The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Which Shortcut Solution Do Question Answering Models Prefer to Learn?

Kazutoshi Shinoda[1,2], **Saku Sugawara**[2], **Akiko Aizawa**[1,2]

[1]The University of Tokyo
[2]National Institute of Informatics
shinoda@is.s.u-tokyo.ac.jp, {saku, aizawa}@nii.ac.jp

Abstract

Question answering (QA) models for reading comprehension tend to exploit spurious correlations in training sets and thus learn shortcut solutions rather than the solutions intended by QA datasets. QA models that have learned shortcut solutions can achieve human-level performance in shortcut examples where shortcuts are valid, but these same behaviors degrade generalization potential on anti-shortcut examples where shortcuts are invalid. Various methods have been proposed to mitigate this problem, but they do not fully take the characteristics of shortcuts themselves into account. We assume that the learnability of shortcuts, i.e., how easy it is to learn a shortcut, is useful to mitigate the problem. Thus, we first examine the learnability of the representative shortcuts on extractive and multiple-choice QA datasets. Behavioral tests using biased training sets reveal that shortcuts that exploit answer positions and word-label correlations are preferentially learned for extractive and multiple-choice QA, respectively. We find that the more learnable a shortcut is, the flatter and deeper the loss landscape is around the shortcut solution in the parameter space. We also find that the availability of the preferred shortcuts tends to make the task easier to perform from an information-theoretic viewpoint. Lastly, we experimentally show that the learnability of shortcuts can be utilized to construct an effective QA training set; the more learnable a shortcut is, the smaller the proportion of anti-shortcut examples required to achieve comparable performance on shortcut and anti-shortcut examples. We claim that the learnability of shortcuts should be considered when designing mitigation methods.

Introduction

Natural language understanding (NLU) models based on deep neural networks (DNNs) have been shown to exploit spurious correlations (also called dataset bias (Torralba and Efros 2011) or annotation artifacts (Gururangan et al. 2018)) in the training set, and produce learning shortcut solutions (Geirhos et al. 2020) rather than the solutions intended by datasets. Shortcut learning by NLU models causes poor generalization to anti-shortcut examples where the spurious correlations no longer hold and the learned shortcuts fail (McCoy, Pavlick, and Linzen 2019; Gardner et al. 2020).

To date, question answering (QA) models for reading comprehension have been reported to learn several types of shortcut solutions (Jia and Liang 2017; Sugawara et al. 2018; Ko et al. 2020). Various approaches have been proposed to mitigate these problems in QA, such as data augmentation (Shinoda, Sugawara, and Aizawa 2021a) and debiasing methods (Ko et al. 2020; Wu et al. 2020). However, those methods have not fully taken the characteristics of shortcuts into account.

We assume that studying the learnability of each shortcut in QA datasets should be useful to construct training sets or design data augmentation methods for mitigating the problem. This assumption is supported by the work by Lovering et al. (2021), who show that the learnability of a shortcut and the proportion of anti-shortcut examples in a training set are the two important factors that affect the shortcut learning behavior in grammatical tasks.

To verify our assumption, we first examine the learnability of representative shortcuts in extractive and multiple-choice QA. In addition, we investigate how the learnability of a shortcut is related to the proportion of anti-shortcut examples required to mitigate the shortcut learning. Namely, we aim to answer the following research questions (RQs): *1) When every shortcut is valid for answering every question in biased training sets, which shortcut do QA models prefer to learn? 2) Why are certain shortcuts learned in preference to other shortcuts from the biased training sets? 3) How quantitatively different is the learnability for each shortcut? 4) What proportion of anti-shortcut examples in a training set is required to avoid learning a shortcut? Is it related to the learnability of shortcuts?*

We answer the first question with behavioral tests using biased training sets as illustrated in Figure 1. These experiments reveal which shortcut solution is preferred by QA models when every shortcut is applicable to the biased training sets. We show that, in extractive QA, the shortcut based on answer-position is preferred over the word matching and question-answer type matching shortcuts. In multiple-choice QA, the shortcut exploiting word-label correlations is preferred to the one using lexical overlap.

We answer the second question from the perspective of the loss landscapes qualitatively. We show that the flatness and depth of the loss surface around each shortcut solution in the parameter space can be the reason of the preference

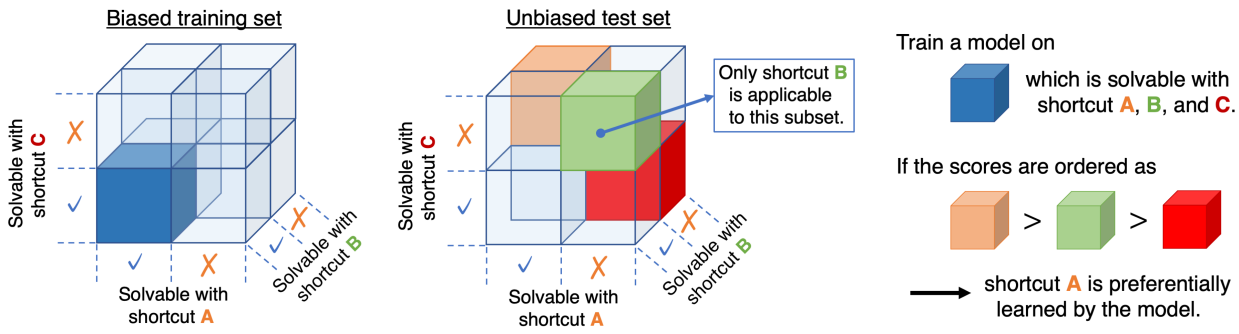

Figure 1: An illustration of the behavioral test to reveal which shortcut solution QA models prefer to learn.

qualitatively.

To quantitatively explain the preference for shortcuts, we answer the third question by quantifying the learnability of shortcuts using the minimum description lengths. We show that the availability of more preferred shortcuts in a dataset tend to make the task easier to learn.

Lastly, we answer the fourth question by simply changing the proportion of anti-shortcut examples in training sets and showing how the gap between the scores on shortcut and anti-shortcut examples changes. We show that more learnable shortcuts require less proportion of anti-shortcut examples during training to achieve the comparable performance on shortcut and anti-shortcut examples. Moreover, we find that only controlling the proportion of anti-shortcut examples is not sufficient to avoid learning less-learnable shortcuts. Our findings suggest that the learnability of shortcuts should be considered when designing mitigation methods.

Shortcut Solutions

Notation

When a training or test set $\mathcal{D}$ of a dataset is given, we define a rule-based function for each shortcut k to split $\mathcal{D}$ into shortcut examples $\mathcal{D}_{\mathrm{k}}$ that are solvable with shortcut k and anti-shortcut examples $\overline{\mathcal{D}_{\mathrm{k}}}$ that are not solvable with shortcut k. Our rule-based functions are deterministic and easy to reproduce, while partial-input baselines that are widely used for detecting shortcut examples (Gururangan et al. 2018) depend on model choice and random seeds.

Examined Shortcuts in Extractive QA

For extractive QA, we compared and analyzed the following three shortcuts, which were found in the existing literature.

Answer-Position Finding answers from the first sentence (Ko et al. 2020): When QA models are trained on examples where answers are contained in the first sentence of the context, they learn to extract answers from the first sentence. ($\mathrm{k} = \mathrm{Position}$)

Word Matching Finding the answer from the most similar sentence (Sugawara et al. 2018): When an answer is contained in a sentence that is the most similar to a question, simple word matching is sufficient to find the correct answer. We define the most similar sentence as the one that contains the longest n-gram in common with the question. ($\mathrm{k} = \mathrm{Word}$)

RACE		ReClor	
w	z^*	w	z^*
and	23.6	a	6.7
above	20.7	result	5.3
may	20.7	an	5.1
b	16.5	the	4.9
c	13.5	motive	4.5
might	10.5	not	4.3
objective	10.0	stays	4.2

Table 1: Top 7 words with the highest z-statistics computed on RACE and ReClor training sets.

Type Matching Matching question and answer types (Weissenborn, Wiese, and Seiffe 2017): When the entity type of the answer to the question can be specified, and the textual spans corresponding to the expected answer type appear only once in the context, models can answer the question correctly by simply extracting the phrase of the entity type. When the context contain two or more named entities of the same type as the answer, we classify the example into $\overline{\mathcal{D}_{\mathrm{k}}}$. To define this shortcut rigorously, we omit answers that are not named entities. We used spaCy[1] for named entity recognition. ($\mathrm{k} = \mathrm{Type}$)

Examined Shortcuts in Multiple-choice QA

For multiple-choice QA, we defined and analyzed the following two shortcuts. We adopted the two shortcuts following the work on natural language inference (NLI) (Gururangan et al. 2018; McCoy, Pavlick, and Linzen 2019) because multiple-choice QA and NLI are similar tasks as models predict whether the context+question (premise) entails the option (hypothesis).

Word-label Correlation Previous studies have shown that multiple-choice QA models can even make correct predictions with options only (Sugawara et al. 2020; Yu et al.

[1]https://spacy.io/

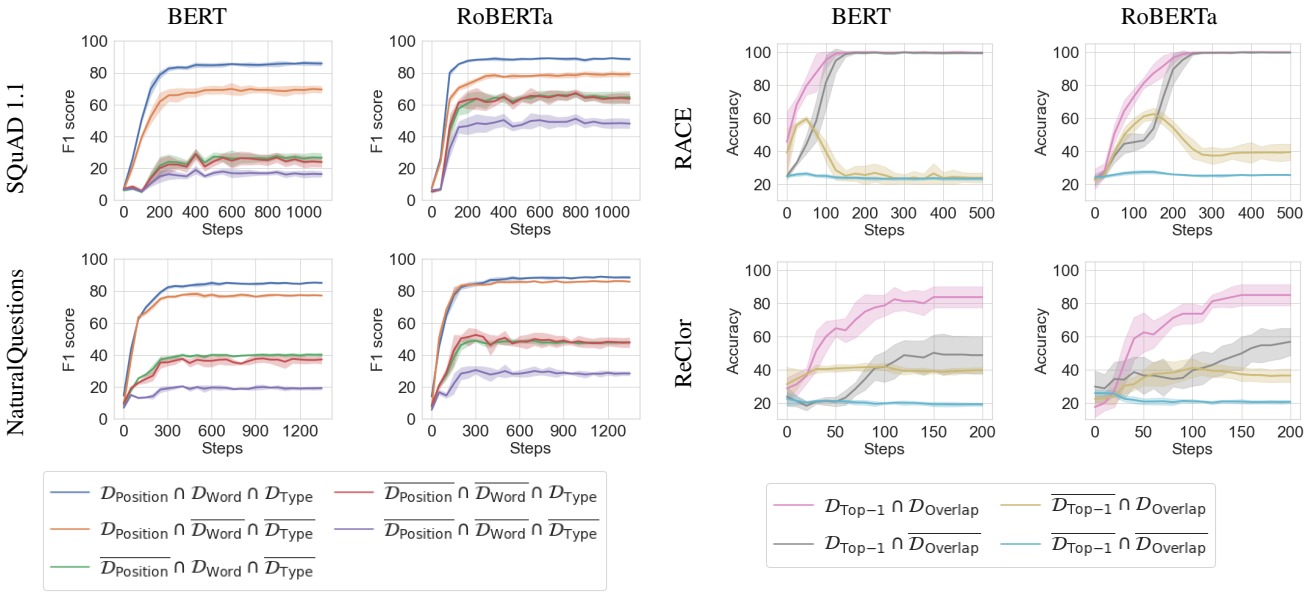

Figure 2: Left: F1 score on subsets of the SQuAD 1.1 and NaturalQuestions evaluation sets during training. Right: Accuracy on subsets of the RACE and ReClor test sets during training. The mean±standard deviations over 5 random seeds are displayed.

2020). NLI models can similarly make correct predictions with hypotheses only because certain words such as negation in hypotheses are highly correlated with labels (Gururangan et al. 2018). When considered in relation to the hypothesis-only bias in NLI, we assumed that multiple-choice QA datasets contain words in options that are highly correlated with binary labels.

Based on this assumption, we attempt to identify words in options that are highly correlated with the labels to define a realistic shortcut that exploits the word-label correlation. Gardner et al. (2021) assumed that no single feature by itself should be informative about the class label. Here, we generally follow their assumption. We use z-statistics proposed by Gardner et al. (2021) to identify word w in options with the conditional probability $p(y|w)$ that significantly deviates from the uniform distribution. Specifically, we compute the z-statistics as

$$z^* = \frac{p(y|w)}{\sqrt{p_0(1-p_0)/n}}, \quad (1)$$

where p_0 is the uniform distribution of label y, n is the frequency of word w, and $p(y|w)$ is the empirical distribution over n samples where word w is contained in the options. p_0 is $1/4$ in RACE and ReClor datasets because they have four options for each question. The top-7 words with the highest z-statistics in RACE and ReClor are shown in Table 1. We choose the top-1 word for the analysis of the word-label correlation shortcut for simplicity. ($\mathrm{k} = \text{Top-1}$)

Lexical Overlap NLI models exploit the lexical overlap between premise and hypothesis to make predictions (McCoy, Pavlick, and Linzen 2019). We assume that multiple-choice QA models can learn a similar shortcut solution using lexical overlap. We define the lexical overlap shortcut as judging an option that has the maximum lexical overlap with context+question among the options to be the answer. We define the lexical overlap as the ratio of the common unigrams contained in both sequences to the number of words in an option. ($\mathrm{k} = \text{Overlap}$)

Experiments

Experimental Setup

Datasets For extractive QA, we used SQuAD 1.1 (Rajpurkar et al. 2016) and NaturalQuestions (Kwiatkowski et al. 2019), which contain more than thousand examples in the biased training sets in Figure 1. For multiple-choice QA, we used RACE (Lai et al. 2017) and ReClor (Yu et al. 2020), where option-only models can perform better than the random baselines (Sugawara et al. 2020; Yu et al. 2020), suggesting that options in these datasets have unintended biases.

Models We used BERT-base (Devlin et al. 2019) and RoBERTa-base (Liu et al. 2019) as encoders, which are widely adopted for extractive and multiple-choice QA (Yu et al. 2020). The task-specific output layers were added on top of the encoders. For extractive QA, models output the probability distributions of the start and end positions of answer spans over context tokens. For multiple-choice QA, models predicted the probability distribution of the correct option over four options. The models were trained with cross-entropy loss minimization. Except for the training steps, we followed the hyperparameters suggested by the original papers.[2]

[2]Our codes are publicly available at https://github.com/KazutoshiShinoda/ShortcutLearnability.

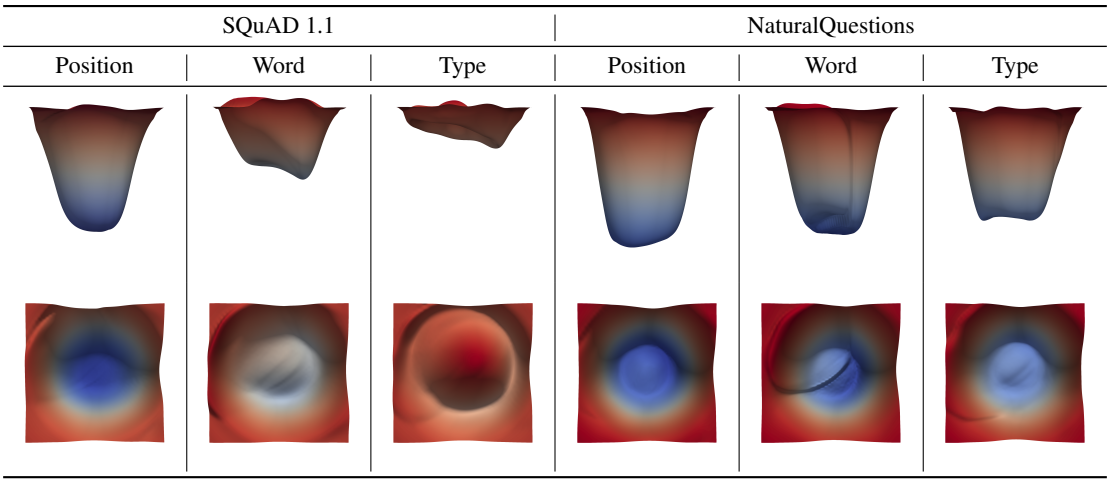

Figure 3: Visualization of loss landscapes around each shortcut in extractive QA datasets. The x and y directions are randomly selected in the parameter space. The center of the surface corresponds to the model that uses each shortcut.

Learning from Biased Training Sets

To compare the learnability of the examined shortcuts, we first answer the following research question (RQ).

RQ1 When every shortcut is valid for answering every question in biased training sets, which shortcut do QA models prefer to learn?

To answer this question, we conducted behavioral tests by training on a biased training set and testing on unbiased test sets as illustrated in Figure 1.

The important factors of shortcut learning are 1) the frequency of anti-shortcut examples in a training set and 2) how easy it is to learn the shortcut from shortcut examples (Lovering et al. 2021). In our biased training sets, all the examples are equally solvable with the examined shortcuts. Therefore, our biased training enabled the impact of pure learnability to be compared.

Setup We first trained the models on $\mathcal{D}_{\text{Position}} \cap \mathcal{D}_{\text{Word}} \cap \mathcal{D}_{\text{Type}}$ sampled from the training sets. Then, the models were evaluated on subsets such as $\mathcal{D}_{\text{Position}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$ sampled from the evaluation sets to clarify which shortcut models learn preferentially. To gain insights into the process of learning shortcut solutions, we also examined the scores during training.

Results of Extractive QA Figure 2 (left) shows the F1 score on each subset of the extractive QA datasets during training. We assume that the higher the score on a subset where only one of the three shortcuts is valid, the more preferentially the model learns the shortcut.

Regardless of the datasets and models, the F1 score on $\mathcal{D}_{\text{Position}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$ is higher than the F1 scores on $\overline{\mathcal{D}_{\text{Position}}} \cap \mathcal{D}_{\text{Word}} \cap \overline{\mathcal{D}_{\text{Type}}}$ and $\overline{\mathcal{D}_{\text{Position}}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \mathcal{D}_{\text{Type}}$ throughout the training. This observation supports that, among the three, the shortcut using answer-position is the most learnable.

Moreover, the scores on $\mathcal{D}_{\text{Position}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$ increased significantly during the first several hundred training steps. This observation is consistent with the experimental (Utama, Moosavi, and Gurevych 2020; Lai et al. 2021) and theoretical results (Hu et al. 2020); neural networks learn simpler functions at the early phase of training.

Conversely, the F1 scores on $\overline{\mathcal{D}_{\text{Position}}} \cap \mathcal{D}_{\text{Word}} \cap \overline{\mathcal{D}_{\text{Type}}}$ and $\overline{\mathcal{D}_{\text{Position}}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \mathcal{D}_{\text{Type}}$ were higher than that on $\overline{\mathcal{D}_{\text{Position}}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$. If the models exclusively learned the answer-position shortcut, the scores on these subsets would be similarly low regardless of the availability of the word and type matching shortcuts. Therefore, this observation implies that the models did not exclusively learn only one shortcut, but a mixture of multiple shortcuts.

Of the two models, RoBERTa generalized better to $\overline{\mathcal{D}_{\text{Position}}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$. RoBERTa is able to learn sophisticated solutions other than the predefined shortcuts. As BERT and RoBERTa have the same model architecture, the observations show that initialization points also affect the shortcut learning behavior.

Results of Multiple-choice QA Figure 2 (right) shows the accuracy curve on each subset of the multiple-choice QA datasets during training. At the end of the training, regardless of the models and the datasets, models learned to exploit word-label correlations more preferentially than lexical overlap because the accuracy on $\mathcal{D}_{\text{Top-1}} \cap \overline{\mathcal{D}_{\text{Overlap}}}$ is ultimately greater than that on $\overline{\mathcal{D}_{\text{Top-1}}} \cap \mathcal{D}_{\text{Overlap}}$ at the end.

Interestingly, learning the shortcut using lexical overlap conversely took precedence over the shortcut using word-label only at the early stage of the training. This may be because recognizing the dataset-specific word-label correlation requires hundreds of training steps as statistical evidence, while transformer-based language models might be originally equipped to recognize lexical overlap via self-attention (Vaswani et al. 2017).

Visualizing the Loss Landscape

RQ2 Why are certain shortcuts learned in preference to other shortcuts from the biased training sets?

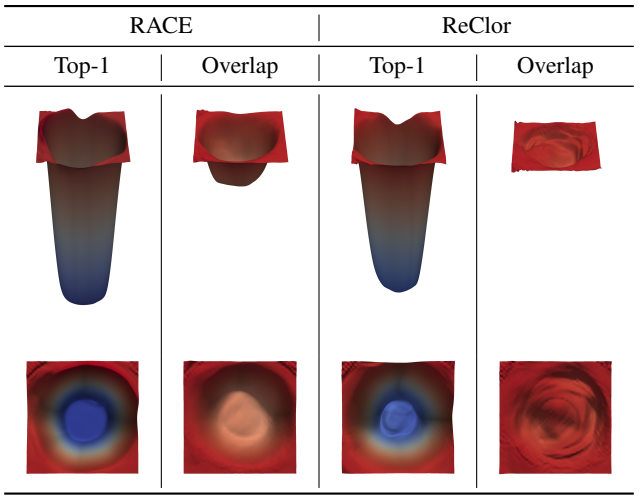

Figure 4: Visualization of loss landscapes around each shortcut in multiple-choice QA datasets.

We attempt to answer this question from the perspective of loss landscapes, as done by Scimeca et al. (2022) in image classification tasks. Specifically, we visualize the loss landscapes around shortcut solutions and compare them. The loss values were computed on subsets that are used as the biased training sets in the previous behavioral tests. By doing so, we aim to compare the flatness of loss surfaces and gain insights into the preference.

Setup To visualize the loss landscape around a shortcut solution in the parameter space, we prepared models that use that shortcut. We assume that models that are trained on subsets where only one shortcut is valid learn to use the shortcut. For example, models trained on $\mathcal{D}_{\text{Position}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$ are likely to exclusively learn the answer-position shortcut. We verified this assumption by confirming that models achieved the best performance on the same subsets of the evaluation sets as the training sets.

For visualization, we first randomly selected two directions in the parameter space. We displayed the loss values computed on $\mathcal{D}_{\text{Position}} \cap \mathcal{D}_{\text{Word}} \cap \mathcal{D}_{\text{Type}}$ and $\mathcal{D}_{\text{Top-1}} \cap \mathcal{D}_{\text{Overlap}}$ on the hyperplane spanned by the two directions following Li et al. (2018).

Results The visualization results for extractive and multiple-choice QA are displayed in Figures 3 and 4. The center of each figure represents each shortcut solution.

The results show that the QA models that learn the preferred shortcuts (Position and Top-1) tend to lie in flatter and deeper loss surfaces.[3] The orders of the flatness and depth of the loss surfaces are roughly correlated with the preferential order of learning shortcuts in the previous behavioral tests. These observations explain why models trained on $\mathcal{D}_{\text{Position}} \cap \mathcal{D}_{\text{Word}} \cap \mathcal{D}_{\text{Type}}$ and $\mathcal{D}_{\text{Top-1}} \cap \mathcal{D}_{\text{Overlap}}$ learned to use the answer-position and word-label correlation shortcuts, respectively.

[3]We follow the definition of the flatness as the size of the connected region in the parameter space where the loss remains approximately constant (Hochreiter and Schmidhuber 1997).

Shortcut	BERT	RoBERTa
SQuAD 1.1		
Position	4.65 ± 0.12	4.22 ± 0.23
Word	4.94 ± 0.24	3.73 ± 0.17
Type	5.75 ± 0.30	4.52 ± 0.06
NaturalQuestions		
Position	6.28 ± 0.15	5.37 ± 0.24
Word	12.24 ± 0.14	9.08 ± 0.20
Type	11.76 ± 0.55	8.83 ± 0.38
RACE		
Top-1	0.52 ± 0.34	0.41 ± 0.29
Overlap	4.16 ± 0.55	3.55 ± 0.10
ReClor		
Top-1	0.33 ± 0.07	0.28 ± 0.03
Overlap	0.55 ± 0.03	0.52 ± 0.02

Table 2: Minimum description lengths (kbits) on biased datasets where only one of the examined shortcut solutions is valid. The means±standard deviations over five random seeds are reported.

Rissanen Shortcut Analysis

RQ3 How quantitatively different is the learnability for each shortcut?

By answering this question, we aim to quantitatively explain the preference for shortcuts. To this end, we approximately computed the minimum description length (MDL) (Rissanen 1978) on the biased datasets where one of the predefined shortcuts is applicable, such as $\mathcal{D}_{\text{Position}} \cap \overline{\mathcal{D}_{\text{Word}}} \cap \overline{\mathcal{D}_{\text{Type}}}$, and investigated how MDL changed for each shortcut. Formally, MDL measures the number of bits needed to communicate the labels y given the inputs x in a biased subset of a dataset. We name this method Rissanen Shortcut Analysis (RSA), after the father of the MDL principle. Intuitively, RSA is simple yet effective to examine how well the availability of a shortcut in a training set makes the task easier to learn in a theoretically grounded manner.

Setup We used the online code (Rissanen 1984) to approximate MDL. In this algorithm, a training set is given to a model in a sequence of portions. At each step, a model is trained from scratch on the portions given up to that point and is used to predict the next portion. Practically, when the dataset is split into S subsets with the time steps set to $\{t_1, t_2, ..., t_S\}$[4], the MDL is estimated with the online code as follows:

$$L = \sum_{i=0}^{S-1} \sum_{n=t_i+1}^{t_{i+1}} -\log_2 p_{\theta_i}(y_n|x_n), \quad (2)$$

[4]The time steps were 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.25, 12.5, 25, 50, and 100 percent of the datasets following Voita and Titov (2020).

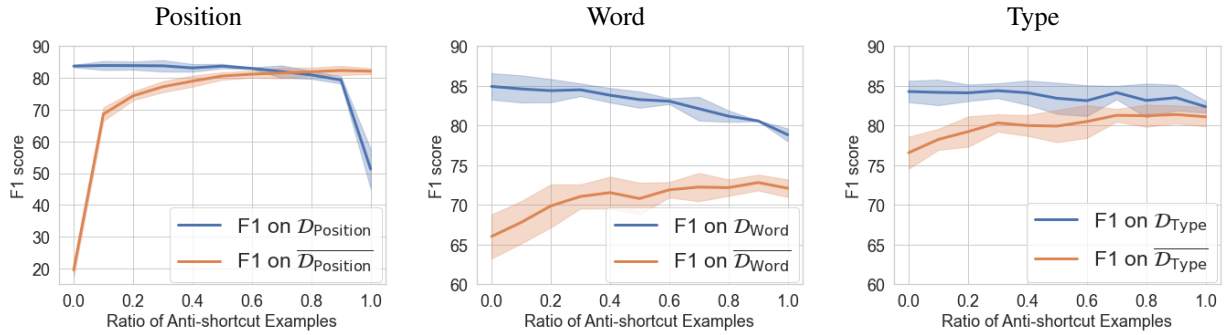

Figure 5: F1 scores on shortcut and anti-shortcut examples from SQuAD with different proportions of anti-shortcut examples in the training set, with the size set to 5k. The mean±standard deviations over 5 random seeds are displayed.

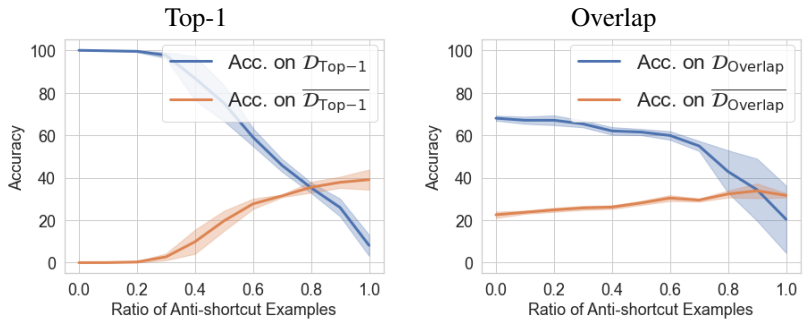

Figure 6: Accuracies on shortcut and anti-shortcut examples from RACE with different proportions of anti-shortcut examples in the training set, with the size set to 4k. The mean±standard deviations over 5 random seeds are displayed.

where θ_i is the parameter of a QA model trained on $\{(x_j, y_j)\}_{j=1}^{t_i}$ and p_{θ_0} is the uniform distribution. Intuitively, the online code is related to the area under the loss curve and measures how much effort is required for the training. See Voita and Titov (2020); Perez, Kiela, and Cho (2021) for more details about the online code. The sizes of the biased dataset were 1400, 4000, 3000, and 300 for SQuAD 1.1, NaturalQuestions, RACE, and ReClor, respectively. The size was set equally for each shortcut within a dataset.

Results The results are shown in Table 2. Note that the MDLs cannot be compared across datasets because the MDLs are dependent on the dataset size t_S as shown in Eq. 2. For SQuAD 1.1 and NaturalQuestions, the availability of the answer-position shortcut made the dataset the easiest to learn among the three shortcuts, with the exception of RoBERTa on SQuAD 1.1. The exception may be because RoBERTa can learn the word matching shortcut better than BERT as shown in Figure 2. The MDLs for the word and type matching shortcuts differed for SQuAD 1.1 and NaturalQuestions. For RACE and ReClor, the availability of the word-label correlation shortcut achieved lower MDLs than that of the lexical overlap shortcut. Except for some cases, these observations align with the results of our behavioral tests in Figure 2 and visualization in Figures 3 and 4.

In addition, RoBERTa consistently lowered the MDLs compared to BERT in all the cases. Given that RoBERTa was more robust to anti-shortcut examples than BERT in Figure 2, the MDLs may also reflect the generalization capability of models as well as the characteristics of shortcuts.

Balancing Shortcut and Anti-shortcut Examples

RQ4 What proportion of anti-shortcut examples in a training set is required to avoid learning a shortcut? Is it related to the learnability of shortcuts?

One of the simplest approaches to mitigate shortcut learning is to reduce the dataset bias by adding anti-shortcut examples to training sets manually or automatically. When a training set contains unintended biases or annotation artifacts, and the majority is solvable with shortcut solutions, models that adopt the shortcuts achieve low loss on the training set. Therefore, increasing the proportion of anti-shortcut examples is a promising approach to avoid learning shortcuts (Lovering et al. 2021).

In addition, Lovering et al. (2021) showed that the requirement of the proportion of anti-shortcut examples is related to the extractability of shortcut cues. We assume that there should be a similar relationship in QA datasets. If we know how many anti-shortcut examples are required to avoid learning shortcuts, the knowledge can be utilized to construct new QA training sets or design data augmentation approaches (Yang et al. 2017; Shinoda, Sugawara, and Aizawa 2021a) to make QA models learn more generalizable solutions.

Setup We changed the proportion of anti-shortcut examples from 0 to 1 with the sizes of the training sets fixed as 5k and 4k for extractive and multiple-choice QA, respectively. For example, for the answer-position shortcut, the proportion of $\overline{\mathcal{D}_{\mathrm{Position}}}$ was changed from 0 to 1, and the scores on $\mathcal{D}_{\mathrm{Position}}$ and $\overline{\mathcal{D}_{\mathrm{Position}}}$ were reported. We conducted the experiment for each shortcut separately on SQuAD 1.1 and RACE using BERT-base.

Results Figures 5 and 6 show the results. When the training sets consist of only shortcut examples, i.e., the x-axis value is 0, the gaps between the scores on $\mathcal{D}_{\mathrm{k}}$ and $\overline{\mathcal{D}_{\mathrm{k}}}$ are significant for all the cases. When the proportion of anti-shortcut examples is 0.7, 0.8, and 0.9, the scores on $\mathcal{D}_{\mathrm{k}}$ and $\overline{\mathcal{D}_{\mathrm{k}}}$ are equal for Position, Top-1, and Overlap, respectively. At these points, models do not use the shortcut but a solution that is equally generalizable to both the subsets. In contrast, increasing the proportion of anti-shortcut examples more than these points degraded the scores on $\mathcal{D}_{\mathrm{k}}$.

When considering the learnability of each shortcut studied in our previous experiments, it is clear that more learnable shortcuts require a smaller proportion of anti-shortcut examples to achieve comparable performance on shortcut and anti-shortcut examples. Moreover, for less-learnable shortcuts, such as Word and Type, we find that the score on $\mathcal{D}_{\mathrm{k}}$ is greater than that on $\overline{\mathcal{D}_{\mathrm{k}}}$ for almost all the points. The results suggests that controlling the proportion of anti-shortcut examples alone is insufficient to mitigate the learning of less-learnable shortcuts. For these less-learnable shortcuts, we may need to apply model-centric approaches such as Clark, Yatskar, and Zettlemoyer (2019) to further mitigate the gap.

Related Work

Shortcut learning in deep neural networks (DNNs) (Geirhos et al. 2020) has received significant interests because it degrades the generalization of DNNs, causing humans to lose trust in AI (Jacovi et al. 2021). QA models for reading comprehension are no exception. Although QA models have achieved human-level performance on some benchmarks (Rajpurkar et al. 2016), they lack robustness to challenging test sets such as adversarial attacks (Jia and Liang 2017), questions that cannot be solved with partial-input baselines (Sugawara et al. 2018), paraphrased questions (Gan and Ng 2019), answers in unseen positions (Ko et al. 2020), and natural perturbations (Gardner et al. 2020).

The causes of this problem can be grouped into two categories: dataset and model. For the data-centric cause, existing studies have found that substantial amounts of examples in QA datasets are solvable with question-answer type matching (Weissenborn, Wiese, and Seiffe 2017) and word matching (Sugawara et al. 2018) for extractive QA, and partial-input baselines (Sugawara et al. 2020; Yu et al. 2020) for multiple-choice QA. As such, various shortcut solutions in QA have been studied individually. To counter these problems, data augmentation approaches have been studied in QA. Jiang and Bansal (2019) constructed adversarial documents. Bartolo et al. (2020) proposed model-in-the-loop annotation. Shinoda, Sugawara, and Aizawa (2021b) found that diversity-oriented question-answer pair generation can improve the robustness.

For the model-centric cause, several approaches have been applied to QA. Ko et al. (2020) used ensemble-based methods to unlearn an answer-position shortcut. Wu et al. (2020) proposed concurrent modeling of multiple biases. Liu et al. (2020) used virtual adversarial training to improve the robustness to adversarial attacks. Wang et al. (2021) introduced mutual-information-based regularizers.

In contrast to the above studies, several studies have attempted to understand shortcut learning. Lai et al. (2021) found that shortcut solutions are learned at the early stage of training compared to a sophisticated solution on SQuAD. Lovering et al. (2021) showed that the more extractable a shortcut cue with a probing classifier, the more anti-shortcut examples are needed to achieve low error on anti-shortcut examples in simple grammatical tasks. Scimeca et al. (2022) compared several shortcut cues in image classification tasks.

We also attempt to understand the characteristics of shortcuts in extractive and multiple-choice QA from the perspectives of the learnability, that is, how easy it is to learn a shortcut. To the best of our knowledge, we are the first to compare the difference of the learnability for each shortcut in QA. Moreover, our study suggests that the learnability of shortcuts should be considered when designing mitigation methods. This perspective is lacking in the existing mitigation studies.

Conclusion

We deepened understanding of the shortcut solutions in extractive and multiple-choice QA by comparing the learnability of shortcuts, that is, how easy it is to learn a shortcut, in a series of experiments. We first showed that when every shortcut is applicable to a training set, extractive QA models prefer the answer-position shortcut whereas multiple-choice QA models prefer the word-label correlation shortcut among the examined shortcuts. From the perspective of the parameter space, QA models that learn the preferred shortcuts tend to lie in flatter and deeper loss surfaces, which explains the cause of the preference. To quantify the learnability of each shortcut, we estimated the MDLs on biased datasets where only one shortcut is valid. The experimental results showed that the availability of more preferred shortcuts tends to make the task easier to learn. To mitigate the shortcut learning behavior, we showed that more learnable shortcuts require less proportion of anti-shortcut examples during training. The results also suggested that controlling the proportion of anti-shortcut examples alone is insufficient to avoid learning less-learnable shortcuts such as word and type matching in extractive QA. We claim that approaches for mitigating shortcut learning should be appropriately designed according to the learnability of shortcuts.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by JSPS KAKENHI Grant Numbers 21H03502, 22J13751 and 22K17954. This work was also supported by NEDO SIP-2 "Big-data and AI-enabled Cyberspace Technologies".

References

Bartolo, M.; Roberts, A.; Welbl, J.; Riedel, S.; and Stenetorp, P. 2020. Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 8: 662–678.

Clark, C.; Yatskar, M.; and Zettlemoyer, L. 2019. Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases. 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)*, 4069–4082. Hong Kong, China: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gan, W. C.; and Ng, H. T. 2019. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6065–6075. Florence, Italy: Association for Computational Linguistics.

Gardner, M.; Artzi, Y.; Basmov, V.; Berant, J.; Bogin, B.; Chen, S.; Dasigi, P.; Dua, D.; Elazar, Y.; Gottumukkala, A.; Gupta, N.; Hajishirzi, H.; Ilharco, G.; Khashabi, D.; Lin, K.; Liu, J.; Liu, N. F.; Mulcaire, P.; Ning, Q.; Singh, S.; Smith, N. A.; Subramanian, S.; Tsarfaty, R.; Wallace, E.; Zhang, A.; and Zhou, B. 2020. Evaluating Models' Local Decision Boundaries via Contrast Sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1307–1323. Online: Association for Computational Linguistics.

Gardner, M.; Merrill, W.; Dodge, J.; Peters, M.; Ross, A.; Singh, S.; and Smith, N. A. 2021. Competency Problems: On Finding and Removing Artifacts in Language Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1801–1813. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 107–112. New Orleans, Louisiana: Association for Computational Linguistics.

Hochreiter, S.; and Schmidhuber, J. 1997. Flat Minima. *Neural Computation*, 9(1): 1–42.

Hu, W.; Xiao, L.; Adlam, B.; and Pennington, J. 2020. The Surprising Simplicity of the Early-Time Learning Dynamics of Neural Networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Jacovi, A.; Marasović, A.; Miller, T.; and Goldberg, Y. 2021. Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 624–635. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383097.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031. Copenhagen, Denmark: Association for Computational Linguistics.

Jiang, Y.; and Bansal, M. 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*, 2726–2736. Florence, Italy: Association for Computational Linguistics.

Ko, M.; Lee, J.; Kim, H.; Kim, G.; and Kang, J. 2020. Look at the First Sentence: Position Bias in Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1109–1121. Online: Association for Computational Linguistics.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; Toutanova, K.; Jones, L.; Kelcey, M.; Chang, M.-W.; Dai, A. M.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7: 452–466.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 785–794. Copenhagen, Denmark: Association for Computational Linguistics.

Lai, Y.; Zhang, C.; Feng, Y.; Huang, Q.; and Zhao, D. 2021. Why Machine Reading Comprehension Models Learn Shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 989–1002. Online: Association for Computational Linguistics.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the Loss Landscape of Neural Nets. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Liu, X.; Cheng, H.; He, P.; Chen, W.; Wang, Y.; Poon, H.; and Gao, J. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lovering, C.; Jha, R.; Linzen, T.; and Pavlick, E. 2021. Predicting Inductive Biases of Pre-Trained Models. In *International Conference on Learning Representations*.

McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3428–3448. Florence, Italy: Association for Computational Linguistics.

Perez, E.; Kiela, D.; and Cho, K. 2021. Rissanen Data Analysis: Examining Dataset Characteristics via Description Length. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8500–8513. PMLR.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. Austin, Texas: Association for Computational Linguistics.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica*, 14(5): 465–471.

Rissanen, J. 1984. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4): 629–636.

Scimeca, L.; Oh, S. J.; Chun, S.; Poli, M.; and Yun, S. 2022. Which Shortcut Cues Will DNNs Choose? A Study from the Parameter-Space Perspective. In *International Conference on Learning Representations*.

Shinoda, K.; Sugawara, S.; and Aizawa, A. 2021a. Can Question Generation Debias Question Answering Models? A Case Study on Question–Context Lexical Overlap. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, 63–72. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Shinoda, K.; Sugawara, S.; and Aizawa, A. 2021b. Improving the Robustness of QA Models to Challenge Sets with Variational Question-Answer Pair Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, 197–214. Online: Association for Computational Linguistics.

Sugawara, S.; Inui, K.; Sekine, S.; and Aizawa, A. 2018. What Makes Reading Comprehension Questions Easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4208–4219. Brussels, Belgium: Association for Computational Linguistics.

Sugawara, S.; Stenetorp, P.; Inui, K.; and Aizawa, A. 2020. Assessing the Benchmarking Capacity of Machine Reading Comprehension Datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 8918–8927.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR 2011*, 1521–1528.

Utama, P. A.; Moosavi, N. S.; and Gurevych, I. 2020. Towards Debiasing NLU Models from Unknown Biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7597–7610. Online: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Voita, E.; and Titov, I. 2020. Information-Theoretic Probing with Minimum Description Length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 183–196. Online: Association for Computational Linguistics.

Wang, B.; Wang, S.; Cheng, Y.; Gan, Z.; Jia, R.; Li, B.; and Liu, J. 2021. Info{BERT}: Improving Robustness of Language Models from An Information Theoretic Perspective. In *International Conference on Learning Representations*.

Weissenborn, D.; Wiese, G.; and Seiffe, L. 2017. Making Neural QA as Simple as Possible but not Simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 271–280. Vancouver, Canada: Association for Computational Linguistics.

Wu, M.; Moosavi, N. S.; Rücklé, A.; and Gurevych, I. 2020. Improving QA Generalization by Concurrent Modeling of Multiple Biases. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 839–853. Online: Association for Computational Linguistics.

Yang, Z.; Hu, J.; Salakhutdinov, R.; and Cohen, W. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1040–1050. Vancouver, Canada: Association for Computational Linguistics.

Yu, W.; Jiang, Z.; Dong, Y.; and Feng, J. 2020. ReClor: A Reading Comprehension Dataset Requiring Logical Reasoning. In *International Conference on Learning Representations (ICLR)*.