Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin[*], **Barlas Oğuz**[*], **Sewon Min**[†], **Patrick Lewis**,
Ledell Wu, Sergey Edunov, Danqi Chen[‡], **Wen-tau Yih**
Facebook AI [†]University of Washington [‡]Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

Abstract

Open-domain question answering relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method. In this work, we show that retrieval can be practically implemented using *dense* representations alone, where embeddings are learned from a small number of questions and passages by a simple dual-encoder framework. When evaluated on a wide range of open-domain QA datasets, our dense retriever outperforms a strong Lucene-BM25 system greatly by 9%-19% absolute in terms of top-20 passage retrieval accuracy, and helps our end-to-end QA system establish new state-of-the-art on multiple open-domain QA benchmarks.[1]

1 Introduction

Open-domain question answering (QA) (Voorhees, 1999) is a task that answers factoid questions using a large collection of documents. While early QA systems are often complicated and consist of multiple components (Ferrucci (2012); Moldovan et al. (2003), *inter alia*), the advances of reading comprehension models suggest a much simplified two-stage framework: (1) a context *retriever* first selects a small subset of passages where some of them contain the answer to the question, and then (2) a machine *reader* can thoroughly examine the retrieved contexts and identify the correct answer (Chen et al., 2017). Although reducing open-domain QA to machine reading is a very reasonable strategy, a huge performance degradation is often observed in practice[2], indicating the needs of improving retrieval.

Retrieval in open-domain QA is usually implemented using TF-IDF or BM25 (Robertson and Zaragoza, 2009), which matches keywords efficiently with an inverted index and can be seen as representing the question and context in high-dimensional, sparse vectors (with weighting). Conversely, the *dense*, latent semantic encoding is *complementary* to sparse representations by design. For example, synonyms or paraphrases that consist of completely different tokens may still be mapped to vectors close to each other. Consider the question *"Who is the bad guy in lord of the rings?"*, which can be answered from the context *"Sala Baker is best known for portraying the villain Sauron in the Lord of the Rings trilogy."* A term-based system would have difficulty retrieving such a context, while a dense retrieval system would be able to better match *"bad guy"* with *"villain"* and fetch the correct context. Dense encodings are also *learnable* by adjusting the embedding functions, which provides additional flexibility to have a task-specific representation. With special in-memory data structures and indexing schemes, retrieval can be done efficiently using maximum inner product search (MIPS) algorithms (e.g., Shrivastava and Li (2014); Guo et al. (2016)).

However, it is generally believed that learning a good dense vector representation needs a large number of labeled pairs of question and contexts. Dense retrieval methods have thus never be shown to outperform TF-IDF/BM25 for open-domain QA before ORQA (Lee et al., 2019), which proposes a sophisticated inverse cloze task (ICT) objective, predicting the blocks that contain the masked sentence, for additional pretraining. The question encoder and the reader model are then fine-tuned using pairs of questions and answers jointly. Although ORQA successfully demonstrates that dense retrieval can outperform BM25, setting new state-of-the-art results on multiple open-domain

[*]Equal contribution

[1]The code and trained models have been released at https://github.com/facebookresearch/DPR.

[2]For instance, the exact match score on SQuAD v1.1 drops from above 80% to less than 40% (Yang et al., 2019a).

QA datasets, it also suffers from two weaknesses. First, ICT pretraining is computationally intensive and it is not completely clear that regular sentences are good surrogates of questions in the objective function. Second, because the context encoder is not fine-tuned using pairs of questions and answers, the corresponding representations could be suboptimal.

In this paper, we address the question: can we train a better dense embedding model using only pairs of questions and passages (or answers), *without* additional pretraining? By leveraging the now standard BERT pretrained model (Devlin et al., 2019) and a dual-encoder architecture (Bromley et al., 1994), we focus on developing the right training scheme using a relatively small number of question and passage pairs. Through a series of careful ablation studies, our final solution is surprisingly simple: the embedding is optimized for maximizing inner products of the question and relevant passage vectors, with an objective comparing all pairs of questions and passages in a batch. Our *Dense Passage Retriever* (DPR) is exceptionally strong. It not only outperforms BM25 by a large margin (65.2% vs. 42.9% in Top-5 accuracy), but also results in a substantial improvement on the end-to-end QA accuracy compared to ORQA (41.5% vs. 33.3%) in the open Natural Questions setting (Lee et al., 2019; Kwiatkowski et al., 2019).

Our contributions are twofold. First, we demonstrate that with the proper training setup, simply fine-tuning the question and passage encoders on existing question-passage pairs is sufficient to greatly outperform BM25. Our empirical results also suggest that additional pretraining may not be needed. Second, we verify that, in the context of open-domain question answering, a higher retrieval precision indeed translates to a higher end-to-end QA accuracy. By applying a modern reader model to the top retrieved passages, we achieve comparable or better results on multiple QA datasets in the open-retrieval setting, compared to several, much complicated systems.

2 Background

The problem of open-domain QA studied in this paper can be described as follows. Given a factoid question, such as "*Who first voiced Meg on Family Guy?*" or "*Where was the 8th Dalai Lama born?*", a system is required to answer it using a large corpus of diversified topics. More specifically, we assume the extractive QA setting, in which the answer is restricted to a span appearing in one or more passages in the corpus. Assume that our collection contains D documents, $d_1, d_2, \cdots, d_D$. We first split each of the documents into text passages of equal lengths as the basic retrieval units[3] and get M total passages in our corpus $\mathcal{C} = \{p_1, p_2, \ldots, p_M\}$, where each passage p_i can be viewed as a sequence of tokens $w_1^{(i)}, w_2^{(i)}, \cdots, w_{|p_i|}^{(i)}$. Given a question q, the task is to find a span $w_s^{(i)}, w_{s+1}^{(i)}, \cdots, w_e^{(i)}$ from one of the passages p_i that can answer the question. Notice that to cover a wide variety of domains, the corpus size can easily range from millions of documents (e.g., Wikipedia) to billions (e.g., the Web). As a result, any open-domain QA system needs to include an efficient *retriever* component that can select a small set of relevant texts, before applying the reader to extract the answer (Chen et al., 2017).[4] Formally speaking, a retriever $R : (q, \mathcal{C}) \rightarrow \mathcal{C}_\mathcal{F}$ is a function that takes as input a question q and a corpus $\mathcal{C}$ and returns a much smaller *filter set* of texts $\mathcal{C}_\mathcal{F} \subset \mathcal{C}$, where $|\mathcal{C}_\mathcal{F}| = k \ll |\mathcal{C}|$. For a fixed k, a *retriever* can be evaluated in isolation on *top-k retrieval accuracy*, which is the fraction of questions for which $\mathcal{C}_\mathcal{F}$ contains a span that answers the question.

3 Dense Passage Retriever (DPR)

We focus our research in this work on improving the *retrieval* component in open-domain QA. Given a collection of M text passages, the goal of our dense passage retriever (DPR) is to index all the passages in a low-dimensional and continuous space, such that it can retrieve efficiently the top k passages relevant to the input question for the reader at run-time. Note that M can be very large (e.g., 21 million passages in our experiments, described in Section 4.1) and k is usually small, such as 20–100.

3.1 Overview

Our dense passage retriever (DPR) uses a dense encoder $E_P(\cdot)$ which maps any text passage to a d-dimensional real-valued vectors and builds an index for all the M passages that we will use for retrieval.

[3] The ideal size and boundary of a text passage are functions of both the retriever and reader. We also experimented with natural paragraphs in our preliminary trials and found that using fixed-length passages performs better in both retrieval and final QA accuracy, as observed by Wang et al. (2019).

[4] Exceptions include (Seo et al., 2019) and (Roberts et al., 2020), which *retrieves* and *generates* the answers, respectively.