask is how close to the state-of-the-art in supervised passage retrieval are these zero-shot models. To test this we looked at BioASQ 8 dataset and compare to the top-participant systems.[10] Since BioASQ provides annotated training data, the top teams typically use supervised models with a first-stage retrieval plus rescorer architecture. For instance, the AUEB group, which is the top or near top system for BioASQ 6, 7 and 8, uses a BM25 first-stage retrieval model plus a supervised neural rescorer (Brokos et al., 2018; Pappas et al., 2019).

In order to make our results comparable to participant systems, we return only 10 passages per question (as per shared-task guidelines) and use the official BioASQ 8 evaluation software.

Table 5 shows the results for three zero-shot systems (BM25, QGen and QGenHyb) relative to the top 4 systems on average across all 5 batches of the shared task. We can see the QGenHyb performs quite favorably and on average is indistinguishable from the top systems. This is very promising and suggests that top-performance for zero-shot retrieval models is possible.

A natural question is whether improved first-stage model plus supervised rescoring is additive. The last two lines of the table takes the two-best first-stage retrieval models and adds a simple BERT-based cross-attention rescorer (Nogueira and Cho, 2019b; MacAvaney et al., 2019). We can see that, on average, this does improve quality. Furthermore, having a better first-stage retriever (QGenHyb vs. BM25) makes a difference.

As noted earlier, on BioASQ, BM25 is a very strong baseline. This makes the BM25/QGenHyb zero-shot models highly likely to be competitive. When we look at NaturalQuestions, where BM25 is significantly worse than neural models, we see that the gap between zero-shot and supervised widens substantially. The last row of Table 4 shows a model trained on the NaturalQuestions training data, which is nearly 2-3 times more accurate than the best zero-shot models. Thus, while zero-shot neural models have the potential to be competitive with supervised counterparts, the experiments here show this is data dependant.

[7] BioASQ: N=100; and Forum: N=1000.

[8] https://trec.nist.gov/trec_eval/

[9] MRR = MAP when there is one relevant item.

[10] participants-area.bioasq.org

	BioASQ 7			BioASQ 8			Forum Travel			Forum Ubuntu		
	MAP	**Prec @10**	**nDCG @10**	**MAP**	**Prec @10**	**nDCG @10**	**MAP**	**Prec @10**	**nDCG @10**	**MAP**	**Prec @10**	**nDCG @10**
NEURAL MODELS												
ICT*	9.31*	3.84*	11.44*	9.31*	3.36*	11.78*	3.66*	11.60*	12.04*	8.93*	21.60*	23.21*
Ngram*	9.17*	3.86*	11.53*	8.81*	2.84*	10.74*	10.00	25.60	28.53	9.44*	22.00*	23.90*
QA†	17.80*	7.46*	21.93*	14.61*	4.26*	17.09*	11.00	27.60	28.32	17.78	**34.00**	34.73
QGen‡	**32.45**	**13.48**	**37.23**	**30.32**	**9.36**	**34.53**	**11.79**	**32.00**	**33.34**	**17.97**	32.40	**36.11**
TERM/HYBRID MODELS												
BM25*	45.12*	**20.66**	50.33*	38.61*	11.94*	42.78*	15.41*	37.60	39.21	16.23*	31.20*	35.16*
QGenHyb‡	**46.78**	20.60	**52.16**	**41.73**	**12.84**	**46.18**	**18.19**	**40.80**	**43.92**	**21.97**	**39.60**	**43.91**

Table 3: Zero-shot first-stage retrieval. Unsupervised*; Out-of-domain†; Synthetic‡. Bold=Best in group. Statistically significant differences (permutation test, $p < 0.05$) from the last row of each group are marked by *.

	MRR	**Prec@1**
BM25*	6.63*	1.84*
ICT*	4.62*	1.58*
Ngram*	7.22*	3.05*
QA†	11.14*	4.35*
QGen‡	14.93	**6.21**
QGenHyb‡	**16.73**	6.05
Supervised	*33.68*	*17.33*

Table 4: Zero-shot ad-hoc retrieval for Natural Questions. Unsupervised*; Out-of-domain†; Synthetic‡. Bold=Best; Underline=Best non-hybrid. Baselines with statistically significant differences (permutation test, $p < 0.05$) from QGen are marked by *.

	B1	B2	B3	B4	B5	Avg.
BM25	31.7	27.8	40.4	40.1	41.8	36.3
QGen	28.9	20.3	30.7	29.0	33.1	28.4
QGenHyb	34.8	31.3	43.4	41.9	45.3	39.3
AUEB-1	33.6	31.8	**44.4**	40.1	46.0	39.2
pa	33.5	**33.0**	43.5	36.0	**48.3**	38.9
bioinfo-3	34.0	31.7	43.7	40.2	46.7	39.2
DeepR-test	30.7	29.1	43.5	39.8	47.5	38.1
BM25→resc.	33.9	29.2	42.4	42.5	457	38.7
QGenHyb→resc.	**37.5**	31.2	43.0	**43.6**	46.6	**40.4**

Table 5: MAP for zero-shot models (above dashed lined) vs. supervised models (below dashed line) on BioASQ8 document retrieval. B1-B5 is batch 1-5.

6.2 Learning Curves

Since our approach allows us to generate queries on every passage of the target corpus, one question is that whether retrieval system trained this way simply memorizes the target corpus or it also generalize on unseen passages. Furthermore, from an efficiency standpoint, how many synthetic training examples are required to achieve maximum performance. To answer these questions, we uniformly sample a subset of documents and then generate synthetic queries only on that subset. Results on BIOASQ 7 are shown in Figure 4, where x-axis denotes the percentage of sampled documents. We can see that retrieval accuracy improves as passage coverage increases. The peak is achieved when using a 20% subset, which covers 21% of the reference passages. This is not surprising because the number of frequently discussed entities/topics are typically limited, and a subset of the passages covers most of them. This result also indicates that the learned system does generalize, otherwise optimal performance would be seen with 100% of the data.

6.3 Generation vs. Retrieval Quality

Another interesting question is how important is the quality of the question generator relative to retrieval performance. Below we measured generation quality (via Rouge-based metrics (Lin and Hovy, 2002)) versus retrieval quality for three systems. The base generator contains 12 transformer layers, the lite version only uses the first 3 layer. The large one contains 24 transformer layers and each layer with larger hidden layer size, 4096, and more attention heads, 16. Retrieval quality was measured on BIOASQ 7 and generation quality with a held out set of the community question-answer data set. Results are shown in Table 6. We can see that larger generation models lead to improved generators. However, there is little difference in retrieval metrics, suggesting that large domain targeted data is the more important criteria.

7 Conclusion

We study methods for neural zero-shot passage retrieval and find that domain targeted synthetic question generation coupled with hybrid term-neural first-stage retrieval models consistently outperforms alternatives. Furthermore, for at least one domain, approaches supervised quality. While out of the scope of this study, future work includes further testing the efficacy of these first-stage models in a full end-to-end system (evaluated briefly in Section 6.1), as well as for pre-training supervised models (Chang et al., 2020).

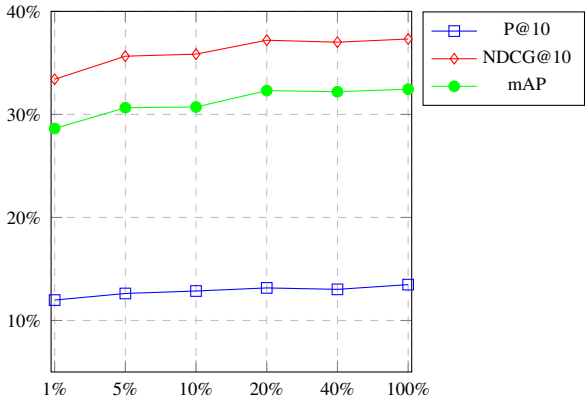

Figure 4: MAP on BioASQ7 (y-axis) w.r.t. the % of documents used for synthesizing queries (x-axis).

	Generation		Retrieval		
	Rouge 1	**Rouge L**	**MAP**	**Prec @10**	**nDCG @10**
Lite	23.55	21.90	32.50	**13.48**	37.23
Base	26.20	24.23	**32.86**	13.42	**37.96**
Large	**26.81**	**24.90**	32.61	13.34	37.53

Table 6: Generation quality vs. retrieval metrics.

Acknowledgements

We thank members of Google Research Language for feedback on this work. In particular, Gonçalo Simões gave detailed feedback on an early draft of this work, and Shashi Narayan evaluated the question generation quality.

References

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. ReQA: An evaluation for end-to-end answer retrieval models. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 137–146, Hong Kong, China. Association for Computational Linguistics.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. *arXiv preprint arXiv:1906.05416*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Sumit Bhatia and Prasenjit Mitra. 2010. Adopting inference networks for online thread retrieval. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1300–1305. AAAI Press.

Alexey Borisov, Ilya Markov, Maarten de Rijke, and Pavel Serdyukov. 2016. A neural click model for web search. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 531–541, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Georgios-Ioannis Brokos, Polyvios Liosis, Ryan McDonald, Dimitris Pappas, and Ion Androutsopoulos. 2018. Aueb at bioasq 6: Document and snippet retrieval. *arXiv preprint arXiv:1809.06366*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 129–136, New York, NY, USA. Association for Computing Machinery.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.

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

Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. 2005. Using odp metadata to personalize search. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185.

Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W. Bruce Croft. 2018. Cross domain regularization for neural ranking models using adversarial learning. *CoRR*, abs/1805.03403.

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*.

Zhuyun Dai and Jamie Callan. 2020. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1533–1536. ACM.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the ACM Web Search and Data Mining Conference*, pages 126–134.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *CoRR*, abs/1811.08008.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2020. MultiReQA: A cross-domain evaluation for retrieval question answering models. *arXiv preprint arXiv:2005.02507*.

Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Trans. Inf. Syst.*, 36(4).

David Hawking. 2004. Challenges in enterprise search. In *ADC*, volume 4, pages 15–24. Citeseer.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A position-aware neural IR model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1058, Copenhagen, Denmark. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019a. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019b. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, page 45–51, USA. Association for Computational Linguistics.

Jimmy Lin. 2019. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*. ACM New York, NY, USA.

Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with graphs. In *Proceedings of the International Conference on Machine Learning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR conference on*

research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020, pages 1573–1576. ACM.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge university press.

Ryan McDonald, George Brokos, and Ion Androutsopoulos. 2018. Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1860, Brussels, Belgium. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019a. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira and Kyunghyun Cho. 2019b. Passage re-ranking with bert. *ArXiv*, abs/1901.04085.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2793–2799. AAAI Press.

Dimitris Pappas, Ryan McDonald, Georgios-Ioannis Brokos, and Ion Androutsopoulos. 2019. AUEB at BioASQ 7: document and snippet retrieval. In *Proceedings of the BioASQ Workshop*.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the ACM Internation Conference on Information and Knowledge Management*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2019. Leveraging pre-trained checkpoints for sequence generation tasks. *CoRR*, abs/1907.12461.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Aliaksei Severyn, Massimo Nicosia, Gianni Barlacchi, and Alessandro Moschitti. 2015. Distributional neural networks for automatic resolution of crossword puzzles. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 199–204, Beijing, China. Association for Computational Linguistics.

Chirag Shah and Jefferey Pomerantz. 2010. Evaluating and predicting answer quality in community qa. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 411–418.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. 2018. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*.

Brandon Tran, Maryam Karimzadehgan, Rama Kumar Pasumarthi, Mike Bendersky, and Don Metzler. 2019. Domain adaptation for enterprise email search. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:138.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008. Curran Associates, Inc.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. 2020. Cord-19: The covid-19 open research dataset. *arXiv preprint arXiv:2004.10706*.

S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. 2016. Understanding data augmentation

for classification: When to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6.

Xiang Wu, Ruiqi Guo, David Simcha, Dave Dopson, and Sanjiv Kumar. 2019. Efficient inner product approximation in hybrid spaces. *arXiv preprint arXiv:1903.08690*.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. 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)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019b. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernández Ábrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019c. Multilingual universal sentence encoder for semantic retrieval. *CoRR*, abs/1907.04307.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3481–3487.

Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the ACM Internation Conference on Information and Knowledge Management*, pages 497–506.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

A Data

Statistics on each evaluation set are listed in Table 7. Document collection of "BioASQ" comes from MEDLINE articles, and we remove roughly 10M articles that only contains a title. Furthermore, for BioASQ 7B and BioASQ 8B we only keep articles published before 2018 December 31 and 2019 December 31, respectively. On "Forum", we remove threads with empty posts. On "NQ" since there is at most one passage annotated as relevant for each question, and we also remove questions that have no answer, thus the number of questions equal to the number of reference passages. Besides zero-shot experiments, we also conduct supervised experiments on NQ, where we randomly samples 5% question from the training data as development set. This yields a training and development set with 70,393 and 3,704 (question, passage) pairs, respectively.

The data resources can be downloaded from the following websites

- BioASQ: `http://participants-area.bioasq.org/`
- Forum: `http://sumitbhatia.net/source/datasets.html`
- Natural Questions: `https://github.com/google/retrieval-qa-eval`
- Pubmed / Medline: `https://www.nlm.nih.gov/databases/download/pubmed_medline.html`
- Stackexchange: `http://archive.org/details/stackexchange`
- Yahoo! Answers: `http://webscope.sandbox.yahoo.com/catalog.php?datatype=l`

	BioASQ7	BioASQ8	NQ	ForumTravel	ForumUbuntu
Q	500	500	1772	25	25
R	2349	1646	1772	1,538	1,188
C	50M	53.4M	29.5M	82,669	106,642

Table 7: Statistics on each evaluation set. "Q" denotes the number of unique questions. "R" denotes the total number of annotated reference passages. "C" is the number of passages in the target collection.

- BioBERT: https://github.com/dmis-lab/biobert
- BERT: https://github.com/google-research/bert

To the extent that we pre-process the data, we will release relevant tools and data upon publication.

B Question Generation Details

Our question generation follows the same implementation of Rothe et al. (2019). Both the encoder and decoder share the same network structure. Parameter weights are also shared and are initialized from a pretrained RoBERTa (Liu et al., 2019) checkpoints. Training data is processed with sentencepiece (Kudo and Richardson, 2018) tokenization. We truncate answers to 512 sentencepiece tokens, and limit decoding to at most 64 steps. The training objective is the standard cross entropy. We use Adam (Kingma and Ba, 2014) with learning rate of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.997$ and $\epsilon = 1e-9$. Learning rate warmup over the first 40,000 steps. Training batch size for the "lite", "base" and "large" models are 256, 128 and 32 respectively. All models are trained on a "4x4" slice of v3 Google Cloud TPU. At inference, results from using beam search decoding usually fall short of diversity, thus we use greedy decoding to speed up question generation.

C Neural Model Details

C.1 Zero shot Retrieval Models

C.1.1 Development Set

Since we are investigating zero-shot scenario where there is no annotated development set available, hyperparameters are set by following best practice reported in previous work. We thus do not have development set numbers. However, in the hyperparameters section below, we do use a subset of the zero-shot training data to test training convergence under different parameters.

C.1.2 Data Generation

For ICT task, we follow Lee et al. (2019b) and randomly select at most 5 sentences from a document, with a mask rate of 0.9. For Ngram models, Gysel et al. (2018) suggests that retrieval models trained with N larger than 16 consistently outperform those trained with N smaller than 8. In addition, further increase N from 16 has little effect on retrieval accuracy. Thus, in our experiment we set N to 16 and move the ngram window with a stride of 8 to allow 8 token overlap between consecutive ngrams. For QGen models, each passage is truncated to 512 sentence tokens and feed to the question generation system. Besides, we also run question generator on individual sentences from each document to promote questions that focus on different aspects of the same document. In particular, we select at most top 5 salient sentences from a document, where salience of a sentence is measure as the max IDF value of terms in that sentence. We then feed these sentences to the question generator.

C.1.3 Hyperparameters

For zero-shot neural retrieval model training, we uniformly sample of a subset of 5K (question, document) pairs from the training data as a noisy development set. Instead of finding the best hyperparameter values, we use this subset to find the largest batch size and learning rate that lead the training to converge (Smith et al., 2018). Take batch size for example, we always start from the largest batch that can fit in the memory of a "8x8" TPU slice. We gradually decrease the batch size by a factor of 2 if the current value causes training diverge. More details of hyperparameter values of each task are listed in Table 8. Note on Forum data, the maximum batch size for QGen is much larger than other tasks. Looking into the data, we found that queries generated by ICT or Ngram task on Forum data tends to contain higher percentage of noisy sentences or ngrams that are either ill-relevant to the topic or too general. For example, "suggestions are welcomed", "any ideas for things to do or place to stay". We train each model for 10 epochs, but also truncate training steps to 200,000 to make training time tractable.

For BM25, the only two hyperparameters are k and b. We set these to $k = 1.2$ and $b = 0.75$ as advised by Manning et al. (2008).

For the hybrid model QGenHyb, the only hyperparameter is λ. We set this to 1.0 without any tuning, since this represented an equal trade-off between the two models and we wanted to keep the systems zero-shot. However, we did try experimentations. For BioASQ 8b and Forum Ubuntu, values near 1.0 were actually optimal. For BioASQ

		Learning Rate	Batch Size
BioASQ	ICT	1e-5	8192
	Ngram	1e-5	8192
	QGen	1e-5	8192
Forum Travel	ICT	2e-6	1024
	Ngram	2e-6	1024
	QGen	2e-6	4096
Forum Ubuntu	ICT	1e-6	512
	Ngram	1e-6	512
	QGen	1e-6	4096
NQ	ICT	1e-5	6144
	Ngram	1e-5	6144
	QGen	1e-5	6144

Table 8: Hyperparameters

7b and Forum Travel, values of 2.0 and 2.1 were optimal and led to improvements in MAP from $0.468 \rightarrow 0.474$ and $0.181 \rightarrow 0.188$, respectively.

C.2 Supervised Models

We also train supervised models on BioASQ and NQ, where we use the development set to do early stopping. For BioASQ, our developement set is data from BioASQ 5 (i.e., disjoint from BioASQ 7 and 8). The development set MAP of our supervised model reranking a BM25 system on this data is 52.1, compared to the BioASQ 8 scores of 38.7. For NQ, the MRR on the development set is 0.141. All other hyperparameters remain the same except we use a smaller batch size of 1024, as we observe that using large batch causes the model quickly overfit the training data. This may due to the number of training examples is 2 orders of magnitude smaller compared to zero-shot setting. For our BioASQ supervised model we follow Pappas et al. (2019) and train it with binary cross-entropy using the top 100 BM25 results as negatives.

C.3 Computational Resources

C.3.1 Question Generation

To train the question generator on 2M questions,

- We used a "4x4" slice of v3 Google Cloud TPU.
- Training time ranges from 20 hours for the lite model and 6 days for the large model.

Once trained, we need to run the generator over our passage collection.

- We distributed computation and used 10,000 machines (CPUs) over the collection.
- For BioASQ, the largest dataset, it took less than 40 hours to generate synthetic questions.

We initialize question generation models from either RoBerta base or Roberta large checkpoint (Liu et al., 2019), and the total number of trainable parameters are 67M for the lite model, 152M for the base model and 455M for the large model.

C.3.2 Neural Retrieval Model

To train the retrieval models, we need to train the query and passage encoders. We share parameters between the two encoders and initialize them from either base BERT (Devlin et al., 2019) or BioBERT (Lee et al., 2019a) checkpoint. Thus retrieval models trained on BioASQ have 108M trainable parameters and retrieval models trained on NQ and Forum data have 110M trainable parameters. After training, we need to run the passage encoder over every passage in the collection to create the nearest neighbour backend.

- Depending on the training batch size, we use either an "8x8" or "4x4" TPU slice.
- Training the "ngram" model on BioASQ took the longest time, which completes in roughly 30 hours.
- Indexing BioASQ, which is our largest passage collection, with 4000 CPUs which took roughly 4 hours.

Having trained models, the inference task is to encode a query with the neural model and query the distributed nearest neighbour backend to get the top ranked passages. The relevant resources are:

- We encode queries on a single CPU.
- Our distributed nearest neighbour search uses 20 CPUs to serve the collections.
- For BioASQ, our largest collection, to run the inference on the test sets of 500 queries took roughly 1m57s. This is approximately 0.2s per instance to encode the query, run brute-force nearest neighbour search on 10s of millions of examples and return the result.