C-MORE: Pretraining to Answer Open-Domain Questions by Consulting Millions of References

Anonymous ACL submission

Abstract

We consider the problem of pretraining a two-stage open-domain question answering (QA) system (retriever + reader) with strong transfer capabilities. The key challenge is how to construct a large amount of high-quality question-answer-context triplets without task-specific annotations. Specifically, the triplets should align well with downstream tasks by: (i) covering a wide range of domains (for open-domain applications), (ii) linking a question to its semantically relevant context with supporting evidence (for training the retriever), and (iii) identifying the correct answer in the context (for training the reader). Previous pretraining approaches generally fall short of one or more of these requirements. In this work, we automatically construct a large-scale corpus that meets all three criteria by consulting millions of references cited within Wikipedia. The well-aligned pretraining signals benefit both the retriever and the reader significantly. Our pretrained retriever leads to 2%-10% absolute gains in top-20 accuracy. And with our pretrained reader, the entire system improves by up to 4% in exact match.

1 Introduction

Open-domain question answering (QA) aims to extract the answer to a question from a large set of passages. A simple yet powerful approach adopts a two-stage framework (Chen et al., 2017; Karpukhin et al., 2020), which first employs a retriever to fetch a small subset of relevant passages from large corpora (i.e., *retriever*) and then feeds them into a *reader* to extract an answer (text span) from them. Due to its simplicity, a sparse retriever such as TF-IDF/BM25 is generally used together with a trainable reader (Min et al., 2019). However, recent advances show that transformer-based dense retrievers trained on supervised data (Karpukhin et al., 2020) can greatly boost the performance, which better captures the semantic relevance between the question and the correct passages. Such approaches, albeit promising, are restricted by the limited amount of human annotated training data.

Inspired by the recent progresses of language models pretraining (Devlin et al., 2019; Lee et al., 2019; Guu et al., 2020; Sachan et al., 2021), we would like to address the following central question: *can we pretrain a two-stage open-domain QA system (retriever + reader) without task-specific human annotations?* Unlike general language models, pretraining such a system that has strong transfer capabilities to downstream open-domain QA tasks is challenging. This is mainly due to the lack of well-aligned pretraining supervision signals. In particular, we need the constructed pretraining dataset (in the form of question-answer-context triplets) to: (i) cover a wide range of domains (for open-domain applications), (ii) link a question to its semantically relevant context with supporting evidence (for training the retriever), and (iii) identify the correct answer in the context (for training the reader).

There have been several recent attempts in addressing these challenges. ORQA (Lee et al., 2019) creates pseudo query-passage pairs by randomly sampling a sentence from a paragraph and treating the sampled sentence as the question while the rest sentences as the context. REALM adopts a retrieve-then-predict approach, where the context is dynamically retrieved during training and an encoder (reader) predicts the masked token in the question based on the retrieved context. The retriever pretraining signals constructed in these approaches are not aligned with question-context pairs in open-domain QA settings. For example, as shown in Figure 1, the context (in blue color) of ORQA pretraining data instance does not contain direct supporting evidence to the question. Likewise, the dynamically retrieved context in REALM cannot be guaranteed to contain direct supporting evidence either. In addition, existing pretraining methods (Lee et al., 2019; Guu et al., 2020) mostly focus on the retriever and do not jointly provide

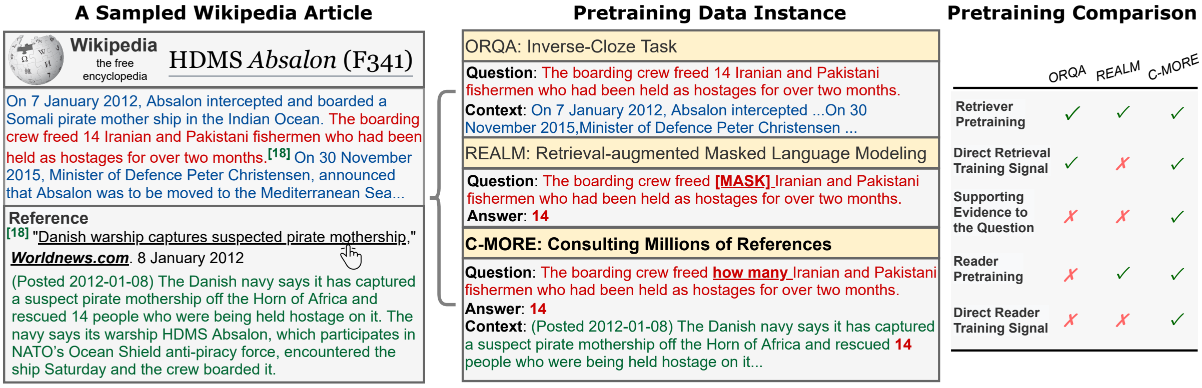

Figure 1: Different pretraining methods for open-domain QA. Our **C-MORE** pretrains both retriever and reader by using direct signals extracted from millions of references cited in the verified knowledge source.

direct pretraining signals for the reader (Figure 1).

To meet all three aforementioned criteria, we propose a pretraining approach named **C**onsulting **M**illions **O**f **RE**ferences (**C-MORE**), which automatically constructs pretraining data with well-aligned supervision signals (Figure 1). Specifically, we first extract three million statement-reference pairs from Wikipedia along with its cited references. Then, we transform them into question-answer-context triplets by replacing a potential answer span in the statement (e.g., *"14"* in the Figure 1) by an interrogative phrase (e.g, *"how many"*). Such kind of pseudo triplets are in the exact same form as human-annotated ones, and the question is linked to the context that contains the most direct-supporting evidence, a highly desirable feature for open-domain QA tasks. We experiment the pretraining with a widely-adopted open-domain QA system, Dense Passage Retriever (DPR) (Karpukhin et al., 2020). The experimental results show that our pretrained retriever not only outperforms both sparse and dense retrieval baselines in the zero-shot retrieval setting (2%-10% absolute gain in top-20 accuracy), but also leads to further improvement in the downstream task fine-tuning. By integrating with our pretrained reader, the entire open-domain pretraining improves the end-to-end QA performance by 4% in exact match.

2 Method

Recall that we want to automatically construct a large-scale open-domain QA pretraining dataset that satisfies three criteria: (i) The dataset should cover a wide range of domains for the open-domain QA purpose. (ii) The context passage is semantically relevant to the question and contains direct supporting evidence for answering the question. (iii) The correct answer span in the context passage for answering the question should also be identified for training the reader. This section first discusses how to extract a large amount of statement-reference pairs from the Wikipedia and then explain how to construct pseudo question-answer-context triplets for pretraining open-domain QA systems.

2.1 Statement-Reference Pairs Collection

Wikipedia articles usually contain a list of knowledge sources (references) at the end that are verified by human editors to support the statements in the articles (Li et al., 2020). And the reference documents always consist of strong supporting evidence to the statements. For example, as shown in Figure 1, the document (in green color) contains the direct evidence *"...rescued 14 people who were being held hostage on it..."* to support the query (red text) *"The boarding crew freed 14 Iranian and Pakistani fishermen who had been held as hostages over two months"*. Additionally, such knowledge sources are often organized in a good structure and can be automatically extracted and processed. Moreover, the statement-reference pairs in Wikipedia cover a wide range of topics and domains. Thus, when converted into question-context pairs, they satisfy the first two criteria and are suitable for training an accurate dense retriever at a large scale.

In our study, we extract around six million statement-reference pairs from Wikipedia. We filter the pairs whose reference documents are not reachable and finally obtain around three million statement-reference pairs (see statistics in Appendix Table 3). The data collection method we proposed is very general and therefore can be easily extended to other domains, e.g., WikiEM (wikem.org) for medical domain or other languages, e.g., Baidu Baike (baike.baidu.com) for Chinese.

2.2 QAC Triplets Construction

We now explain how to further convert the statement-reference pairs into question-answer-context pairs. Inspired by previous unsupervised extractive QA work (Lewis et al., 2019), we extract entities as potential answers to construct pseudo question-answer-context pairs where an answer span is extracted from the context given an question to accommodate the extractive QA setting. Specifically, we first adopt an off-the-shelf named entity recognition tool spaCy (Honnibal and Montani, 2017) to identify entities in each query. Next, we filter the entities that do not appear in the evidence based on string matching. If multiple entities are found, we sample one of them as the potential answer to the query. The sampled entity in the query is replaced by an interrogative phrase based on the entity type (e.g., a `[DATE]` entity will be replaced by phrases such as *"when"*, *"what time"*, *"what date"*. In this way, we can construct question-answer-context triplets to train open-domain QA models. See more question reformation rules in Appendix Table 4).

3 Experiment

3.1 Experimental Setup

Pretraining Model Architecture. Since conceptually the construed triplets is in the same format as the annotated QA data, they can be used to pretrain any existing neural open-domain QA model. Here, we adopt DPR (Karpukhin et al., 2020), which consists of a dual-encoder as the retriever and a BERT reader, considering its effectiveness and popularity. Specifically, the retriever first retrieves top-k (up to 400 in our experiment) passages, and the reader assigns a passage score to each retrieved passage and extracts an answer with a span score. The span with the highest passage selection score is regarded as the final answer. The reader and retriever can be instantiated with different models and we use `BERT-base-uncased` for both of them following (Karpukhin et al., 2020).

Pretraining Data Processing. For our extracted pseudo question-answer-context triplets, sometimes the context (reference document) is too long to fit into a standard BERT (maximum 512 tokens) in the DPR model. Thus, we chunk a long document into 128-word text blocks with a stride of 64. Then we calculate relevance scores (using BM25) of the derived blocks with the question and select the most relevant block as the context. Note that the retrieval step is done within the single document (usually less than 10 text blocks). In contrast, the baseline model (Section 3.2) - sparse retriever BM25 - looks up the entire knowledge corpus (20M text blocks). In this way, we can automatically collect the most relevant context that directly supports the query from a long article.

Finetuning QA Datasets. We consider three popular open-domain QA datasets for finetuning: NaturalQuestions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), and WebQuestions (WebQ) (Berant et al., 2013), whose statistics are shown in Appendix (Table 3).

Following the setting of DPR (Karpukhin et al., 2020), we use the Wikipedia as the knowledge source and split Wikipedia articles into 100-word units for retrieval. All the datasets we use are the processed versions from the DPR implementation.

Implementation Details. For pretraining, we set training epochs to 3, batch size to 56 for retrievers and 16 for readers, and learning rate to 2e-5. We select the best checkpoint based on the pretraining dev set. For finetuning, we use the same set of hyperparameters as the original DPR paper. For comparing baselines ORQA and REALM, we replicate the results based on their released checkpoints.

3.2 Retrieval Performance

We consider three settings to demonstrate the usefulness of our pretrained retriever.

Unsupervised. We assume no annotated training QA pairs are available. In this setting, We compare our method with existing unsupervised retrievers: a sparse retriever BM25 and two pretrained dense retrievers ORQA and REALM.

Domain Adaptation. We consider the condition in which there are QA training pairs in the source domain but no training data in the target domain. The task is to obtain good retrieval performance on the target test set only using source training data. We compare our method with two baselines: one is to directly train a dense retriever on the source domain while the other is to first pretrain a dense retriever on our constructed corpus and then finetune it on the source domain training set.

Supervised. In this setting, all the annotated QA training instances are used. Similar to the previous setting, we compare a supervised retriever with and without our **`C-MORE`** pretraining.

For all settings, we report the top-k retrieval accuracy ($k \in \{20, 100\}$) on the test set following

Settings	Methods	Training Data	Top-20 Accuracy			Top-100 Accuracy		
			NQ	TQA	WebQ	NQ	TQA	WebQ
Unsupervised	BM25	-	59.1*	66.9*	55.0*	73.7*	76.7*	71.1*
	ORQA (Lee et al., 2019)	Wikipedia	32.5	52.5	38.6	51.4	70.5	58.9
	REALM (Guu et al., 2020)	CCNews	57.1	67.6	58.3	72.2	78.3	75.6
	C-MORE	**Wikipedia**	**61.9**	**72.2**	**62.7**	**75.8**	**81.3**	**78.5**
Domain Aaptation	DPR-NQ	NaturalQuestion	-	69.7	69.0	-	79.2	78.8
	+ w/ C-MORE	+ Wikipedia	-	**72.8**	**71.2**	-	**81.6**	**81.3**
	DPR-TQA	TriviaQA	69.2	-	71.5	80.3	-	81.0
	+ w/ C-MORE	+ Wikipedia	**71.0**	-	**74.3**	**81.7**	-	**83.2**
	DPR-WebQ	WebQ	56.1	66.1	-	70.7	77.6	-
	+ w/ C-MORE	+ Wikipedia	**67.3**	**74.2**	-	**79.2**	**82.6**	-
Supervised	DPR-supervised	Supervised Data	78.4*	79.4*	73.2*	85.4*	85.0*	81.4*
	+ w/ C-MORE	+ Wikipedia	**80.3**	**81.3**	**75.0**	**86.7**	**85.9**	**83.2**

Table 1: Overall retrieval performance of different models. Results marked with "*" are from DPR (Karpukhin et al., 2020), and "-" means it does not apply to the current setting.

(Karpukhin et al., 2020). See the overall retrieval performance of different models in each setting in Table 1. We have the following observations.

In the **unsupervised** setting, compared with the strong sparse retrieval baseline BM25, our pretrained dense retriever shows significant improvement. For example, we obtain around 7% absolute improvement in terms of both Top-20 and Top-100 accuracy on the WebQuestion dataset. Compared with pretrained dense retrievers (i.e., ORQA and REALM), our pretrained model outperforms them by a large margin. This is not surprising as our pretraining data contain better aligned retrieval supervision signals: reference documents often have supporting evidence for the question while their retrieval training signals are relatively indirect.

In the **domain adaptation** and **supervised** settings, our pretrained dense retriever provides a better finetuning initialization and leads to improvement compared with randomly initialized DPR models. Another surprising result is that our pretrained dense retriever even outperforms some DPR domain adaptation models. For example, on the TriviaQA testing set, our pretrained DPR model achieves 72.2% top-20 and 81.3% top-100 accuracy while the DPR-NQ model obtains 69.7% and 79.2% respectively. This indicates that our pretrained dense retriever can generalize well even without using any annotated QA instances.

All the results demonstrate the usefulness and generalization of our pretrained dense retriever for open-domain QA tasks.

3.3 End-to-End QA performance

We now examine how our pretrained retriever and reader improve the end-to-end QA performance, measured in exact match (EM). The results are shown in Table 2, from which we make the following observations. (i) Surprisingly, our fully-unsupervised system (pretrained retriever + pretrained reader) shows a certain level of open-domain QA ability (see row #1). For example, on TriviaQA, our fully-unsupervised system can answer around 25% of questions correctly. (ii) Compared to the system with BM25 retriever (row #2), the one with our pretrained dense retriever (line #3) retrieves more relevant passages, leading to better QA performance. (iii) Initializing either the retriever or the reader from our pretrained checkpoint can lead to further improvement (rows #4-#6). For example, on the TriviaQA and WebQuestion datasets, our entire pipeline pretrain leads to about 4% absolute gain in terms of EM.

Row	Retriever		Reader		NQ	TQA	WebQ
	P	F	P	F			
1	✓	✗	✓	✗	11.3	24.8	4.5
2	✗	✗	✗	✓	32.6	52.4	29.9
3	✓	✗	✗	✓	**35.3**	**55.1**	**32.1**
4	✗	✓	✗	✓	41.5	56.8	34.6
5	✓	✓	✗	✓	**41.9**	58.6	35.6
6	✓	✓	✓	✓	41.6	**60.3**	**38.6**

Table 2: End-to-end QA performance based on different retrievers and readers. P: Pretraining. F: Finetuning. The retriever of Row 2 is BM25, which does not involve either pretraining or finetuning.

4 Conclusion

This paper proposes an effective approach for pretraining open-domain QA systems. Specifically, we automatically construct three million pseudo question-answer-context triplets from Wikipedia that align well with open-domain QA tasks. Extensive experiments show that pretraining a widely-used open-domain QA model (DPR) on our constructed data achieves promising performance gain in both retrieval and QA accuracies. Future work includes exploring the effectiveness of the constructed data on more open-domain QA models.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, pages 4171–4186. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. 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.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised qa. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard em approach for weakly supervised question answering. 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 2851–2864.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *ACL/IJCNLP 2021*, pages 6648–6662. Association for Computational Linguistics.

A Appendix

Data Type	Dataset	Train	Dev	Test
Pretraining	Our WikiRef	2.96M	40K	-
Finetuning QA Data	NaturalQuestion	58,880	8,757	3,610
	TriviaQA	60,413	8,837	11,313
	WebQuestion	2,474	361	2,032

Table 3: Statistics of pretraining and finetuning data.

NER Type	Candidate Question Phrases
CARDINAL	"what",
DATE	"when","what time", "what date",
EVENT	"what event","what", "which event",
FAC	"where","what buildings",
GPE	"where", "what country",
LANGUAGE	"what language","which language",
LAW	"which law","what law",
LOC	"where", "what location", "which place", "what place",
MONEY	"how much money","how much",
NORP	"what", "what groups", "where",
ORDINAL	"what rank","what",
ORG	"which organization", "what organization", "what",
PERCENT	"what percent", "what percentage",
PERSON	"who", "which person",
PRODUCT	"what", "what product",
QUANTITY	"how many", "how much",
TIME	"when", "what time",
WORK_OF_ART	"what", "what title"

Table 4: Question phrase replacement rules for different types of entities.