Unsupervised Open-Domain Question Answering with Higher Answerability

Anonymous ACL submission

Abstract

Open-domain Question Answering (ODQA) has achieved significant results in terms of supervised learning manner. However, data annotation cannot also be irresistible for its huge demand in an open domain. Though unsupervised QA or unsupervised Machine Reading Comprehension (MRC) has been tried more or less, unsupervised ODQA has not been touched according to our best knowledge. This paper thus pioneers the work of unsupervised ODQA by formally introducing the task and proposing a series of key data construction methods. Our exploration in this work inspiringly shows unsupervised ODQA can reach up to 86% performance of supervised ones.

1 Introduction

Open-domain Question Answering (ODQA) is the task of answering questions based on information from a very large collection of documents which has a variety of topics (Chen and Yih, 2020). Unlike Machine Reading Comprehension (MRC) task where a passage containing evidences and answers is provided for each question, ODQA is more challenging as there is no such supporting passage beforehand. ODQA systems need to go through a large collection of passages such as the whole Wikipedia to find the correct answer.

While tremendous progress on ODQA have been made based on pretrained language models such as BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), and T5 (Raffel et al., 2020), fine-tuning these language models requires large-scale labeled data, i.e., passage-question-answer triples (Lewis et al., 2019). Apparently, it is costly and practically infeasible to manually create a dataset for every new domain.

Though previous studies which have made attempts in unsupervised MRC like (Lewis et al., 2019; Li et al., 2020; Fabbri et al., 2020; Hong et al., 2020; Perez et al., 2020), as to our best knowledge, no such manner of attempts have been made in terms of ODQA. Thus in this paper, for the first time, we tackle the ODQA setting without human-annotated data, which we term Unsupervised ODQA (UODQA). Concretely, our setting is: starting from an automatically generated question or question-like sentence, we employ a lexical-based retriever like BM25 to retrieve positive passages that contain the answer and negative passages without the answer, from the Wikipedia corpus. Together with these, we can effectively train a question answering model which can handle multiple passages.

Unlike UQA which the supporting passage is certain for each question, UODQA needs to construct more than one passages through retrieval-based method and solve a multi-passage MRC problem.

As the first attempt to tackle UODQA, we propose a series of methods about how to synthesize data from a set of selected natural sentences and compare end-to-end performance, and finally our proposed method outperforms unsupervised method GPT-3 (zero-shot) (Brown et al., 2020) by a large margin and achieves up to 86% performance of previous popular supervised method DPR (Karpukhin et al., 2020) on three ODQA benchmarks.

2 Task Definition

For UODQA task, there is no limitation to use or construct data for training, only development and test sets from ODQA benchmark have to be used for evaluation and fair comparison. Therefore we will focus on data construction hereafter.

Based on a specific corpus C, several $< Q, P^+, P^-, A >$ triples are constructed, For each constructed example, Q denotes the question, A denotes the answer, P^+ denotes multiple positive passages that contains the answer supporting the question to solve, P^- denotes multiple negative passages that do not contain the answer, and help

make the model learn to distinguish distracting information. To train a reader model, these data are leveraged to learn a function $F(Q, P^+, P^-) = A$.

3 Method

3.1 Data Construction

The procedure is shown in Figure 1. The purpose is to automatically construct $< Q, P^+, P^-, A >$ triples for model training. Obviously, the quality of constructed data decides whether a model can be trained well.

Firstly, based on some specific corpus C, we select a set of sentences to construct $< Q, A >$. In ODQA, most of the questions are factoid. Many works show that knowing Named Entities (NEs) may help construct $< Q, A >$ pairs (Glass et al., 2020; Guu et al., 2020), thus a good practice is to select NEs as A in the constructed data. Meanwhile the sentence where the NE is from is Q after the selected NE is masked. The constructed Q is thus a pseudo-question, or conceptually defined as **Information Request**. In fact, both pseudo-question and real question may be used for effective model training to learn a question answering manner. Many works adopt question generation methods for better training, however, they also inevitably introduce noises.

When selecting sentences to generate Q from corpus C, previous works on UQA do not set constraints (Lewis et al., 2019; Li et al., 2020; Hong et al., 2020), which brings no guarantee the constructed information request is reasonable or answerable. Such none-guarantee will become much more severe in UODQA. Basically, the source sentences selected need to have complete information. For example, *"It was instead produced by Norro Wilson, although the album still had a distinguishable country pop sound."* is ambiguous because of too many coreferences. Moreover, when selecting phrases as A, it needs to be answerable based on the constructed information request. For example, in sentence *"Yao Ming played for the Houston Rockets of the National Basketball Association (NBA)."* if the phrase *"Yao Ming"* is selected as A, the constructed *"___ played for the Houston Rockets of the National Basketball Association (NBA)."* is not certain and answerable because there is not a short answer or named entity (which ODQA focuses on) that can be uniquely determined without ambiguity according to the context.

To obtain $< Q, A >$ pairs with higher quality, we use sentences from the dataset in (Elsahar et al., 2019), which is an alignment corpus for WikiData and natural language. Each sentence is aligned with a Subject-Predicate-Object triple, and we select the object as A. The dataset does not involve human labeling and is automatically created through a pipeline in terms of toolkits, a set of rules and several distant supervision assumptions.

Dataset	train	dev	test
Natural Questions	79,168	8,757	3,610
WebQuestions	3,417	361	2,032
TriviaQA	78,785	8,837	11,313

Table 1: Data statistics of three datasets.

To obtain P^+ and P^-, our model retrieves documents from knowledge source S, and selects the documents containing A as positive P^+ otherwise negative. This heuristic can not assure enough evidences but still make the model learn reasoning. To filter the trivial cases of P^+ that the context text surrounding the answer has too much overlap with that in the Q so the answer can be simply generated based on shortcuts, we set a window size n and check the left and right n-gram of the selected A. Thus, $< Q, P^+, P^-, A >$ triples are constructed.

3.2 Model Training

Following previous common practice in ODQA, we adopt retriever-reader architecture to perform UODQA. BM25 serves as retrieval metric in an unsupervised manner. After retrieving top K passages, a reader receives the questions and passages as input to output an answer. Following (Izacard and Grave, 2020b), we adopt a generative reader based on T5 (Raffel et al., 2020).

4 Experiments and Analysis

4.1 Evaluation Settings

The evaluation metrics are Exact Match (EM). For EM, if generated answer hits any one of the labeled list of possible golden answer, the sample is positive. The accuracy of EM is calculated as $EM = N^+/N$ where N^+ is number of positive samples and N is number of all evaluated samples.

We evaluate our model on three ODQA benchmarks, Natural Questions (Kwiatkowski et al., 2019), WebQuestions (Berant et al., 2013) and TriviaQA (Joshi et al., 2017). Statistics are shown in Table 1. The train/dev/test split follows (Lee et al., 2019). As this is an unsupervised ODQA task, we discard training set and only adopt de-

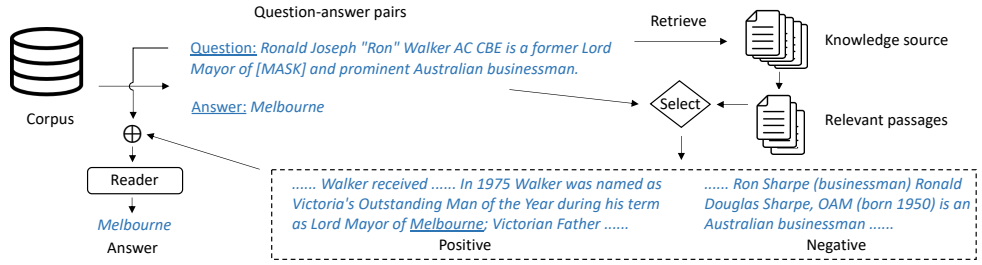

Figure 1: Our proposed method for synthesizing data and training.

velopment and test sets for evaluation. **Natural Questions(NQ)** is commonly used ODQA benchmark which was constructed according to real Google search engine queries. The answers are short phrases from Wikipedia articles containing various NEs. **WebQuestions(WQ)** contains questions collected from Google Suggest API, and the answers are all entities from the structured knowledge base (Freebase). **TriviaQA(TQA)** consists of trivia questions from online collection.

4.2 Implementation Details

We adopt dataset from (Elsahar et al., 2019) and select sentences that have only one object to construct question-answer pairs. Sentences containing character number more than 250 or less than 50 are discarded. The object is used as answer and we use the token [MASK] to replace the answer in the sentence as the question. Following (Karpukhin et al., 2020), the version of Wikipedia corpus we use is Dec. 20, 2018 dump and we split the whole corpus into 100-word segments as units of retrieval. For retrieving documents, we use Apache Lucene [1] to build index and perform BM25 retrieval. To filter P^+ using n-gram, we use n as 3. We first retrieve top 100 documents, and select the top 40 documents to construct the input for reader. If none of top 40 documents contains the answer, we further find top 41-100 documents that contains the answer, and replace the 40th document with it, otherwise this sample is discarded. Finally, we obtain 844,100 samples to train for 60 hours using 8 nVidia V100s.

We implement the reader following (Izacard and Grave, 2020b) and perform training using learning rate of 1e-4, batch size of 256 and the number of concatenated passages each sample is 40. The model size setting we use is T5-base. We save and evaluate the model checkpoint every 500 training steps and stop training if the performance does not increase any more in 5 evaluations, and the checkpoint of best EM score is selected.

[1] https://lucene.apache.org/

		WQ	NQ	TQA
sup.	DPR(2020)	42.4	41.5	57.9
	FiD(2020b)	-	**51.4**	**67.6**
unsup.	GPT3 ZeroShot (2020)	14.4	14.6	-
ours	RandSent_{10}	12.01	15.90	40.39
	RandEnt_{10}	15.01	18.14	45.38
	QuesGen_{10}	10.43	13.88	43.44
	OurMethod_{10}	16.14	18.73	46.64
	OurMethod_{50}	**18.60**	**20.69**	**50.23**

Table 2: Experimental results EM on test set of three datasets. "sup." denotes supervised methods and "unsup." denotes unsupervised methods. The subscript denotes number of passages to input the reader.

4.3 Results

In Table 2, the subscripts *10* and *50* denote number of passages to input the reader. As shown in this table, we perform experiments based on four kinds of settings, to study to what extent the quality of constructed training data affects performance. The difference among four settings *RandSent*, *RandEnt*, *QuesGen* and *OurMethod* is how to sample the *question* and *answer*. *OurMethod* refers to what we discuss in Section 3.1 and it is to use the alignment corpus to help sampling. *RandSent* means we select random sentences from Wikipedia articles and NEs to construct question-answer pairs. *RandEnt* means we use data from (Elsahar et al., 2019) and select a random NE from each sentence as answer. This expands the scope of types of answers and makes the model learn more diversified knowledge. *QuesGen* means we perform a question generation step after obtaining the constructed data based on our method. This makes the expression of the pseudo-question more close to the real question and makes the model learn a question answering manner better, but it may hurts the reasonibility of constructed questions because of the noise introduced by question generation methods. Some examples are shown in Table 3.

Setting // Question // Answer
RandSent// He had 16 caps for Italy, from 1995 to [MASK], scoring 5 tries, 25 points in aggregate. // 1999
RandEnt// [MASK] stiphra is a species of sea snail, a marine gastropod mollusk in the family Raphitomidae. // Daphnella
QuesGen// What is a multi-state state high-way in the New England region of the United States, running across the southern parts of New Hampshire, Vermont and Maine, and numbered, owned, and maintained by each of those states? // Route 9
OurMethod// Ronald Joseph "Ron" Walker AC CBE is a former Lord Mayor of [MASK] and prominent Australian businessman. // Melbourne

Table 3: Examples of the settings for comparison experiments.

We evaluate the degree of answerability through manual labeling. Consider four levels of answerability which are scored as 0, 1, 2 and 3, 3 means the answer can be constrained as a very specific kind of thing and 0 means there is no idea what the pseudo-question is talking about (eg. have too many ambiguous pronouns) or the contextual clues are too insufficient to infer the answer. We randomly sample 100 examples of each setting *RandSent*, *RandEnt* and *OurMethod*, and the average scores are 1.16/3, 1.86/3 and 2.3/3 respectively.

As shown in Table 2, our model outperforms GPT-3 (Brown et al., 2020) by a large margin (+4.2% on WebQuestions and +6.09% on NaturalQuestions) and achieves up to 86% of popular supervised method DPR's performance. Besides, improving the quality of constructed training data improves the performance by a large margin. Moreover, the performance gap between supervised and unsupervised method indicates that the task is very challenging and shows huge space for improvements.

4.4 Analysis and Discussion

There are three main factors of the differene among different settings, reasonability, answerability and strategy to select answer span. Reasonability indicates to what extent the question conforms to the expression of natural language, answerability means whether the sentence describes an fact with accurate meanings and has enough information for deducing answer, and strategy to select answer span determines what knowledge the model learns.

For the setting of *RandSent*, because random sentences are usually ambiguous and lack enough evidence to infer corresponding answer, the answerability is very weak. For the setting of *RandEnt*, though the original sentence contains complete information and expresses accurate fact, the randomly masked NE may be too difficult to deduce. Compared with this, our strategy that only selects the object as answer is better, because in the structure of Subject-Predicate-Object, the object usually can be accurately deduced. *QuesGen* attempts to reform the expression of question to make it more like a real question, however, it also introduces noise to do harm to the performance. For the purpose of implementing unsupervised manner, we only adopt simple rule-based question-generation method, which applies semantic role labeling on the original question and selects one of the parsed argument as answer, and converts the order and tense of the sentence to reform it as a question expression. It indicates that if the question generation method introduces too much noise and hurts the reasonability of sentences too much, it is even worse than doing nothing and maintaining the statement expression of original constructed sentences.

In this paper, we exploit the possibility of completely discarding the training set to train an ODQA model, however, our results may be further improved when the training set can be adopted for further training or refining the reader like a course-to-fine process.

We note that many sampling methods for ODQA were raised in recent years such as ICT (Lee et al., 2019) and batch negtive (Karpukhin et al., 2020), and they aim at training a better dense retriever and need to further fine-tune on the training data. However, our paper focuses on exploring what extent the ODQA system can achieve without high-quality human-annotated QA data for the first time. Besides, many works on Unsupervised Question Answering are proposed in recent years, though we aim at solving more challenging ODQA problem where no passage is given, the approaches of UQA may be adopted to further improve our method. We will leave this for future work.

5 Conclusion

In this paper, we first propose the task of Unsupervised Open-domain Question Answering, and explore to what extent it can perform based on our suggested data construction methods. We compare several strategies for synthesizing better data, as a result achieve up to 86% performance of previous supervised method. We hope this work inspires a new line of ODQA in the future and helps build more practical readers for real use.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *EMNLP*.

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.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Elena Simperl, and Frederique Laforest. 2019. T-rex: A large scale alignment of natural language with knowledge base triples.

Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. *arXiv preprint arXiv:2004.11892*.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G P Shrivatsa Bhargav, Dinesh Garg, and Avi Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782, Online. 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*.

Giwon Hong, Junmo Kang, Doyeon Lim, and Sung-Hyon Myaeng. 2020. Handling anomalies of synthetic questions in unsupervised question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3441–3448.

Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

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, Florence, Italy. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980*.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised qa. *arXiv preprint arXiv:2005.02925*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

A Appendix

A.1 Related Work

A.1.1 Open-Domain Question Answering

Open-domain Question Answering (ODQA) needs to find answers from tremendous open domain information such as Wikipedia or web pages. Traditional methods usually adopt **retriever-reader** architecture (Karpukhin et al., 2020), which is to first retrieve relevant documents and then generate answers based on these retrieved documents, which is the main focus of our paper. Besides, there is also end-to-end method (Guu et al., 2020), but it costs too much computation resources to be widely applied. The improvements of **retriever** (Izacard and Grave, 2020a) and **reader** (Izacard and Grave, 2020b) are both critical for the overall performance, and there is still huge room for improvements.

A.1.2 Unsupervised Question Answering

Unsupervised Question Answering (UQA) is to alleviate the problem of huge cost of data annotation. Generally speaking, the key issue of UQA aims at automatically generating context-question-answer triples from publicly available data. (Lewis et al., 2019) uses an Unsupervised Neural Machine Translation method to generate questions. (Fabbri et al., 2020) proposes to retrieve relevant sentence that contains the answer and reform the sentence with template-based rules to generate questions. (Li et al., 2020) proposes an iterative process to refine the generated questions turn by turn. (Hong et al., 2020) proposes paraphrasing and trimming methods to respectively solve the problem of word overlap and unanswerable generated questions.