Can Generative Pre-trained Language Models Serve as Knowledge Bases for Closed-book QA?

Cunxiang Wang[♠♣*], **Pai Liu**[♣*] **and Yue Zhang**[♣♡†]
[♠]Zhejiang University, China
[♣]School of Engineering, Westlake University, China
[♡]Institute of Advanced Technology, Westlake Institute for Advanced Study, China
{wangcunxiang, zhangyue, liupai}@westlake.edu.cn

Abstract

Recent work has investigated the interesting question using pre-trained language models (PLMs) as knowledge bases for answering open questions. However, existing work is limited in using small benchmarks with high test-train overlaps. We construct a new dataset of closed-book QA using SQuAD, and investigate the performance of BART. Experiments show that it is challenging for BART to remember training facts in high precision, and also challenging to answer closed-book questions even if relevant knowledge is retained. Some promising directions are found, including decoupling the knowledge memorizing process and the QA finetune process, forcing the model to recall relevant knowledge when question answering.

1 Introduction

Large-scare pre-trained language models (PLMs) such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018) have significantly improved the performance of NLP tasks (Radford et al., 2019). There is increasing evidence showing that PLMs contain world knowledge (Petroni et al., 2019; Zhou et al., 2020; Talmor et al., 2020). As a result, recent research considers generative PLMs such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a) for **Closed-book QA**, which has only question-answer pairs without external knowledge source. For example, after being finetuned on a few QA pairs, a generative LM can directly output *"Florence"* after being given the question *"Where was Dante born?"*. Roberts et al. (2020) find that generative PLMs can store and use knowledge as they can achieve relatively high performance in closed-book QA task on three datasets. However, Lewis et al. (2020b) find that the excellent results are mainly due to high question/answer overlap rates between training and testing data.

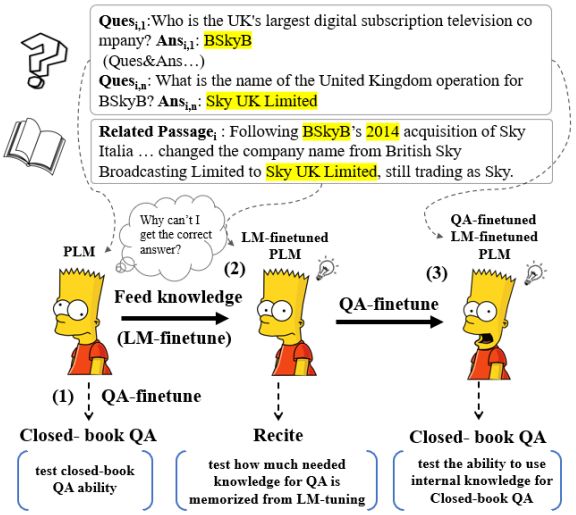

Figure 1: Process of generative PLMs for closed-book QA. (1) BART performs poorly on closed-book QA after QA finetuning; (2) We LM-finetune BART with related passages to feed knowledge and use a reciting task to evaluate how much knowledge the LM-finetuned model memorizes; (3) Though memorizing most needed knowledge, BART still faces challenge on closed-book QA after QA finetuning.

Existing research leaves many open questions on the potential of generative pre-trained LMs on closed-book QA. For example, the used datasets consist of question-answer pairs only, and there is no mechanism to control what factual knowledge is already used to train a generative PLM before taking the closed-book questions. In addition, the high overlapping rates between training and testing questions and answers make it difficult to understand whether the answer that a model gives comes from its inherent knowledge or superficial cues in training data. To address these issues, we make a new benchmark of question-answer pairs from SQuAD (Rajpurkar et al., 2018), where each ques-

[*]Equal contribution
[†]The corresponding author

tion has a corresponding Wikipedia passage as a traceable knowledge source for pre-training. We find that despite giving around 25% accuracy on existing test sets (i.e., WebQuestions and TriviaQA), BART gives only 1.5% accuracy on the SQuAD dataset.

This result shows that there is still much challenge in using BART for closed-book QA directly. We further investigate the reason by separately examining whether BART can remember factual knowledge accurately, and whether it can make use of remembered knowledge to answer questions. The general process of investigating these two issues is presented in Figure 1.

For the first issue, we use related passages in SQuAD to further extra pre-train BART, which we call as **LM-finetuning**, and test the ratio of retained factual knowledge using a language modeling task, which we call as **reciting**. Results show that as the number of training passages grows, BART demonstrates severe issues of forgetting, losing track of exact facts in the LM task. For example, when the number of passage is around 500, BART can memorize 66% needed knowledge. But when the number of passage increases to about 5000, the ratio becomes 4%.

For the second issue, we use versions of LM-finetuned BART that can retain the majority of factual knowledge for further QA finetuning, by constraining the number of passages. Although all the training and testing questions concern the passages in LM-finetuning, BART still fails to answer the majority of questions. This demonstrates difficulties in making use of internal knowledge for QA. In addition, further experiments show that QA finetuning can negatively influence the retained factual knowledge as measured using the original LM task.

While reporting such challenges, we also find some promising directions by using simple data augmentation tricks. For example, simply adding related passages to test outputs can help BART retrieve relevant factual knowledge and give the correct answer. In addition, rather than treating QA finetuning in the same way as LM pre-training (Roberts et al., 2020), decoupling the LM pre-training task and the QA finetuning tasks can also allow a model to better retain factual knowledge through the QA-finetuning task. [1]

[1]We have released the code and dataset at https://github.com/wangcunxiang/Can_PLM_Server_as_KB for future study.

	Train Set	Dev Set	Test Set
WebQuestions	3778	1016	1016
TriviaQA	961091	4975	4976
NaturalQuestions	107369	900	900

(a) The QA pairs of three datasets.

	Train Set	Dev Set	Test Set
SQuAD	86396(19035)	2968(602)	2930(602)

(b) The QA pairs and passages statistics of SQuAD. The numbers in () are the passage amounts.

Table 1: Details of each dataset after our processing.

Models \ Dataset	SQuAD	WB	TQ	NQ
original BART-Large → QA-finetune	1.5%	30.0%	24.9%	23.0%
original BART-Large → pre-trained with all passages → QA-finetune	1.8%	-	-	-

Table 2: Closed-book QA performance of BART on four datasets. For SQuAD, only QA pairs are used in this experiments. WB, TQ and NQ means WebQuestions, TriviaQA and NaturalQuestions, respectively.

2 Using SQuAD for Closed-book QA

In the closed-book QA task (Roberts et al., 2020), a model needs to answer questions without external resources. Formally, the input is a question q, and the output is a sequence of tokens o. For evaluation, the correct golden answer g will be compared with o. Previous work (Roberts et al., 2020) uses the Exact Match (EM) metric to score o against g.

We conduct closed-book QA by using the **BART** model (Lewis et al., 2020a) on four datasets-WebQuestions (Berant et al., 2013), TriviaQA (Joshi et al., 2017), NaturalQuestions (Kwiatkowski et al., 2019) and SQuAD2 (Rajpurkar et al., 2018). BART is a transformer-based (Vaswani et al., 2017) sequence-to-sequence generative PLM, which we choose because it has achieved several state-of-the-art results on generative tasks. We use the publicly released checkpoint BART-Large in this work.[2]

To use a generative PLM on each dataset, the model is first finetuned using the training question-answer pairs. We call this process as **QA-finetuning**. While the other three datasets are used by following previous work (Roberts et al., 2020), we make a novel adaptation of the SQuAD dataset for closed-book QA. SQuAD (Rajpurkar et al., 2018) is a wildly-adopted QA dataset typically for extractive QA, where the input is a question to-

[2]https://huggingface.co/facebook/bart-large/tree/main

Dataset\Overlap Type	Answer Overlap	Question Overlap
NaturalQuestions	61.5%	32.5%
TriviaQA	78.7%	33.6%
WebQuestions	59.3%	27.5%
SQuAD	24.0%	1.0%

Table 3: Question and Answer Overlaps on four datasets. Question overlaps data of NaturalQuestions, TriviaQA and WebQuestions are from Lewis et al. (2020b); Answer overlaps on the three datasets are a bit different from Lewis et al. (2020b) because of our dataset pre-processing.

Dataset \ Overlap Type	O^{test} Overlap with G^{train}	G^{test} Overlap with G^{train}
WebQuestions	88.5%	59.3%
SQuAD	39.8%	24.0%

Table 4: Overlap analysis between test outputs/golden answers and training answers. We select the top-performing results to analyze.

gether with a passage containing the answer fact, and the answer is a span from the passage. However, no previous work has used SQuAD for closed-book QA yet. Compared to other QA datasets, SQuAD is the most suitable for our setting, containing corresponding passages, lower test-train overlap, and receiving more research attention. To apply SQuAD on closed-book QA, we only use QA pairs for input and output when QA-finetuning. For TriviaQA and WebQuestions, many questions have multiple answers. In order to align with the other two data sets, we split one question with several answers into several same questions with one answer when training, and take one test output as correct if it appears in the answer list when testing. As the test sets of SQuAD, NaturalQuestions and TriviaQA are not fully publicly released yet and WebQuestions does not have a development set, we split the development set of the three datasets and the test set of WebQuestions into two subsets to serve as a new development set and a new test set. We report performance on the new test sets in Table 2 while analyzing the overlaps on the two subsets together in Table 4 and Table 5. The details of four datasets after our pre-processing are shown in Table 1.

Previous work shows that T5 and BART can achieve promising results (Roberts et al., 2020; Lewis et al., 2020b) on WebQuestions, TriviaQA and NaturalQuestions. However, recently, Lewis et al. (2020b) find that the high performance is mainly because the three datasets have severe **test-train overlap** problems. In particular, we use **answer overlap** to denote the situation where the answer a in a test (q, a) pair exists in training answers, and the term **question overlap** to denote the fact that a training question with similar meaning can be found for q. To analyze whether SQuAD has the same problem, we also compute the overlap of it. Answer overlap can be easily calculated. For question overlap, following Lewis et al. (2020b), we first randomly sample 1,000 (q, a) pairs from the SQuAD test set. Then for each test question, we automatically select SQuAD training questions whose answer is a sub-sequence of the test answer. Then we ask three human experts to find whether the test q overlaps with any training question.

	Overlap	Non-Overlap
Correct	29.8% (604)	0.2% (5)
Incorrect	58.7% (1189)	11.3% (228)

(a) On WebQuestions

	Overlap	Non-Overlap
Correct	1.3% (77)	0.1% (6)
Incorrect	38.5% (2272)	60.1% (3530)

(a) On SQuAD

Table 5: Overlap analysis of test outputs on WebQuestions and SQuAD by BART. In the result cells, we present both percentages and case numbers. We select the top performing result to analyze.

The breakdown statistics are given in Table 3. SQuAD has much fewer test-train overlapped cases than the other three datasets. For example, only around 1% of SQuAD test questions overlap with training questions while the number is around 30% in the other three datasets.

2.1 Results

The overall QA results on the four datasets are shown in the first row of Table 2. BART achieves relatively high results on the three datasets WebQuestions, TriviaQA, and NaturalQuestions. However, it performs poorly on SQuAD in closed-book QA, with only 1.5% accuracy. We also use SQuAD passages to further pre-train BART and then conduct QA-finetuning. The result is shown in the second row of Table 2, the performance is 1.8% a bit better than 1.5% but still extremely low.

According to Lewis et al. (2020b), the results are influenced by test-train overlap rates. For simplicity, we define the set of gold standard answers in the train set as G^{train}, the set of gold standard answers in the test set as G^{test}. We define the set of output answers of BART on the test set as O^{test}, the set of output answers which are correct as $O^{correct}$.

To further investigate how overlap influences

BART's outputs, we choose WebQuestions as the high-overlap dataset representative to compare with the low-overlap dataset SQuAD. Results are shown in Table 4. The O^{test} of BART on WebQuestions have an 88.5% overlap with G^{train}, which is a decisive proportion. However, the G^{test} have only 59.3% overlap with G^{train}. For BART on SQuAD, the ratios are 39.8% to 24.9%, which is relatively less severe. This indicates that if testing questions have a large overlap with training questions, the model tends to generate the targets and words in the train set.

We further measure the relationship between how correct/incorrect outputs and overlap/non-overlap with G^{train}. The results are shown in Table 5, 604 of $O^{correct}$ of BART on WebQuestions overlap with G^{train}, and only 5 instances of $O^{correct}$ do not exist in G^{train}. However, all the five non-overlapping O^{test} on WebQuestions are combinations of words of G^{train} and question words, which can be viewed as a mild type of overlap. The situation is similar but sightly better on SQuAD. These results indicate that it is much easier for BART to answer correctly by superficial cues than by using its internal knowledge.

3 Task Design

The original purpose of previous research (Petroni et al., 2019; Roberts et al., 2020) is to use pre-trained language models (PLMs) as knowledge bases (KBs) and answer questions according to internal knowledge the model contains. However, if the model tends to match test questions with training questions for retrieving answers, then the source of knowledge is restricted to training questions. This deviates from the ultimate goal.

We are interested in quantitatively measuring the capability of pre-trained model in closed-book QA using its own internal knowledge from pre-training. This capability can be broken down into two components. First, the capability of a memorizing knowledge from pre-training. Second, the ability of retrieving memorized knowledge for question answering. We show investigations and report the results in the two sections below.

3.1 Procedure

As shown in Figure 2, our design is motivated by classroom teaching. A teacher first teaches the content of a textbook and then asks the student to recite the important points of the book in order

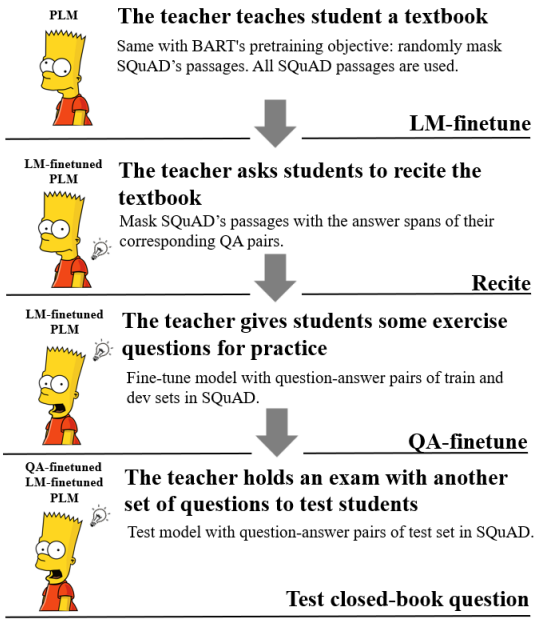

Figure 2: The main task design. The lower right bold context of each process are names of this process. The bold context in the upper middle of each process is the corresponding process in the classroom teaching. The middle context is the purpose of this process. The left icon represent the state of the model.

Models \ Dataset	ALL SQuAD (20279)
random-initialized BART	0.0%
original BART	2.2%
BART → LM-finetuning	2.7%

Table 6: The reciting performance on all SQuAD passages. We use the BART-Large checkpoint. LM-finetuning and reciting are both conducted on the same 20279 passages.

to test how well they know the book. Next, the teacher gives the student some exercise questions for practice. Finally, the teacher gives a different set of exam questions to test the student. Note that the whole book is taught and recited, rather than a split of the book, and the exercise questions and exam questions are all related to the book.

Section 4 (**Knowledge Memory**) corresponds the teaching and reciting processes in the classroom teaching. Section 5 (**Question Answering**) corresponds the practice and exam processes.

4 Knowledge Memory

To investigate whether BART can acquire and store knowledge from raw corpus, we use passages from SQuAD to finetune the BART model, which we call **LM-finetuning**. This period can be seen as

Models \ Dataset	20	160	547	1094	1641	6020
original BART	1.5%	5.2%	3.6%	3.2%	2.9%	2.2%
BART → LM-finetuning	87.3%	72.6%	66.3%	34.3%	14.0%	3.9%
BART → LM-finetuning (Added Prefix/Suffix)	85.5%	79.6%	59.5%	40.4%	15.8%	4.0%

Table 7: Performance of reciting. We use the BART-Large checkpoint. For the header of each column, the numbers stand for passage amounts of the subset. Note that LM-finetuning and reciting are both conducted on the same passages. The last row of this table will be discussed in Section 5.3

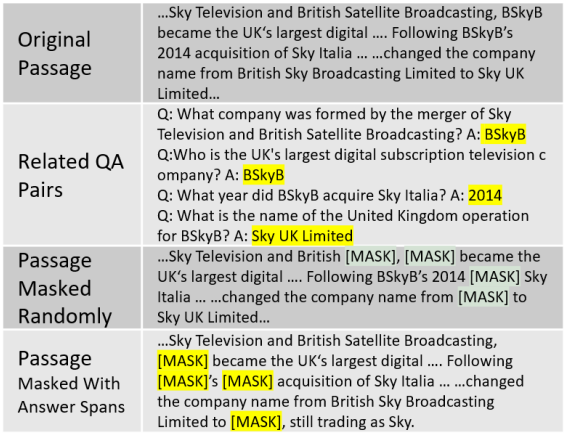

Figure 3: Examples of two types of MASK policies in training and testing periods of LM-finetuning. The passage masked randomly is for training and the passage masked with answer spans is for testing (reciting).

feeding knowledge into BART. Then we test the model to examine how much knowledge BART can memorize. We also call this testing process as **reciting**.

Training of LM-finetuning. We follow the original training objective of BART for the MLM-finetune step, which is a denoising auto-encoding process. The original BART training objective involves five operations, namely token masking, sentence permutation, document rotation, token deletion and text infilling (Lewis et al., 2020a). We only adopt token infilling in this work because it shows benefits on all downstream tasks (Lewis et al., 2020a). In addition, the sentence permutation task is shown harmful for tasks despite only being useful for text summarization (Lewis et al., 2020a). For each input passage, we randomly mask 30% tokens following Lewis et al. (2020a). An example is shown in the third row of Figure 3. We ask the model to recover the passage as the output, and use the output and the original passage to compute loss.

Testing of LM-finetuning (Reciting). In testing period of LM-finetuning, we develop a task called **'Reciting'** to probe how much (specific) knowledge the model has. Inspired by Petroni et al. (2019) and Talmor et al. (2020), who ask discriminative PLMs to fill masks of given masked passages/sentences, our reciting task is to give a generative PLM several masked passages and ask it to recover them. For each passage, we mask the token spans which are answers of related questions. An example is shown in the last row of Figure 3. In this way, we can assume that if the BART can recover the specific-masked passages, it must have the knowledge needed for further QA. Note that doing training for LM-finetuning, the masked tokens are randomly chosen, following BART (Lewis et al., 2020a). Besides, because the answer spans are mostly entities or independent knowledge segments, it is relatively less likely for models to recover them by heuristics or superficial cues. It is natural to do reciting to probe the model's internal knowledge since it is most related to the Masked Language Model process (LM-finetuning and BART's pre-training task).

Evaluation Metrics. We use the accuracy of masked spans recovery to measure how much knowledge the model memorizes. Because many answer spans appear several times in passages, we cannot simply treat the presence of the span as correct. In addition, even when the masked token is generated correctly, if its contextual words change, the meaning of the sentence may be different. Considering these, we choose a more strict evaluation metric for the reciting accuracy. We treat a span as correctly predicted only if subsequent words after the current mask and before the next mask (or the subsequent 10 tokens if the span between masked tokens is more than 10) are also correctly predicted.

4.1 Results

We first conduct reciting experiments on all SQuAD passages using the original BART, a random-initialized BART and a LM-finetuned BART. The results are shown in Table 6. The random-initialized BART gives zero accuracy, demonstrating that the task is difficult and there

is no possibility of guessing. The original BART scores 2.2%, showing that it contains certain but limited knowledge. The LM-finetuned BART gives 2.7% accuracy. This result shows that LM-finetuning is useful to a certain extent. However, despite that 100% knowledge is given, LM-finetuning only increases the result by 0.5%, demonstrating that BART faces significant challenges in memorizing important knowledge contained in pre-training SQuAD texts.

Given above observations, we try to reduce the challenge by producing smaller datasets by extracting subsets from SQuAD. The subsets include 20, 160, 547, 1094, 1641, 6020 passages, respectively, where the three numbers indicate the passage amounts. For these reciting experiments, we consider only the original and LM-finetuned BART.

The results are shown in the first two rows of Table 7. We can find that (1) using LM-finetuning, BART can memorize some knowledge. For example, when passage subset is 547, the original BART can only recover 3.6% masked spans correctly while the LM-finetuned BART can recover 66.3% masked spans; (2) The memorization ability quickly decreases when the passage amount increases. For example, when passage subset are 20, BART can recover 87.3% masks correctly; when it is 1094, the accuracy falls to 34.3%; when it is 6020, the accuracy is only 3.9%.

We conclude that BART has a certain ability to store (factual) knowledge, but the capacity is rather weak. If we control the number of passages for LM-finetuning, we can make sure that BART can memorize most needed knowledge. The LM-finetuned model trained on smaller subsets gives a more useful setting for testing QA abilities of BART when we are confident that relevant knowledge is retained.

5 Question Answering

We employ the settings in the first three columns in Table 7, where models can memorize at least 50% of needed knowledge, for further analyzing the relationship between memory and QA ability. For these experiments, all QA pairs come from passages that BART has been LM-finetuned on.

5.1 Overall Results

Besides Exact Match (EM) which is commonly used in previous closed-book QA work (Roberts et al., 2020; Lewis et al., 2020b), we also consider

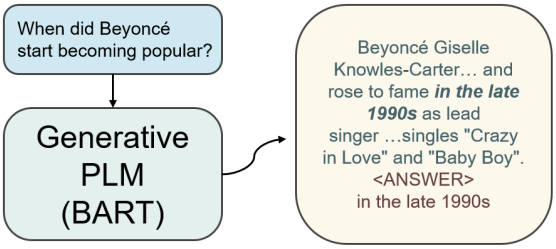

Figure 4: An intuitive approach to QA-bridge-tuning. To make the model more dependent on the internal knowledge to answer the question, the model is required to generate not only answer but also the corresponding passage. The outputs should be 'P <ANSWER> A', where 'P' stands for the corresponding passage, <ANSWER> is a special marker and the A stands for the answer.

Human Evaluation (HE) and F1 for two reasons. First, we observe that EM cannot fully indicate correctness. For example, a question is *"What century did ... ?"* and the golden answer is *"10th century"*. The model outputs "10th" which is actually correct in but taken incorrect by EM. Second, F1 can help indicate the similarity between the outputs and golden answers.

The overall results are presented in the first two rows of Table 8. According to the result of 'original BART-Large→LM-finetuning→QA-finetuning', compared to Reciting Accuracy (RA) of each model, the QA accuracy is much lower (87.3% vs 30%, 72.6% vs 6.5%, 66.3% vs 6.7% in HE). This result shows that BART's ability to use its internal knowledge to answer questions is weak. In addition, comparison between the first row and the second row shows that memorized knowledge helps the models better answer questions, though the help is not much (30% vs 0.0%, 6.5% vs 4.3%, 6.9% vs 4.9% in HE).

For the reciting-QA-accuracy gap, we propose two possible explanations, the first is that the model cannot activate related memory for question answering; the second is that the memorized knowledge is somehow corrupted during QA-finetuning.

5.2 Strengthening Memory Retrieval

Qualitative cases show that, even the model contains needed knowledge, the model does not necessarily refer to the most relevant memory for question answering after QA-finetuning. We list several this kind of examples in the 'QA-finetune' column of Table 9. For example, in the first row of Table 9, for the question *"What is Southern California often abbreviated as?"*, despite of the model

Models \ Dataset	20 (16/2/2;125/8/10)				160 (128/16/16;653/107/93)				547 (442/53/52;2334/314/306)			
	RA(%)	EM(%)	HE(%)	F1(%)	RA(%)	EM(%)	HE(%)	F1(%)	RA(%)	EM(%)	HE(%)	F1(%)
BART → QA-finetuning	1.5	0.0	0.0	11.0	5.2	2.2	4.3	6.4	3.6	1.9	4.9	7.0
BART → LM-finetuning → QA-finetuning	87.3	10.0	30.0	15.4	72.6	3.2	6.5	9.0	66.3	2.3	6.9	6.7
BART → LM-finetuning → QA-finetuning **(Added Prefix/Suffix)**	85.5	10.0	30.0	21.0	79.6	3.2	10.8	10.1	59.5	2.9	7.8	8.2
BART → LM-finetuning → **QA-bridge-tuning**	87.3	20.0	40.0	27.8	72.6	9.7	20.4	15.3	66.3	4.6	11.8	9.3
BART → LM-finetuning → **QA-bridge-tuning (Added Prefix/Suffix)**	85.5	20.0	40.0	31.7	79.6	11.8	22.6	16.3	59.5	5.6	12.7	10.3

Table 8: QA performance on three subsets of SQuAD. The numbers in headers are the passage and QA pair amounts, for example, '160 (128/16/16;653/107/93)' indicates this subset has overall 160 passages and 128/16/16 passages, 653/107/93 QA pairs in train/dev/test set, respectively. The number in RA column stands for reciting accuracy, which is the same with Table 7. The RAs in the table can show how much knowledge BART memorizes before QA-finetuning, of which values the model should achieve in QA accuracy if it can fully use internal knowledge to answer questions. The cells with bold text are our methods. EM, HE indicate Exact Match, Human Evaluation, respectively. 'BART' denotes the 'BART-Large' checkpoint.

Question&Answer	Model Output	
	QA-finetune	QA-bridge-tune
Q: What is Southern California often abbreviated as? A: SoCal	Southern California	Southern California, often abbreviated **SoCal**, is... <ANSWER> SoCal
Q: What century did the Normans first gain their separate identity? A:10th century	20th century	... distinct cultural and ethnic identity of the Normans emerged initially in the first half of the **10th century** ... <ANSWER> 10th
Q: What is the largest stadium in Australia? A: Melbourne Cricket ground	Australia Stadium	... <ANSWER> Melbourne Cricket ground
Q: When did the 1973 oil crisis begin? A: October 1973	1973	... <ANSWER> October 1973

Table 9: Four real output examples on QA-finetuning and QA-bridge-tuning by BART.

	A > B	A = B	A < B
Relevance	30.2%	53.3%	16.6%

Table 10: Human-evaluated relevance between the results using and not using QA-bridge-tune with correct answers. A > B means that A's outputs are more related to correct answers than B's, etc. A = QA-bridge-tune, B = QA-finetune in this Table.

Models \ Dataset	16/2/2	128/16/16	442/53/52
BART → LM-finetuning	87.3%	72.6%	66.3%
BART → LM-finetuning (Added Prefix/Suffix)	85.5%	79.6%	59.5%
BART → LM-finetuning → QA-finetuning	2.8%	10.9%	2.4%
BART → LM-finetuning → QA-finetuning (Added Prefix/Suffix)	5.7%	51.4%	16.2%

Table 11: Performance of reciting after QA. The numbers in the header is the passage amount of this subset. 'BART' denotes the 'BART-Large' checkpoint.

is trained with *"Southern Californi, often abbreviated **SoCal**"*, it still answers 'Southern California', which indicates that the model cannot retrieve related memory for answering questions.

We propose a simple way to strength knowledge retrieval, namely **QA-bridge-tune**, which is a extended QA-finetuning process. The process is illustrated in Figure 4, for each question input, the output concatenates the related passage with the answer. Thus, the model can explicitly recall the memorized passages when answering questions, by which QA-bridge-tune builds a bridge between QA and memorized knowledge so that the model can answer questions with learned knowledge. In addition, this method can help improve interpretability.

The results are shown in Table 8. We can see that QA-bridge-tune can help the model wake up the related memorize knowledge when QA, thus improving EM accuracy and by two or three times on baselines. In addition to answer correctness, we also consider the relevance between model outputs and golden answers regardless whether the answer is correct. For example, the question is *"The Amazon rainforest makes up what amount of Earth's rainforests?"* and the golden answer is *"over half"*, and two generated answers are "60%" and "the Amazon rainforest". They are both incor-

rect but the former is more relevant and therefore a better answer. We ask human experts to manually compare the results between using and not using QA-bridge-tuning, selecting results by using 'BART→LM-finetuning→QA-finetuning' and 'oBJ→LM-finetuning→QA-bridge-tuning' strategies on the '128/16/16' subset. The results are shown as Table 10. According to human experts, in 30.2% cases, the outputs of QA-bridge-tuning are more relevant to the golden answer than those of QA-finetuning while only in 16.6% cases, QA-finetuning is more relevant. This result shows that QA-bridge-tuning can help BART find more relevant knowledge. We also list several examples showing in Figure 9. As the example in the first paragraph of this subsection, for question *"What is Southern California often abbreviated as?"*, BART can output the corresponding passage along with the correct answer *"SoCal"* after QA-bridge-tuning. These results suggests that QA-bridge-tuning can effectively help the model recall the remembered knowledge.

5.3 Influence of QA on Memory

To explore whether QA-finetune interferes with the memory of LM-finetuned models, we use QA-finetuned models for the reciting task. The results are given in Table 11. After QA-finetuning, the models' reciting accuracy declines. We have two possible explanations for this phenomenon. First, QA-finetune process disrupts the models' internal memory with regard to representation; Second, the tasks are different, so model output space is disturbed, but the model still retains knowledge. Though we cannot qualitatively understand the influence of each reason above, isolating the QA functionality from pre-trained denoising auto-encoding can potentially address interference issues.

We experiment with a simple intuitive solution to this issue, namely to decouple the QA-finetune process and the LM-finetune process, so that the two task input/output spaces are differentiated to some extent. This is done simply in the input and output level. We add <PASSAGE>/<QUESTION> prefix tokens and </PASSAGE>/</QUESTION> suffix tokens to each input passage/question when LM-finetuning and Reciting/QA-finetuning, respectively, and also add </PASSAGE>/</ANSWER> suffix tokens to each output passage/answer.

The results are shown in the rows with (Added Prefix/Suffix) in Table 11. The reciting accuracy with prefix/suffix after LM-finetuning is not much different compared without prefix/suffix. However, the QA accuracy significantly improves when adding prefix/suffix (2.8% to 5.7%, 10.9% to 51.4%, 2.4% to 16.2% in HE). The results show that our decoupled methods can help the model distinguish the input type to find the appropriate semantic space, thus alleviating this problem. Besides, according to the comparison between the second row and the third row in Table 8, adding prefix/suffix can help models better answer questions. We suppose it is also because this method can help models distinguish the input/output space.

Models \ Dataset	**16/2/2**	**128/16/16**	**442/53/52**
original GPT-2 → LM-finetuning → QA-finetuning	0%	1.1%	1.0%

Table 12: Performance of GPT2 in the same setting as the second row of 8. The numbers in the header is the passage amount of this subset. The score is evaluated with Exact Match (EM).

5.4 GPT-3

GPT3 has also been shown to have certain capabilities to answer factual closed-book questions. As shown in Table 3.3 of Brown et al. (2020), it can achieve relatively high performance on TriviaQA in closed-book task even in zero-shot learning setting. However, it underperforms T5 (Roberts et al., 2020) in the other two datasets WebQuestions and NaturalQuestions, which indicates that super large scale pre-training is not the ultimate solution to the issue we discussed. There is also a possibility that GPT-3 has seen most test QA pairs of TriviaQA in the pre-training stage as it crawls extremely large documents from the internet.

We also apply GPT-2 to LM-finetuning and QA-finetuning, which has similar architecture, pre-training and finetune process with GPT-3. Thus we believe that they can have the same fundamental problem. The results are shown in Table 12. LM-finetuned GPT-2 has worse performance compared to LM-finetuned BART. This confirms that the architecture and the training process of GPT3/GPT-2 do not solve the problems we find using BART.

6 Related Work

There are two types of pre-trained language models (PLMs), discriminative PLMs such as BERT (Devlin et al., 2019), ELMo (Peters et al., 2018) and generative PLMs such as GPT (Radford et al.,

2018), BART (Lewis et al., 2020a). The key difference is that generative PLMs are of encoder-decoder architectures so they can generate text sequences of any length or token. An increasing number of works have shown that PLMs contains world knowledge. Petroni et al. (2019) first solves that discriminative PLMs such as BERT (Devlin et al., 2019) can be used for Cloze-style QA using a mask language modeling task without external resources, such as *"Dante was born in [MASK]."* → *"Florence"*. Their results show that PLMs have certain factual knowledge. Talmor et al. (2020) set eight types of Cloze-style QA, such as 'ALWAYS-NEVER' and 'AGE COMPARISON', to test different types of knowledge in several discriminative PLMs, including BERT and RoBERTa (Liu et al., 2019). They also use the mask language modeling task to do QA without finetuning, and results show that the evaluated PLMs indeed contain those kinds of knowledge. Wang et al. (2019); Zhou et al. (2020) adopt some discriminative PLMs on commonsense reasoning QA tasks such as ComVE (Wang et al., 2020) and Swag (Zellers et al., 2018) without finetuning, indicating the PLMs have commonsense knowledge. Bosselut et al. (2019) show that pretrained transformer models can be used to help construct commonsense knowledge graphs, such as ConceptNet (Speer and Havasi, 2012). However, Poerner et al. (2019) argue that BERT uses some superficial cues such as stereotypical characters to solve factual questions. GPT-3 (Brown et al., 2020) seems to have ability to answer factual questions in zero-shot setting, but there exists some evidence that GPT-3 is limited in storing and using knowledge (Bergdahl, 2020).

Roberts et al. (2020) firstly use closed-book QA to detect how much knowledge is in pre-trained language models' parameters. They perform experiments on three datasets WebQuestions (Berant et al., 2013), TriviaQA (Joshi et al., 2017) and NaturalQuestions (Kwiatkowski et al., 2019) by T5 model (Raffel et al., 2020). The results are relatively pleasant. However, Lewis et al. (2020b) find that the high performance of Roberts et al. (2020) is mainly due to the high test-train overlap of the three datasets rather than the model's internal knowledge. Our findings confirm the conclusions of Lewis et al. (2020b), and we further experiment with a more controlled SQuAD dataset, and discussed the weakness of BART in both memorization and knowledge retrieval. Because T5 (Raffel et al., 2020) is more resource demanding, considering the balance of effectiveness and experimental feasibility, we choose BART rather than the T5 model.

Different from closed-book QA, where no additional resource is available when answering questions, open-domain QA requires models to generate a sequence of tokens as the answer to each question by looking up related text from unstructured documents (Chen et al., 2017). Chen et al. (2017) first try to retrieve related passages from Wikipedia for each question and encode both the question and passages into the model, then output the answer. Guu et al. (2020) integrate the retrieval process into pre-training process, helping the PLMs better retrieve information from external knowledge source when needed, and finding benefits on open-domain QA task. Retriever-based models have the advantage of relieving the burden of pre-trained language models to remember every factual detail. The retrieval QA setting is slightly reminiscent to our data augmentation setting in Figure 4, but with the related passage being the input, rather than the output. In contrast, the settings we consider fully rely on a neural model for all knowledge.

SQuAD (Rajpurkar et al., 2016, 2018) is a widely-used dataset for machine reading comprehension, which is also a type of QA task. It asks models to use a text span from a given referential passage to answer questions. It is also used in other type of QA task, for example, Chen et al. (2017) adopt it in the open-domain QA task. We first apply it on closed-book QA and analyze why it is superior than other three commonly used datasets.

7 Conclusion

We investigated by using SQuAD, finding that closed-book QA is still challenging for generative pre-trained language models such as BART. The challenge lies both in remembering the knowledge details and in answering the questions after remembering the knowledge. Potential solutions include explicitly asking models to recall relevant knowledge when answering questions and decoupling LM-finetuning process and QA-finetuning process.

8 *Acknowledgement

The work was supported by NSPC 61976180. We thank Yongjing Yin, Chuang Fan, Yuchen Niu, Sara Gong, Tony Ou, Libo Qin and all reviewers for their generous help and advice during this research.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Jacob Bergdahl. 2020. No, gpt-3 is not superintelligent. it's not tricking humans, and it's not pretending to be stupid. *Medium Website*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, A. Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *ACL*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

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.

Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.

Mandar Joshi, Eunsol Choi, Daniel 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, Vancouver, Canada. Association for Computational Linguistics.

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*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020b. Question and answer test-train overlap in open-domain question answering datasets.

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. *arXiv preprint arXiv:1907.11692*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. Bert is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *ArXiv*, abs/1911.03681.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

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*, volume 30, pages 5998–6008. Curran Associates, Inc.

Cunxiang Wang, Shuailong Liang, Y. Jin, Yilong Wang, X. Zhu, and Y. Zhang. 2020. Semeval-2020 task 4: Commonsense validation and explanation. In *SEMEVAL*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026, Florence, Italy. Association for Computational Linguistics.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9733–9740.

A *Ethics / Impact Statement

Our used data is from open source datasets, including NaturalQuestions[3], TriviaQA[4], WebQuestion[5] and SQuAD2[6]. We split the development set of the NaturalQuestions, TriviaQA and SQuAD2 and the test set of WebQuestions into two subsets to serve as a new development set and a new test set. And we extract several subsets from SQuAD2 to serve as our new datasets. There is no additional data collection process.

B Frequently Asked Questions and Answers

It will continue to be updated on the arXiv version. So please feel free to ask us questions through any channels.

- Q0: Can you describe this work briefly?
- A0: Some works have proved PLMs can contain knowledge in their parameters. We want to know whether PLMs can learn knowledge through pretraining and whether PLMs can use their internal knowledge to solve problems. So we first continue pretrain (LM-finetune) the PLM with some passages and ask the PLM to recite the passages. Then, we QA-finetune the model and ask it to answer questions which relates the passages. The results show that the PLMs cannot memorize much knowledge through pretraining and it is weak of them to use internal knowledge to answer questions after finetuning.
- Q1: What is your conclusion?
- A1: The results show that the PLMs cannot memorize much knowledge through pretraining and it is weak of them to use internal knowledge to answer questions after finetuning. So, it is difficult to use PLMs as KBs in current Pre-training -¿ Fine-tuning paradigm.
- Q2: Why you randomly mask tokens during LM-finetuning while mask specific tokens during reciting?

[3] https://ai.google.com/research/NaturalQuestions

[4] http://nlp.cs.washington.edu/triviaqa/

[5] https://nlp.stanford.edu/software/sempre/

[6] https://rajpurkar.github.io/SQuAD-explorer/

- A2: For LM-finetuning, we want to explore whether PLMs can learn knowledge from pretraining. So, we set LM-finetuning the same with original pretraining process, which randomly mask tokens. For reciting, we want to link this process to the subsequent QA process, therefore, we can naturally compare the reciting accuracy and QA accuracy.

- Q3: Why don't you split train/dev/test set during knowledge memorization experiments?

- A3: We don't think it makes sense to ask the model to learn some passages and then ask it to recite others.

- Q4: If you do XXX things in knowledge memorization/question answering, it will improve the accuracy in reciting/QA, why do not you do these?

- A4: Our approach is based on the most classic Pre-training -¿ Fine-tuning paradigm. We suppose it is more valuable to use the most popular and standard paradigm when researching on this question. In addition, some methods may improve accuracy on both tasks, but we suppose it will affect the conclusion too much, after all, current results are far away from indicating strong abilities of knowledge memorizing and question answering.

- Q5: Does this paper concludes that PLMs cannot serve as knowledge bases?

- A5: Yes and no. If you simply use the Pretraining -¿ Fine-tuning paradigm, we suppose it will not work. However, if you optimize the paradigm, we suppose it is still promising to research on this topic because PLMs can indeed store and utilize knowledge.

- Q6: Can the conclusion of this paper also apply to other downstream tasks? Why you choose Closed-book QA as the representative task?

- A6: We suppose the conclusion is very likely the same with other downstream tasks. We choose Closed-book QA is because it is most direct and suitable task for explore why how much knowledge can models have, we have also considered other tasks, but none of them is as suitable as Closed-book QA.