Why Machine Reading Comprehension Models Learn Shortcuts?

Yuxuan Lai, Chen Zhang, Yansong Feng[*] **, Quzhe Huang,** and **Dongyan Zhao**
Wangxuan Institute of Computer Technology, Peking University, China
The MOE Key Laboratory of Computational Linguistics, Peking University, China
{erutan, zhangch, fengyansong, huangquzhe, zhaody}
@pku.edu.cn

Abstract

Recent studies report that many machine reading comprehension (MRC) models can perform closely to or even better than humans on benchmark datasets. However, existing works indicate that many MRC models may learn *shortcuts* to outwit these benchmarks, but the performance is unsatisfactory in real-world applications. In this work, we attempt to explore, instead of the expected comprehension skills, why these models learn the *shortcuts*. Based on the observation that a large portion of questions in current datasets have *shortcut* solutions, we argue that larger proportion of *shortcut* questions in training data make models rely on *shortcut* tricks excessively. To investigate this hypothesis, we carefully design two synthetic datasets with annotations that indicate whether a question can be answered using *shortcut* solutions. We further propose two new methods to quantitatively analyze the learning difficulty regarding *shortcut* and challenging questions, and revealing the inherent learning mechanism behind the different performance between the two kinds of questions. A thorough empirical analysis shows that MRC models tend to learn *shortcut* questions earlier than challenging questions, and the high proportions of *shortcut* questions in training sets hinder models from exploring the sophisticated reasoning skills in the later stage of training.

1 Introduction

The task of machine reading comprehension (MRC) aims at evaluating whether a model can understand natural language texts by answering a series of questions. Recently, MRC research has seen considerable progress in terms of model performance, and many models are reported to approach or even outperform human-level performance on different benchmarks. These benchmarks are designed to address challenging features, such as evidence checking in multi-document inference (Yang et al., 2018), co-reference resolution (Dasigi et al., 2019), dialog understanding (Reddy et al., 2019), symbolic reasoning (Dua et al., 2019), and so on.

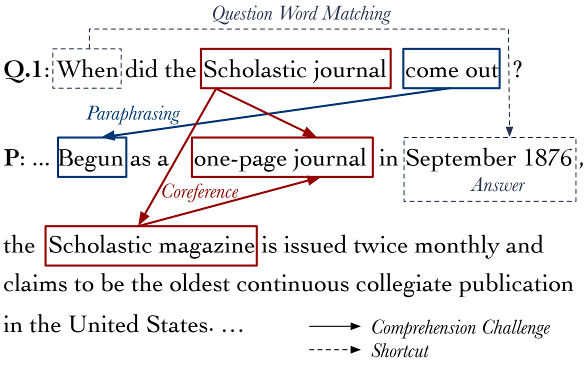
Question Word Matching
Q.1: When did the Scholastic journal come out ?
Paraphrasing
P: ... Begun as a one-page journal in September 1876,
Answer
Coreference
the Scholastic magazine is issued twice monthly and claims to be the oldest continuous collegiate publication in the United States. ...
Comprehension Challenge
Shortcut

Figure 1: An illustration of *shortcuts* in Machine Reading Comprehension. **P** is an excerpt of the original passage.

However, recent analysis indicates that many MRC models unintentionally learn *shortcuts* to trick on specific benchmarks, while having inferior performance in real comprehension challenges (Sugawara et al., 2018). For example, when answering **Q.1** in Figure 1, we expect an MRC model to understand the semantic relation between *come out* and *begun*, and output the answer, *September 1876*, by bridging the co-reference among *Scholastic journal*, *Scholastic magazine* and *one-page journal.* In fact, a model can easily find the answer without following the mentioned reasoning process, since it can just recognize *September 1876* as the only time expression in the passage to answer a *when* question. We consider such kind of tricks that use *partial* evidence to produce, perhaps unreliable, answers as ***shortcuts*** to the expected comprehension challenges, e.g., *co-reference resolution* in this

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 989–1002
August 1–6, 2021. ©2021 Association for Computational Linguistics

example. The questions with *shortcut* solutions are referred to as ***shortcut questions***. For clarity, a model is considered to have ***learned shortcuts*** when it relies on those tricks to obtain correct answers for most *shortcut* questions while performing worse on questions where challenging skills are necessary.

Previous works have found that, relying on *shortcut* tricks, models may not need to pay attention to the critical components of questions and documents (Mudrakarta et al., 2018) in order to get the correct answers. Thus, many current MRC models can be either vulnerable to disturbance (Jia and Liang, 2017), or lack of flexibility to question/passage changes (Sugawara et al., 2020). These efforts disclose the impact of *shortcut* phenomenon on MRC studies. However, concerns have been raised on **why MRC models learn these *shortcuts* while ignoring the designed comprehension challenges.**

To properly investigate this problem, our first obstacle is that there are no existing MRC datasets that are labeled whether a question has *shortcut* solutions. This deficiency makes it hard to formally analysis how the performance of a model is affected by the *shortcuts questions*, and almost impossible to examine whether the model correctly answers a question via *shortcuts*. Secondly, previous methods disclose the *shortcut* phenomenon by analyzing the model outputs through a series of carefully designed experiments, but fail to explain how the MRC models learn the *shortcuts* tricks. We need new methods to help us quantitatively investigate the learning mechanisms that make the difference when MRC models learn to answer the *shortcuts* questions and questions that require challenging reasoning skills.

In this work, we carefully design two synthetic MRC datasets to support our controlled experimental analysis. Specifically, in these datasets, each (passage, question) instance has a *shortcut* version paired with a challenging one where complex comprehension skills are required to answer the question. Our construction method ensures that the two versions of questions are as close as possible, in terms of style, size, and topics, which enable us to conduct controlled experiments regarding the necessary skills to obtain answers. We design a series of experiments to quantitatively explain how *shortcut* questions affect MRC model performance and how the models learn these tricks and challenging skills during the training process. We also propose two evaluation methods to quantify the learning difficulty of specific question sets. We find that *shortcut* questions are usually easier to learn, and the dominant gradient-based optimizers drive MRC models to learn the *shortcut* questions earlier in the learning process. The priority of fitting *shortcut* questions hinders models from exploring sophisticated reasoning skills in later stage of training. Our code and datasets can be found in `https://github.com/luciusssss/why-learn-shortcut`

To summarize, our main contributions are the following: 1) We design two synthetic datasets to study two commonly seen *shortcuts* in MRC benchmarks, *question word matching* and *simple matching*, against a challenging reasoning pattern *paraphrasing*. 2) We propose two simple methods as a probe to help investigate the inherent learning mechanism behind the different performance on *shortcut* questions and challenging ones. 3) We conduct thorough experiments to quantitatively explain the behaviors of MRC models under different settings, and show that the proportions of *shortcut* questions greatly affect model performance, which may hinder MRC models from learning sophisticated reasoning skills.

2 Synthetic Dataset Construction

To study the impact of *shortcut* questions in model training, we require the datasets to be annotated with whether each question has *shortcut* solutions, or can only be answered via complex reasoning. However, none of existing MRC datasets have such annotations. We thus design two synthetic datasets where it is known whether *shortcut* solutions exist for a question.

More importantly, we need to conduct controlled experiments and ensure, for each question, the existence of *shortcuts* solutions is the only independent variable. The extraneous variables, such as sizes of datasets, topics, answer types, and even the vocabulary, should be controlled relatively steady. Thus, in our designed datasets, each entry has a *shortcut* version instance and a challenging version. The question of the *shortcut* version can be correctly solved by a certain *shortcut* trick, while an expected comprehension skill is required to deal with the challenging version. Note that we expect the two versions of questions are as close as possible so that we can switch between the two versions freely while maintaining other factors relatively steady.

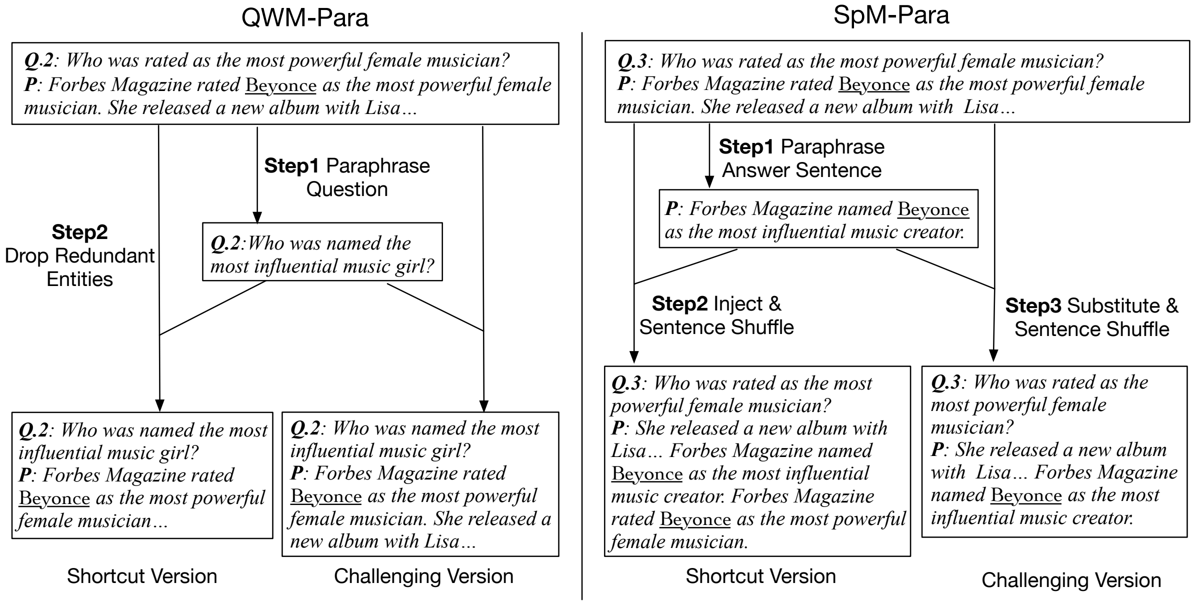
QWM-Para
Q.2: Who was rated as the most powerful female musician?
P: Forbes Magazine rated Beyonce as the most powerful female musician. She released a new album with Lisa…
Step1 Paraphrase Question
Q.2: Who was named the most influential music girl?
Step2 Drop Redundant Entities
Q.2: Who was named the most influential music girl?
P: Forbes Magazine rated Beyonce as the most powerful female musician…
Q.2: Who was named the most influential music girl?
P: Forbes Magazine rated Beyonce as the most powerful female musician. She released a new album with Lisa…
Shortcut Version
Challenging Version
SpM-Para
Q.3: Who was rated as the most powerful female musician?
P: Forbes Magazine rated Beyonce as the most powerful female musician. She released a new album with Lisa…
Step1 Paraphrase Answer Sentence
P: Forbes Magazine named Beyonce as the most influential music creator.
Step2 Inject & Sentence Shuffle
Step3 Substitute & Sentence Shuffle
Q.3: Who was rated as the most powerful female musician?
P: She released a new album with Lisa… Forbes Magazine named Beyonce as the most influential music creator. Forbes Magazine rated Beyonce as the most powerful female musician.
Q.3: Who was rated as the most powerful female musician?
P: She released a new album with Lisa… Forbes Magazine named Beyonce as the most influential music creator.
Shortcut Version
Challenging Version

Figure 2: An illustration of how the instances in the synthetic datasets are constructed from original SQuAD data. Each instance has a *shortcut* version paired with a challenging version where comprehension skills are necessary.

In this work, we focus on *paraphrasing* (***Para***) as the complex reasoning challenge, since it widely exists in many recent MRC datasets (Trischler et al., 2017; Reddy et al., 2019; Clark et al., 2019). The paraphrasing challenge requires MRC models to identify the same meaning represented in different words. Regarding the *shortcut* tricks, we study two typical kinds: *question word matching* (***QWM***) and *simple matching* (***SpM***) (Sugawara et al., 2018). For *QWM*, MRC models can simply obtain an answer phrase by recognizing the expected entity type confined by the *wh*-question words of question Q. For *SpM*, a model can find the answers by identifying the word overlap between answer sentences and the questions.

***QWM-Para* Dataset:** As elaborated in Algorithm 1, given an original instance (Q, P) from SQuAD (Rajpurkar et al., 2016), we paraphrase the question Q in Q_p to embed the *paraphrasing* challenge, and derive the corresponding *shortcut* version by dropping the sentences containing other entities with the matched type according to the question words from the given passage.

An example is shown in the left of Figure 2. In the challenging version of **Q.2**, both *Beyonce* and *Lisa* are person names which match the question word *who*. Thus, one should at least recognize the paraphrasing relationship between *named the most influential music girl* and *rated as the most powerful female musician* to distinguish between the two names to infer the correct answer. For the *shortcut* version, removing the sentence containing *Lisa* from the passage, which is also of the expected answer type *person* indicated by the question word *who*, would help a model easily get the correct answer, *Beyonce*.

Algorithm 1 Construction of *QWM-Para*

```
Input: SQuAD
Output: QWM-Para
QWM-Para ← ∅
for each instance $(Q, P)$ in SQuAD do
    if $Q$ does not start with who, when, where then
        Discard this instance.
    end if
    if the answer sentence contains other entities matching the question word then
        Discard this instance.
    end if
    Use back translation to paraphrase $Q$, obtain $Q_p$
    if the non-stop-word overlap rate between $Q_p$ and the answer sentence $> 25\%$ then
        Discard the instance.
    end if
    Delete sentences in passage $P$ that does not contain the golden answer but containing other entities matching the question word, note the modified passage as $P_s$.
    $I_s$ ← the shortcut instance version $(Q_p, P_s)$
    $I_c$ ← the challenging instance version $(Q_p, P)$
    Append the pair of questions, $(I_s, I_c)$, to QWM-Para.
end for
```

***SpM-Para* Dataset:** As shown in Algorithm 2, for instances from SQuAD, we paraphrase the answer sentences in the passage to embed the *paraphrasing* challenge and obtain its challenging version. We insert the paraphrased answer sentence in front of the original one in the passage to con-

Algorithm 2 Construction of *SpM-Para*

```
Input: SQuAD
Output: SpM-Para
$SpM\text{-}Para \leftarrow \emptyset$
for each instance $(Q, P)$ in SQuAD do
    if the non-stop-word overlap rate between $Q$ and the answer sentence $S < 75\%$ then
        Discard the instance.
    end if
    Use back translation to paraphrase the answer sentence $S$ in $P$, obtain $S_p$.
    if the answer span no longer exists in $S_p$ then
        Discard this instance.
    end if
    if the non-stop-word overlap rate between $Q$ and $S_p$ $> 25\%$ then
        Discard the instance.
    end if
    Replace $S$ in $P$ with $S_p$ and shuffle sentences, noted the modified passage as $P_c$.
    Append $S_p$ to $P$ and shuffle sentences, noted the modified passage as $P_s$.
    $I_s \leftarrow$ the shortcut instance version $(Q, P_s)$
    $I_c \leftarrow$ the challenging instance version $(Q, P_c)$
    Appen d the pair of questions, $(I_s, I_c)$, to SpM-Para.
end for
```

struct the corresponding *shortcut* version, where a model can obtain the answers by either identifying the paraphrase in the passage or using the *simple matching* trick via the original answer sentences. We randomly shuffle all sentences in the passage to prevent models from learning the pattern of sentence orders in the *shortcut* version, i.e., there are two adjacent answer sentences in the passage. Here, we assume the sentence-level shuffling operation will not affect the answers and solutions for most questions, since the supporting evidence is often concentrated in a single sentence. This can also be supported by Sugawara et al. (2020)'s findings that the performance of BERT-large (Devlin et al., 2019) on SQuAD only drops by around 1.2% after sentence order shuffling.

For example, in the *shortcut* version of **Q.3** shown in the right of Figure 2, MRC models can find the answer, *Beyonce*, either from the matching context, *rated as the most powerful female musician*, or via the paraphrased one, *named as the most influential music girl*. For the challenging version, only the paraphrased answer sentence is provided, thus, the paraphrasing skill is necessary.

Dataset Details Our synthetic training and test sets are derived from the accessible training and development sets of SQuAD, respectively. We adopt back translation to obtain paraphrases of texts (Dong et al., 2017). A sentence is translated from English to German, then to Chinese, and finally back to English to obtain its paraphrased version.[1]

The *QWM-Para* dataset contains 7072 entries, each containing two versions of (question, passage) tuples, 6306/766 for training and testing, respectively. And for *SpM-Para*, there are 8514 entries, 7562/952 for training and testing, respectively.

Quality Analysis We randomly sample 20 entries from each training set of the synthetic datasets, manually analyzing their answerability. We find that 76/80 questions could be correctly answered. The unanswerable questions result from the wrong paraphrasing. Furthermore, among the answerable questions, the paraphrasing skill is necessary in 30 out of 36 questions in the challenging version, and 36 out of 40 questions of the *shortcut* version can be correctly answered via the corresponding *shortcut* trick.

3 How the *Shortcut* Questions Affect Model Performance?

Previous efforts show that *shortcut* questions widely exist in current datasets (Sugawara et al., 2020). However, there are few quantitative analysis to discuss how these *shortcut* questions affect the model performance. A reasonable guess is that, when trained with too many *shortcut* questions, the models tend to fit the *shortcut* tricks, which are possible solutions to a large amount of questions in training. We thus argue that ***the high proportions of shortcut questions in training data make models rely on the shortcut tricks.***

One straightforward way to elaborate on this point is to observe the model performance on challenging test questions when the model is trained with different proportions of *shortcut* questions. For example, if a model trained on a dataset, in which 90% of questions are *shortcut* ones, cannot perform as well as its 10% variant on challenging test questions, that will probably indicate that higher proportions of *shortcut* questions in the training data may hinder the model from learning other challenging skills.

Setup We evaluate two popular MRC models, BiDAF (Minjoon et al., 2017) and BERT-base (Devlin et al., 2019), which are widely adopted in the research for *shortcut* phenomena (Sugawara et al., 2018; Min et al., 2019; Si et al., 2019; Sugawara et al., 2020). For each combination of model and

[1] We use Baidu Translate API (`http://api.fanyi.baidu.com`).

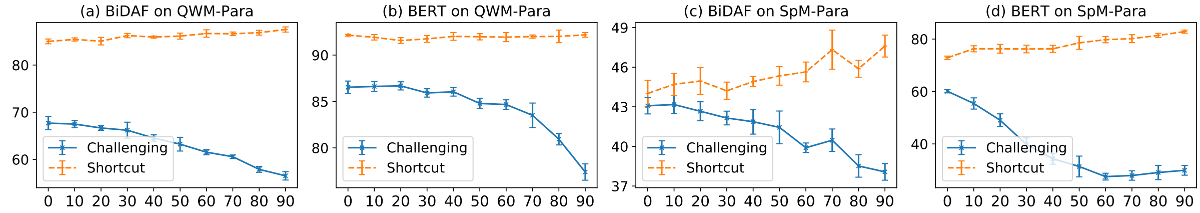

Figure 3: F1 scores on challenging and *shortcut* questions with different proportions of *shortcut* questions in training. The error bars represent the standard deviations of five runs.

dataset, we train 10 versions of the model, adjusting the proportion of *shortcut* questions in the training set from 0% to 90%, and report performance on pure challenging and pure *shortcut* test sets. We report the mean and standard deviation in five runs to alleviate the impact of randomness. Detailed settings are elaborated in Appendix A.

Results and Analysis Figure 3 shows the performance of BiDAF and BERT on *QWM-Para* and *SpM-Para* when trained with various proportions of *shortcut* questions. For both models, the F1 scores on challenging versions of both test sets drop substantially with the increase in *shortcut* questions for training (Figure 3 (a) $\sim$ (d)). This result indicates that higher proportions of *shortcut* questions in training limit the model's ability to solve challenging questions.

Take BiDAF on *QWM-Para* as an example (Figure 3 (a)). The F1 score on the test set of challenging questions is 69% after training BiDAF with a dataset entirely composed of challenging questions, showing that even a simple model is able to learn the paraphrasing skill from *shortcut*-free training data. As the proportion of *shortcut* training questions increases, the model tends to learn *shortcut* tricks and performs worse on the challenging testing data. The F1 score on challenging questions drops to 55% when 90% of the training data are *shortcut* questions. This drop shows that training data with a high proportion of *shortcuts* actually hinders the model from capturing paraphrasing skills to solve challenging questions. In contrast, the performance on *shortcut* questions are relative steady to the changes of *shortcut* proportions during training. When trained with sufficient challenging questions, models not only perform well on comprehension challenges, but also correctly answer the *shortcut* questions where only partial evidence is required.

In Figure 3, we can observe similar trends in model performance on *SpM-Para*. The performance on challenging questions also drops with higher proportions of *shortcut* training questions. Compared with BiDAF, although the overall scores of BERT are better, BERT also performs poorly on questions that require to perform paraphrasing when trained with more *shortcut* questions, as shown in Figure 3 (b) & (d).

Case Study When answering the example question **Q.4** from *QWM-Para*, BiDAF trained with pure challenging questions tends to detect the correlation between *graduate* and *received his master's degree*, and locates the correct answer *1506* when there are two spans matching the question word *when*. However, when there are more than 70% *shortcut questions* in training, BiDAF only captures the type constraint from the question word *when*, and fails to identify the paraphrasing phenomenon to answer the challenging version.

Q.4: When did Luther graduate?
P-challenging: In 1501, at the age of 19, he entered the University of Erfurt. ... He received his master's degree in *[1506]*$_{Ans}$
P-shortcut: He received his master's degree in *[1506]*$_{Ans}$

4 Whether *Question Word Matching* is Easier to Learn than *Paraphrasing*?

It is still confusing that, with the coexistence of both *shortcut* and challenging questions for training, even in a 50%-50% distribution, both BERT and BiDAF still learn *shortcut* tricks better, thus, achieve much higher performance on *shortcut questions* comparing to the challenging ones. We think one possible reason is that ***MRC models may learn the shortcut tricks, like QWM, with less computational resources than the comprehension challenges, like identifying paraphrasing***. In this case, MRC models could better learn the *shortcut* tricks with equal or even lower proportions of *shortcut* questions during training.

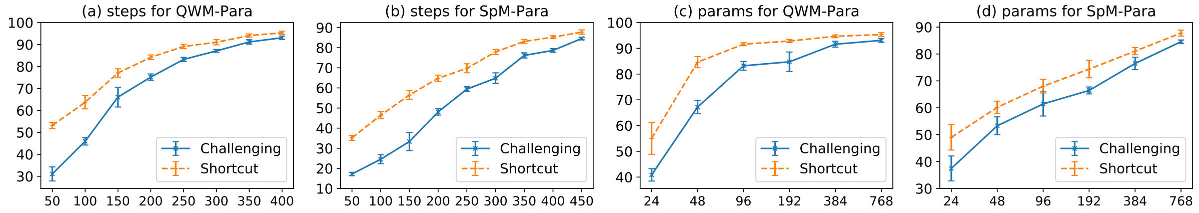

Figure 4: F1 scores on training sets when BERT learns challenging and *shortcut* questions with different optimizing steps ((a) & (b)) and parameter size (represented by the unmasked hidden size in the last hidden layer of BERT, (c) & (d)). The error bars represent the standard deviations of five runs.

To validate this hypothesis, we propose two simple but effective methods to measure the difference in required computational resources. Specifically, we can train models with either pure *shortcut* questions or challenging ones, and compare the learning speed and required parameter sizes when achieving certain performance levels on the training sets.

For learning speed, we train MRC models with different steps and observe how the performance changes. Intuitively, models should converge faster on easier training data.

For parameter sizes, our intuition is that the models should learn the easier questions with fewer parameters. However, the high computational consumption prevents us from pre-training the models like BERT with different parameter sizes. To simulate BERT with fewer parameters, we mask some hidden units in the last hidden layer of BERT and use the number of unmasked units to reflect the parameter size. The information in these masked units could not be conveyed to the span boundary prediction module. Thus, only partial parameters could be used to make the final predictions.

Setup We use BERT as the basic learner and train on the training sets of *QWM-Para* and *SpM-Para*. We report model performance on the training data with various learning settings. We use all the parameters when adjusting learning steps. When tuning parameter size, we fix the learning steps to 400 and 450 for *QWM-Para* and *SpM-Para*, respectively. All other settings including batch size, optimizer, and learning rate are fixed. We report the mean and standard deviation in five runs to alleviate the impact of randomness. The implementation details are similar to §3, elaborated in Appendix A.

Results and Analysis Figure 4 compares the performance of BERT trained on the *shortcut* questions and challenging questions separately under different settings. On both *QWM-Para* and *SpM-Para*, BERT converges faster in learning *shortcut* questions than learning challenging ones (Figure 4 (a) & (b)). When fixing the training steps, BERT could learn to answer the *shortcut* questions with fewer parameters (Figure 4 (c) & (d)). These results show that *shortcut* questions may be easier for models to learn than the ones requiring complex reasoning skills.

Take *QWM-Para* as an example. As can be seen from Figure 4 (a), BERT trained on the *shortcut* questions achieves a 90% F1-score on the training set after 250 steps. When trained on the challenging version, this score will not reach 90% until 350 steps. This result indicates that models could learn to answer the *shortcut* questions with the *QWM* trick faster than the paraphrasing skill.

When we train BERT on *QWM-Para* with different numbers of output units masked (Figure 4 (c)), BERT could reach the F1-score of 91% on *shortcut* data with no fewer than 96 unmasked hidden units. However, when trained on the challenging questions, BERT has to use 384 hidden units to reach the 91% F1 score, which indicates that the questions with the *paraphrasing* challenge may require more parameters to learn.

We observe similar trends on *SpM-Para* (Figure 4 (b) & (d)). BERT requires more parameters and training steps to learn the challenging version questions in *SpM-Para* than the *shortcut* version. To some extent, these results confirm our hypothesis that learning to answer questions with *shortcut* tricks like *SpM* or *QWM* requires smaller amounts of computational resources than the questions requiring challenging skills like paraphrasing.

Case Study For the example question **Q.5**, BERT trained on *shortcut* questions could correctly answer its *shortcut* version and find the only location name, *Palácio da Alvorada*, with only 48 unmasked hidden units. However, when trained with the challenging data only, the model predicts the other location name, *the Monumental Axis* as the answer with such parameter size. BERT could not recognize the paraphrasing relationship between *the place the president live* and *presidential residence* and choose the correct answer, *Palácio da Alvorada*, from the distracting location name, until using all 748 parameters.

Q.5: Where does the president of Brazil live, in Portuguese?
P-challenging: ... on a triangle of land jutting into the lake, is the Palace of the Dawn (*[Palácio da Alvorada]*$_{Ans}$; the presidential residence). Between the federal and civic buildings on the Monumental Axis is the city 's cathedral...
P-shortcut: ... on a triangle of land jutting into the lake, is the Palace of the Dawn (*[Palácio da Alvorada]*$_{Ans}$; the presidential residence)

5 How do Models Learn *Shortcuts*?

In previous section, we show that *shortcut* questions are *easier* to learn compared to the questions that require the complex *paraphrasing* skill. Then, it is interesting that, trained with a mixture of both versions of questions, how such discrepancy affects or even drives the learning procedure, e.g., how the increasing of challenging training questions alleviate the over reliance on *shortcut* tricks.

We guess one of the possible reasons is how most existing MRC models are optimized. We hypothesize that with a larger proportion of *shortcut* questions for training, ***the models have learned the shortcut tricks at the early stage, which may affect the models' further exploration for challenging skills.*** In other words, in the early stage of training, models tend to find the easiest way to fit the training data with gradient descent. Since the *shortcut* tricks require less computational resources to learn, fitting these tricks may be a priority. Afterwards, since the *shortcut* trick can be used to answer most of the training questions correctly, the limited unsolved questions remained may not motivate the models to explore sophisticated solutions that require challenging skills.

To validate this idea, we investigate how the models converge during training with different *shortcut* proportions in the training data. Notice that if a model can only answer the *shortcut* version of a question correctly, it is highly likely that the model only adopts the *shortcut* trick for this question instead of performing sophisticated reasoning skills. Thus, we think the performance gap on two versions of test data may indicate to what extent the model relies on the *shortcut* tricks, e.g., the smaller the performance gap is, the stronger complex reasoning skills the model have learned.

Setups We explore how BERT and BiDAF converge with 10% and 90% *shortcut* training questions on *QWM-Para* and *SpM-Para*. We report the F1 scores on the challenging and *shortcut* test questions, respectively, and together with their performance gaps. We compare the model performance at different learning steps to investigate when and how well the models learn the *shortcut* tricks and the challenging comprehension skills. The implementation details are the same as §3, elaborated in Appendix A.

Results and Analysis Figure 5 illustrates how the MRC models converge during training under different settings. The gap line (green with "$\star$") shows the gap between models' performance on *shortcut* questions and that on challenging ones.

For all settings, in the first few epochs, the model performance on *shortcut* questions increases rapidly, much faster than that on challenging questions, causing a steep rise of the performance gap. This result indicates that models may learn the *shortcut* tricks at the early stage of training, thus quickly and correctly answering more *shortcut* questions. And then, for the following epochs after reaching the peaks, the gap lines slightly go down (Figure 5 (a), (c), (e), and (f)) or maintain almost unchanged (Figure 5 (b), (d), (g), and (h)), which also indicates the models may learn the challenging skills later than *shortcut* tricks. One possible reason is the gradient based optimizer drives the model to optimize the global target greedily via the easiest direction. Thus, trained with a mixture of *shortcut* and challenging questions, models choose to learn the *shortcut* tricks, which require less computational resources to learn, earlier than the sophisticated paraphrasing skills.

Comparing models with different proportions of *shortcut* training questions, we find that, with 90% *shortcut* training questions (Figure 5 (b), (d), (f) and (h)), the performance gap remains at a high level in the later training stage, where the performance on the challenging test questions is relatively lower. These results provide evidence that, for most cases, after fitting on the *shortcut* questions, models seem to fail to explore the sophisticated reasoning skills.

When there are only 10% of *shortcut* training data (Figure 5 (a), (c), (e), and (g)), we can ob-

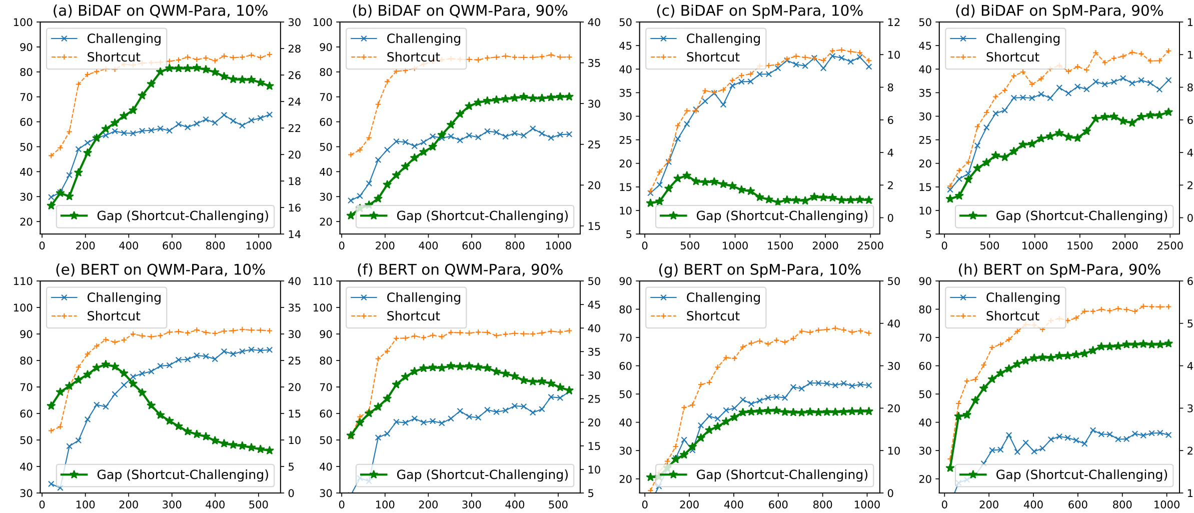

Figure 5: F1 scores on challenging and *shortcut* questions with different training steps under different settings. 10% and 90% are the proportions of *shortcut* questions in the training datasets. *Gaps* (green lines with "⋆" dots) represent the performance gap between *shortcut* questions and challenging ones, which is smoothed by averaging over fixed-size windows to mitigate periodic fluctuations.

serve that after a few hundreds of steps, the gap lines stop increasing and even slightly go down. This phenomenon shows that higher proportions of challenging questions in the training set could encourage the models to explore the sophisticated reasoning skills, but in a later stage of training.

Take BiDAF trained on *QWM-Para* as an example (Figure 5 (a) & (b)). The F1 scores on *shortcut* test questions increase quickly in the first 300 steps, while the performance gap also widens rapidly, indicating a possible fast fitting on the *shortcut* tricks. In Figure 5 (b), with 90% *shortcut* training questions, the model performance on challenging questions are relatively steady during the next 800 steps, while the F1 score on *shortcut* questions maintains a high level of about 85%. This result shows that after fitting on the *shortcut* tricks, the model trained with a higher *shortcut* proportion has almost correctly answered all the *shortcut* questions but fail to answer the challenging ones. Actually, with the gradient based optimizer, it is difficult for the model to learn the challenging questions while keeping the high performance on the *shortcut* ones, which account for 90% of the training set. We guess it is because the few unsolved challenging questions could not motivate the model to explore sophisticated reasoning skills.

On the contrary, when 90% of the training data require challenging skills, the gap line peaks at 0.27, as shown in Figure 5 (a). Afterwards, the gap decreases to 0.24, with the F1 score on challenging questions increasing to more than 60%. Larger proportions of challenging questions for training prevent the models from heavily relying on the *shortcut* tricks. This phenomenon may be because, with fewer *shortcut* questions in training, the fitting of *shortcut* tricks only benefits the training objective in a small favor. The large number of challenging questions that can not be correctly answered during the early training steps now encourage models to explore more complicated reasoning skills.

Case Study When answering the example question **Q.6** from *SpM-Para*, BERT trained with 10% *shortcut* questions tends to learn the *simple matching* trick quickly and correctly answers the *shortcut* version as early as 380 steps. However, the model cannot correctly answer the challenging variant until 630 steps. This difference demonstrates that, training with both type of questions, BERT can learn the *simple matching* trick earlier than identifying the required paraphrasing between *why defections occur* and *errors caused by*.

Q.6: Why do these defections occur?
P-challenging: ... Most of these errors are caused by *[economic or financial factors]*$_{Ans}$ ...
P-shortcut: ... Most of these defections occur because of *[economic or financial factors]*$_{Ans}$. Most of these errors are caused by *[economic or financial factors]*$_{Ans}$. ...

6 Related Works

Reading documents to answer natural language questions has drawn more and more attention in recent years (Xu et al., 2016; Minjoon et al., 2017; Lai et al., 2019; Glass et al., 2020). Most previous works focus on revealing the *shortcut* phenomenon in MRC from different perspectives. They find that manually designed features (Chen et al., 2016) or simple model architectures (Weissenborn et al., 2017) could obtain competitive performance, indicating that complicated inference procedure may be dispensable. Even without reading the entire questions or documents, models can still correctly answer a large portion of the questions (Sugawara et al., 2018; Kaushik and Lipton, 2018; Min et al., 2019). Therefore, current MRC datasets may lack the benchmarking capacity on requisite skills (Sugawara et al., 2020), and models may be vulnerable to adversarial attacks (Jia and Liang, 2017; Wallace et al., 2019; Si et al., 2019). However, they do not formally discuss or analyze why models could learn *shortcuts* from the perspectives of the learning procedure.

On the way of designing better MRC datasets, Jiang and Bansal (2019) construct adversarial questions to guide model learning the multi-hop reasoning skills. Bartolo et al. (2020) propose a model-in-loop paradigm to annotate challenging questions. More recent works (Jhamtani and Clark, 2020; Ho et al., 2020) propose new datasets with evidence based metrics to evaluate whether the questions are solved via *shortcuts*. Our work aims at providing empirical evidence and analysis to the community by tracing into the learning procedure and explaining how the MRC models learn *shortcuts*, which is orthogonal to the existing works.

For a more general machine learning perspective, there are also efforts trying to explain how models learn easy and hard instances during training. Kalimeris et al. (2019) prove that models tend to learn easier decision boundaries at the beginning stage of training. Our results empirically confirms this theoretical conclusion in the task of MRC and quantitatively explain that larger proportions of *shortcut* questions in training make MRC models rely on *shortcut* tricks, rather than comprehension skills like recognizing the paraphrase relationship.

7 Conclusions

In this work, we try to answer why many MRC models learn *shortcut* tricks while ignoring the pre-designed comprehension challenges that are purposely embedded in many benchmark datasets. We argue that ***large proportions of shortcut questions in training data push MRC models to rely on shortcut tricks excessively***. To properly investigate, we first design two synthetic datasets where each instance has a *shortcut* version paired with a challenging one which requires *paraphrasing*, a complex reasoning skill, to answer, rather than performing *question word matching* or *simple matching*. With these datasets, we are able to adjust the proportion of *shortcut* questions in both training and testing, while maintaining other factors relatively steady. We propose two methods to examine the model training process regarding the *shortcut* questions, which enable us to take a closer look at the learning mechanisms of BiDAF and BERT under different training settings. We find that learning *shortcut* questions generally requires less computational resources, and MRC models usually learn the *shortcut* questions at their early stage of training. Our findings reveal that, with larger proportions of *shortcut* questions for training, MRC models will learn the *shortcut* tricks quickly while ignoring the designed comprehension challenges, since the remaining truly challenging questions, usually limited in size, may not motivate models to explore sophisticated solutions in the later training stage.

Acknowledgments

This work is supported in part by the National Hi-Tech R&D Program of China (No.2020AAA0106600), the NSFC under grant agreements (61672057, 61672058). For any correspondence, please contact Yansong Feng.

References

Max Bartolo, A. Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932, Hong Kong, China. Association for Computational Linguistics.

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.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 137–150, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. 2019. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019*.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yuxuan Lai, Yansong Feng, Xiaohan Yu, Zheng Wang, Kun Xu, and Dongyan Zhao. 2019. Lattice cnns for matching based chinese question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6634–6641.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.

Seo Minjoon, Kembhavi Aniruddha, Farhadi Ali, and Hajishirzi Hannaneh. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Lixin Su, and Xueqi Cheng. 2019. Has-qa: Hierarchical answer spans model for open-domain question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6875–6882.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does bert learn from multiple-choice reading comprehension datasets? *ArXiv*, abs/1910.12391.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Jindou Wu, Yunlun Yang, Chao Deng, Hongyi Tang, Bingning Wang, Haoze Sun, Ting Yao, and Qi Zhang. 2019. Sogou machine reading comprehension toolkit. *arXiv preprint arXiv:1903.11848*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Hybrid question answering over knowledge base and free text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2397–2407, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

A Implement Details

Synthetic Dataset Construction During the construction of synthetic datasets, we used Stanford CoreNLP (Manning et al., 2014) to identify named entities and stop-words.

We set two empirical thresholds for identifying questions can be solved by *Simple Matching* or requiring paraphrasing skills. We consider a question as solvable via the *simple matching* trick if more than 75% of non-stop words in the question exactly appear in the answer sentence. On the other hand, if the matching rate is below 25%, we think it is unsolvable via *simple matching*, calling for other skills like paraphrasing. Thus, in dataset construction, if the matching rate is above 25% after paraphrasing, we consider that the back translation fails to incorporate paraphrasing skills into the instance.

We construct the synthetic datasets from SQuAD (Rajpurkar et al., 2016). Compared with more recent MRC datasets (Yang et al., 2018; Kwiatkowski et al., 2019), most questions in SQuAD can be solved by a single sentence with *simple matching* so that we can conveniently use back translation to construct questions with paraphrasing challenges.

Paraphrasing in *SpM-para* When constructing the *SpM-Para* dataset, we only select the instances whose questions are very similar to the corresponding answer sentences (overlap $>$ 75%) to ensure that a *simple matching* step can obtain the answers. For the *shortcut*-version of an instance, we insert the paraphrased answer sentence into passage and keep both the original answer sentence and paraphrased answer sentence (see Algorithm 2). This operation aims to control the *shortcut* instances to have both *shortcut* solutions and challenging solutions. For the challenging version, we only keep the paraphrased answer sentences in the passages and discard the original answer sentence, so that such instances can only be solved by identifying the embedded paraphrasing relationship.

Hyper-Parameters for QA models We adopt BERT (BERT-based uncased) (Devlin et al., 2019) and BiDAF (Minjoon et al., 2017) models with the implementation in SogouMRC tools (Wu et al., 2019). The hyper-parameters are shown in Table 1. We used 100-d glove vectors (Pennington et al., 2014) in BiDAF. Notice that these hyper-parameters are adopted in §3, §4, and §5. Our code and datasets can be found in `https://github.com/luciusssss/why-learn-shortcut`

For the *simple matching* setting where multiple answer spans may appear in one passage, we follow (Pang et al., 2019) and aggregate the possibilities of each span before computing the likelihood losses.

Data Sampling in Difficulty Evaluation In §4, we train BERT on the training sets of *QWM-Para* and *SpM-Para* and observe how the model converges with different learning steps and parameter sizes. However, we find BERT achieves outstanding performance on both datasets with only one or two epochs. This is because the strong learning ability of BERT model and, with only one kind of answering pattern, both the pure *shortcut* and challenging training sets are relatively easy to learn.

Under this circumstance, BERT performance on most of the evaluation checkpoints after one epoch will almost reach the final performance, which make the comparison vague. If we compare the checkpoints within one epoch, considering that models have only been trained on partial training data, the evaluation results would reflect the models' generalization ability on unseen questions. This differs from our purpose of evaluation, namely comparing the fitting difficulty of different kinds of questions. Therefore, we randomly sample 1000 pair of instances for training and evaluation. With less training data, BERT will not converge in only one or two epochs, thus we could truly evaluate the learning ability.

Computation Cost We train the models on an NVIDIA 1080 Ti GPU. The number of parameters is 5M for BiDAF and 110M for BERT. The average training time on synthetic datasets for an epoch is 1 minute for BiDAF and 10 minutes for BERT.

	BERT-base Uncased	BiDAF
# Epoch	3	15
Batch size	6	30
Optimizer	AdamWeightDecay	Adam
Learning rate	3e-5	1e-3

Table 1: Hyper-Parameters for the experiments in §4, §5, and §6.

B A Variant of *QWM-Para* Dataset

When we train models with different proportions of *shortcut* questions on *QWM-Para* (Figure 3 (a) & (b), which is described in §3), we observe that even with pure challenging questions in training, BiDAF

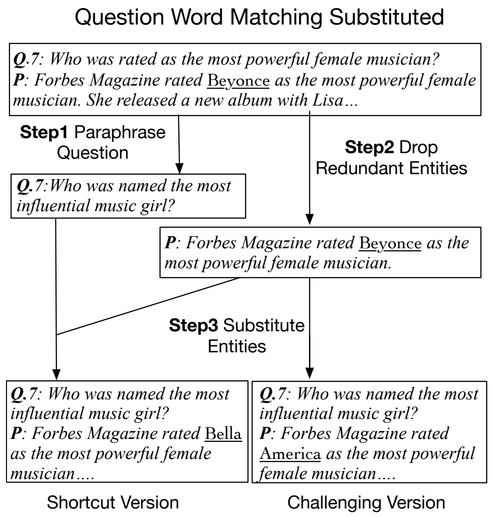
Question Word Matching Substituted
Q.7: Who was rated as the most powerful female musician?
P: Forbes Magazine rated Beyonce as the most powerful female musician. She released a new album with Lisa…
Step1 Paraphrase Question
Step2 Drop Redundant Entities
Q.7: Who was named the most influential music girl?
P: Forbes Magazine rated Beyonce as the most powerful female musician.
Step3 Substitute Entities
Q.7: Who was named the most influential music girl?
P: Forbes Magazine rated Bella as the most powerful female musician….
Q.7: Who was named the most influential music girl?
P: Forbes Magazine rated America as the most powerful female musician….
Shortcut Version
Challenging Version

Figure 6: An illustration of how the questions in the new synthetic datasets, *question word matching substituted*, are constructed from original questions.

and BERT still perform much better on *shortcut* questions than on the challenging ones. We think this is possibly because in these settings, models fail to exploit the paraphrasing skill but learn to guess one from the the entities matching the question words as the answer. Using such a guessing trick instead of the paraphrasing skill could improve the performance on the challenging questions to some extent, but it results in more gains on the *shortcut* questions. Therefore, even with 100% challenging questions in training, the gap between the performance on challenging and *shortcut* test questions is still wide.

To avoid these guess solutions, we redesign a variant of the *QWM-Para* dataset, named as ***QWM/subs-Para***.[2] We aim at investigating: 1) Whether this variant could avoid the guessing alternative and decrease the performance gaps between challenge questions and *shortcut* ones when training with relatively lower *shortcut* proportions. 2) Whether the experiments on this variant still confirms our previous findings about how *shortcut* questions in training affect model performance and learning procedure, as described in §3 and §5.

B.1 Dataset Construction

The construction process of *QWM/subs-Para* is shown in Figure 6. The first two steps, question paraphrasing and redundant entities dropping, are the same as those in the construction of *shortcut* questions in *QWM-Para* (see §2). Then, we perform entity substitution to avoid the potential guessing solutions.

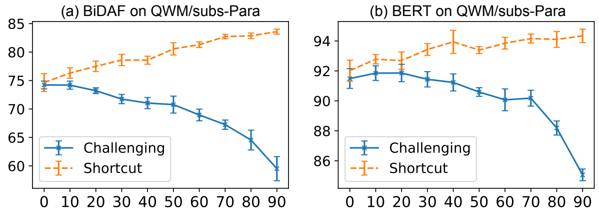

Figure 7: F1 scores on challenging and *shortcut* questions with different proportions of *shortcut* questions in training. This experiment is conducted on *QWM/subs-Para*. The error bars represent the standard deviations of five runs.

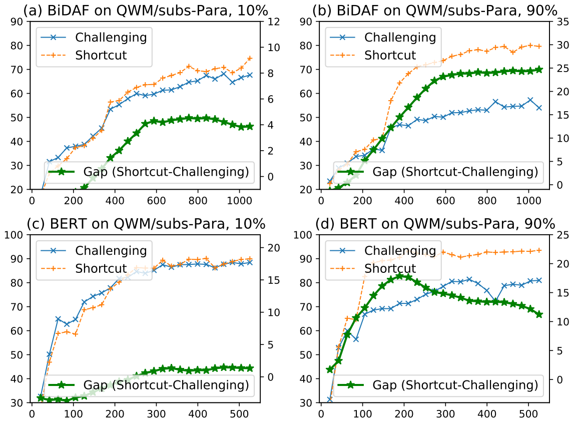

Figure 8: F1 scores on challenging and *shortcut* questions with different training steps under different settings. This experiment is conducted on *QWM/subs-Para*. 10% and 90% are the ratios of *shortcut* questions in the training datasets. *Gaps* (green lines with "⋆" dots) represents the performance gap between *shortcut* questions and challenging ones, which is smoothed by averaging over fixed-size windows to mitigate periodic fluctuations.

Particularly, for each candidate question, we substitute all the entities in the passage with random ones whose type uniformly distributes in Person/Time/Location to construct the challenging questions. With this random substitution, one can hardly guess the correct answer via the question words. As shown in **Q.7**, after substituting the answer entity *Beyonce* to *America*, one can not answer the new question by simply finding a *Person* entity according to the question word *who*. Replacing a person's name with a location may break the original semantic, but it will force the model to comprehend the context to find the answers. For the *shortcut* version, we also conduct the random entity substitution, but within the same entity types,

[2] *subs* refers to substituted, elaborated in §B.1.

e.g., from *Beyonce* to *Bella*.

This strategy could avoid the models from learning the trick that identifying replaced words as the answers.

B.2 Results and Analysis

Shown in Figure 7, we can see that, when constructing the challenging questions with entity substitution, both BiDAF and BERT model perform comparably between challenging and *shortcut* test questions with 100% challenging questions in training. These results provide evidence that, after the substitution, models could not use guessing as an alternative solutions to the paraphrasing skill.

We conduct similar experiments in §3, §5 on *QWM/subs-Para*, which is shown in Figure 7 and Figure 8, respectively. The tendency could also support our previous findings. For example, the larger *shortcut* ratio expands the performance gaps between challenging and *shortcut* questions in Figure 7.