Debiasing Reading Comprehension Models Towards Out-of-Distribution Generalization

（分布外汎化に向けた読解モデルのバイアス除去）

by

Kazutoshi Shinoda

篠田 一聡

A Doctor Thesis

博士論文

Submitted to
the Graduate School of the University of Tokyo
on December 9, 2022
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Akiko Aizawa　相澤 彰子
Professor of Computer Science

ABSTRACT

One of the ultimate goals of natural language processing (NLP) is to make machines understand natural language as humans do. The reading comprehension task is one of the central natural language understanding (NLU) tasks because it can evaluate the broad language understanding abilities of NLU models in a unified format. Question answering (QA) models for reading comprehension based on pre-trained language models have surpassed human-level accuracy on independent and identically distributed (IID) test sets in many reading comprehension benchmarks. However, recent studies have found that QA models tend to exploit spurious correlations in a training set, which are also called shortcut solutions, to achieve high accuracy on IID test sets. When spurious correlations in training sets do not hold in out-of-distribution (OOD) test sets, such shortcut solutions do not yield correct answers. Therefore, shortcut learning prevents QA models from generalizing to OOD test sets, which could cause severe problems in real-world applications and diminish human trust in artificial intelligence. Moreover, exploiting spurious correlations to achieve high accuracy is the exact opposite of the goal of the reading comprehension task, which is to learn human-like reading comprehension skills.

In this thesis, we tackle this problem from four perspectives: data augmentation, modifying loss functions, mitigation method selection, and model architectures. In Chapter 1, we introduce the brief background of NLU tasks and shortcut learning of NLU models and clarify the focus of this thesis by showing the limitations of existing studies. In Chapter 2, we give an overview of existing studies for analyzing shortcut solutions, designing shortcut mitigation methods, and understanding the shortcut learning behavior, thereby clarifying our contributions in this thesis. In Chapter 3, we propose a diversity-oriented data augmentation method based on question-answer pair generation (QAG) for improving the QA performance on 12 OOD test sets. Improving the diversity of questions and answers in a training set is an effective approach to improve the OOD generalization. However, we find that the existing QAG methods based on generative models unintentionally amplify dataset bias in terms of question-context lexical overlap. We propose a simple data augmentation method based on synonym replacement to mitigate this problem. This method improves the scores on questions with low lexical overlap, but it has a limitation in degrading the scores on questions with high lexical overlap. In Chapter 4, we aim to refine mitigation methods based on loss function modification that has the potential to compensate for the drawbacks of data augmentation methods. First, we propose a new biased model that works well with an existing loss function to mitigate a newly discovered shortcut cue, relative answer positions, that extractive QA models can exploit. However, such a method can't be applied to unknown shortcut solutions. To cope with unknown shortcut solutions, we next focus on the insensitivity of QA models to large perturbations unlike humans. We hypothesize that QA models are not sensitive to large perturbations because they use spurious correlations that still exist in perturbed inputs. We show a negative result of our proposed loss function to make QA models sensitive to several types of perturbations to inputs, providing a direction of future research on perturbations and OOD generalization. In Chapter 5, we claim that the learnability of shortcut solutions is useful to design shortcut mitigation methods. We show the correlation between the learnability of shortcut solutions and the requirement of data balancing for unlearning shortcut solutions in extractive and multiple-choice QA. The implication of this chapter is that whether a shortcut solution can be unlearned via only data augmentation or not depends on the learnability of the shortcut solution. In Chapter 6, we focus on a relatively understudied direction of shortcut mitigation, changing model architectures. In the interactive instruction following task, where agents are required to follow language instructions and interact with objects in 3D simulated environments, we reveal that an end-to-end neural model lacks robustness to variations of objects and language instructions. We assume that the lack of robustness to variations of vision-and-language inputs is partly caused by the entirely continuous representations of end-to-end neural models. We show that utilizing discrete representations of inputs in the computation process and task-specific rules for selecting objects mitigate the lack of robustness. In Chapter 7, we discuss the limitation of our studies and provide possible directions of future work. In Chapter 8, we summarize this thesis. We expect that our contributions will advance the analysis, mitigation, and understanding of shortcut learning in NLU towards building machines that understand language like humans.

論文要旨

自然言語処理分野における大きな目的の１つは、機械に人間のように言語を理解させることである。読解タスクは自然言語理解モデルの言語理解のための広範な能力を統一的な形式で評価できるため、自然言語理解の中心的なタスクの１つとなっている。事前学習済み言語モデルに基づく読解のための質問応答モデルは、多くの読解ベンチマークの独立同分布テストセットにおいて人間の精度を超えた。しかし、近年の研究によって、質問応答モデルは独立同分布テストセットで高い精度を達成するために訓練セット内の擬似相関（ショートカット解法）を利用する傾向があることがわかった。訓練セット内の擬似相関が成り立たないような分布外テストセットではそのような解法を使っても正しい回答を導けない。よって、ショートカット学習は質問応答モデルの分布外テストセットへの汎化を妨げるため、実世界応用で深刻な問題を引き起こす可能性があり、人間の人工知能への信頼を損なうことにつながりうる。さらに、擬似相関を利用して高い精度を達成することは、人間のような読解スキルを学習するという読解タスクの本来の目的に反する。

本論文では、データ拡張、損失関数の変更、緩和手法の選択、そしてモデル機構の４つの観点からこの問題の解決に取り組む。１章では、自然言語理解タスクと自然言語理解モデルのショートカット学習の背景を紹介したのちに、既存研究の限界を示して本論文の焦点を明らかにする。２章では、ショートカット解法の分析、ショートカット学習の緩和手法、ショートカット学習の機序の理解についての既存研究の全体像を説明し、本論文の貢献を明確にする。３章では、12 の分布外テストセットでの精度を向上するために質問回答ペア生成に基づく多様性志向のデータ拡張手法を提案する。訓練セットの質問と回答の多様性を向上することは分布外汎化性能を向上するために有効であることを示す。しかし、既存の生成モデルに基づく質問回答ペア生成手法は質問文章間の語彙の重複に関するデータセットバイアスを意図せず増幅することを示した。この問題を改善するために、同義語置換に基づく単純なデータ拡張手法を提案する。この手法は語彙の重複が少ない質問での精度を向上するが、語彙の重複が多い質問での精度を悪化させる点で課題がある。４章では、データ拡張手法の欠点を補える可能性がある、損失関数の変更に基づく緩和手法の改善を目指す。まず、新たに発見した回答の相対位置という抽出型質問応答モデルが利用しうるショートカットの手がかりの学習を回避するための、既存のバイアス除去のための損失関数と有効に働く新しい Biased model を提案する。しかし、このような手法は未知のショートカット解法に適用することができない。未知のショートカット解法に対応するために、次に質問応答モデルが人間とは異なり大きい摂動に鈍感であることに着目する。質問応答モデルは摂動が加えられた入力にまだ存在する擬似相関を利用しているために大きな摂動に敏感ではないと仮定する。期待に反して、質問応答モデルが人間のように入力に加えられた複数のタイプの摂動に対して敏感にするような損失関数が分布外汎化性能を向上しないことを示す。これにより摂動と分布外汎化についての今後の研究の方向性を提示する。５章では、ショートカット解法の学習可能性が緩和手法の設計に有用であることを主張する。ショートカット解法の学習可能性と、ショートカット解法の忘却に必要なデータバランスの要件には相関があることを多肢選択型と抽出型の質問応答で

示す。この章によって分かったことは、ショートカット解法の忘却がデータバランスの調整によってのみ可能かどうかはショートカット解法の学習可能性に依存するということである。6章では、モデルの機構を変更するというショートカット学習の緩和の比較的研究されていない方向性に着目する。エージェントが3Dのシミュレーション環境において言語指示に従って物体と相互作用する必要があるinteractive instruction followingタスクにおいて、end-to-endニューラルモデルが多様な物体と言語指示への頑健性を欠いていることを発見する。この多様な視覚・言語入力への頑健性の欠如は、end-to-endニューラルモデルの完全に連続的な表現に部分的に起因していると仮定する。入力の離散的な表現を計算の過程で利用し、タスク固有のルールを活用して物体選択を行うことで頑健性の欠如を緩和できることを示す。7章では、本論文の限界について議論し、今後の研究の方向性を示す。8章では、本論文を総括する。本論文の貢献によって、人間のように言語を理解する機械の構築に向けて、自然言語理解におけるショートカット学習の分析、緩和、理解の研究が発展することを期待する。

Acknowledgements

First of all, I am grateful to my adviser, Professor Akiko Aizawa, who has supported my research since I entered the graduate school. Although there was a long period of time when my research did not go well, she patiently guided me. Without her support, I could not literally finish a PhD life.

Secondly, I am grateful to Dr. Saku Sugawara, who has advised me since I was a master's student as well. His passion for research significantly inspired me to pursue the PhD course. I feel very happy to collaborate with him during the whole PhD life.

Lastly, I am grateful to my family, friends, and partner, who always support me mentally. When my research did not go well, just spending time with them helped me a lot. Moreover, they sometimes listened to my worries about research and encouraged me. I would like to support them as well when they have some difficulties in their lives.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Natural Language Understanding

For decades, building machines that understand natural language as humans do has attracted significant attention from researchers. The most representative and primitive task for this goal is the Turing Test (Turing, 1950), in which machines are required to mimic human responses in dialog with humans. To assess variable skills for language understanding comprehensively, a reading comprehension task, called the Winograd Schema Challenge (Levesque et al., 2012), is proposed as an alternative to the Turing Test. Now, question answering (QA) tasks for reading comprehension have become one of the central NLU tasks due to its comprehensiveness and applicability (Rajpurkar et al., 2016; Trischler et al., 2017; Joshi et al., 2017). Reading comprehension tasks enable simultaneous evaluation of various language comprehension skills. QA models for reading comprehension are required to read a textual context and answer questions about the context. The input in QA is a pair of question and context, and the output depends on the format of a QA task. Specifically, answers can be textual span in context, one of multiple choices, or free-form text.

Context: The university is the major seat of the Congregation of Holy Cross (albeit not its official headquarters, which are in Rome). Its main seminary, Moreau Seminary, is located on the campus across St. Joseph lake from the Main Building. Old College, the oldest building on campus and located near the shore of St. Mary lake, houses undergraduate seminarians. Retired priests and brothers reside in Fatima House (a former retreat center), Holy Cross House, as well as Columba Hall near the Grotto. The university through the Moreau Seminary has ties to theologian Frederick Buechner. While not Catholic, Buechner has praised writers from Notre Dame and Moreau Seminary created a Buechner Prize for Preaching.
Question: What is the primary seminary of the Congregation of the Holy Cross?
Answer: Moreau Seminary

Table 1.1: An example of extractive QA taken from SQuAD (Rajpurkar et al., 2016).

Recently, researchers cautioned the NLU community that NLU models that are trained on only text (surface form) can not really understand the meanings of text (Bender and Koller, 2020). Therefore, grounding language to the physical world is regarded as the next stage of NLU research (Bisk et al., 2020). For example, NLU models trained on

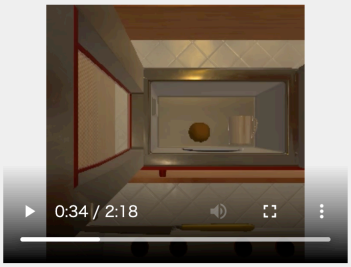
Context: The rain had continued for a week and the flood had created a big river which were running by Nancy Brown's farm. As she tried to gather her cows to a higher ground, she slipped and hit her head on a fallen tree trunk. The fall made her unconscious for a moment or two. When she came to, Lizzie, one of her oldest and favorite cows, was licking her face. At that time, the water level on the farm was still rising. Nancy gathered all her strength to get up and began walking slowly with Lizzie. The rain had become much heavier, and the water in the field was now waist high. Nancy's pace got slower and slower because she felt a great pain in her head. Finally, all she could do was to throw her arm around Lizzie's neck and try to hang on. About 20 minutes later, Lizzie managed to pull herself and Nancy out of the rising water and onto a bit of high land, which seemed like a small island in the middle of a lake of white water. Even though it was about noon, the sky was so dark and the rain and lightning was so bad that it took rescuers more than two hours to discover Nancy. A man from a helicopter lowered a rope, but Nancy couldn't catch it. A moment later, two men landed on the small island from a ladder in the helicopter. They raised her into the helicopter and took her to the school gym, where the Red Cross had set up an emergency shelter. When the flood disappeared two days later, Nancy immediately went back to the "island." Lizzie was gone. She was one of 19 cows that Nancy had lost in the flood. "I owe my life to her," said Nancy with tears.
Question: What did Nancy try to do before she fell over?
Options: 1. Measure the depth of the river / 2. Look for a fallen tree trunk / 3. Protect her cows from being drowned / 4. Run away from the flooded farm
Answer: 3. Protect her cows from being drowned

Table 1.2: An example of multiple-choice QA taken from RACE (Lai et al., 2017).

only text may not be able to perform the task by understanding the instructions shown in Figure 1.1 — what the difference between "right" and "left" is, what a potato slice looks like, how one can interact with a potato slice, etc. — because they do not have vision or a physical body.

1.2 Shortcut Learning of Natural Language Understanding Models

Not limited to QA, neural NLU models based on pre-trained language models (Devlin et al., 2019) have achieved human-level performance on many NLU benchmarks (Rajpurkar et al., 2016; Wang et al., 2018, 2019a). However, researchers have revealed that NLU models fail to generalize well to out-of-distribution (OOD) test sets whose distributions are different from that of a training set (Jia and Liang, 2017; Gururangan et al., 2018). One of the primary causes of this problem is known as shortcut learning of neural networks including NLU models (Geirhos et al., 2020). Here, the shortcut learning indicates that neural models tend to learn to exploit spurious correlations in training sets as shortcuts, rather than more complex and generalizable ones intended by the task, when the shortcuts are applicable to the majority of training examples. This behavior results



Goal Instruction	Chill a potato slice in the refrigerator.
Step-by-Step Instructions	Turn right and go to the counter. Pick up a knife on the counter. Turn around and go to the microwave. Cut the potato in the microwave. Turn around and go to the refrigerator. Put the knife in the refrigerator. Turn around and go to the microwave. Take a potato slice out of the microwave. Turn and go to the refrigerator. Put the potato slice in the refrigerator. Take the potato slice out of the refrigerator. Turn around and go to the sink. Put the potato in the sink.
Permanent Link	https://askforalfred.com/?vid=21281

Figure 1.1: An example demonstration of the ALFRED benchmark (Shridhar et al., 2020). Models are given first-person view and language instructions and predict actions to interact with objects in the 3D environments.

in failure on examples where only using shortcuts is insufficient to predict the correct labels.

QA models are no exception. For example, previous studies found that extractive QA models can exploit lexical overlap between question and context (Jia and Liang, 2017; Sugawara et al., 2018), types of question and answer (Lewis and Fan, 2019), and answer-position as shortcut cues (Ko et al., 2020) to find answers. In multi-modal learning, the same problems are also observed. For example, visual QA models that answer questions about images can correctly answer questions without looking at images (Agrawal et al., 2018).

Inherently, lack of OOD generalization is also caused by other several reasons in addition to shortcut learning. Bommasani et al. (2021) stated that spurious correlations and temporal drift are the two *persistent challenges* for neural networks when improving OOD generalization. Lack of OOD generalization caused by temporal drift means that for example models trained on existing datasets can not handle changes of world knowledge in the future. Combatting such problems requires re-training models on new data because causal relationships between inputs and labels change (Lazaridou et al., 2021). On the other hand, we claim that avoiding exploiting spurious correlations has to be mitigated first before handling temporal drift. This is because models may learn other spurious correlations when re-training models if we can't mitigate spurious correlations.

To mitigate the shortcut learning behavior, various approaches have been proposed. First, the approaches can be categorized as one of the three types, data-centric, loss-centric, and model-centric. For data-centric approaches, increasing the diversity of training sets is a promising approach. As mentioned before, models tend to learn to use simple cues when they are highly correlated with labels. Thus, for example, augmenting training sets with examples where the correlation do not hold is effective, which is known as data augmentation. For loss-centric approaches, modifying the standard loss

functions such as cross-entropy has been proposed. The standard training procedure tends to make deep learning based models learn simple decision boundaries, which is called simplicity bias (Shah et al., 2020). For model-centric approaches, model architectures are dynamically changed while the main structure of neural models are not changed in the above two categories.

Second, the mitigation approaches are also grouped by another direction, bias is known or unknown when designing approaches. If certain type of bias is known to exist in a training set a priori and the corresponding shortcut is applicable to the majority of training examples, mitigation methods specialized for the bias can be designed. For example, when using ensemble-based methods, intentionally preparing a biased model that relies on the predetermined shortcut cue encourage another model to learn more generalizable solutions via ensemble.

1.3 Limitations of Existing Studies

Given the overview of the existing studies in §1.2, we focus on the limitations from the following four perspectives in each chapter. The first one is automatic question-answer pair generation (QAG) for augmenting QA datasets. Existing studies of QAG have implicitly assumed that improving the quality of generated questions and answers is necessary for improving the generalization of QA models. However, they do not answer the following two questions. (1) Does improving the diversity of generated questions and answers lead to enhanced OOD generalization? (2) Does the existing QAG approaches debias QA datasets and models? We answer these questions in Chapter 3.

The second one is modifying loss functions to debias QA models. Existing studies have found that extractive QA models can exploit answer positions, lexical overlap, and types of questions and answers as shortcut cues. To avoid learning to rely on these shortcut cues, debiasing approaches specifically designed for one of the above shortcut cues have been proposed. However, there may be unknown shortcut cues that researchers have not found yet in existing or upcoming extractive QA datasets. To alleviate this problem, we (1) find a new shortcut cue, the relative position of answers, which QA models can exploit, and (2) pose a new possible direction for devising debiasing methods that can mitigate unknown shortcuts. We provide the details of these studies in Chapter 4.

The third one is understanding the characteristics of shortcuts in QA datasets. Most existing studies have developed debiasing methods including data augmentation or modifying loss functions for a specific type of shortcuts independently, ignoring the characteristics of the shortcuts. We assume that the learnability of shortcuts, i.e., how easy it is to learn shortcuts, can be utilized to design new debiasing methods. We verify this assumption in Chapter 5.

The forth one is the lack of studies on the relationship between the continuous nature of neural NLU models and the lack of robustness to variations of inputs. Most NLU models adopted neural architectures that use entirely continuous representations in the computation process. Whereas, such end-to-end neural models are known to be often sensitive to small changes in inputs in both computer vision and natural language processing tasks. While various loss functions or data augmentation approaches have been proposed to counter this problem, the continuous nature of computation has not received attention well regarding the problem. We hypothesize that utilizing discrete features of inputs as intermediate representations could mitigate the lack of robustness to small changes in inputs. We verify this hypothesis on a recently proposed vision-and-language task in Chapter 6.

Overall, existing studies fail to sufficiently clarify the pros and cons of different

types of debiasing methods; data-, loss-, and model-centric approaches. This thesis also aims to deepen the understanding of them by studying three types of debiasing methods comprehensively. We will discuss this point in Chapter 7 based on our contributions.

1.4 Thesis Outline

The remaining part of this thesis is structured as follows.

- In Chapter 2, we give an overview of related work. We categorize existing studies into analysis, method, and understanding, and provide the detailed explanations for each study in the three categories.
- In Chapter 3, we propose a diversity-oriented data augmentation method based on question-answer pair generation (QAG) for improving the QA performance on 12 OOD test sets. Improving the diversity of questions and answers in a training set is an effective approach to improve the OOD generalization. However, we find that the existing QAG methods based on generative models unintentionally amplify dataset bias in terms of question-context lexical overlap. We propose a simple data augmentation method based on synonym replacement to mitigate this problem. This method improves the scores on questions with low lexical overlap, but it has a limitation in degrading the scores on questions with high lexical overlap.
- In Chapter 4, we aim to refine mitigation methods based on loss function modification that has the potential to compensate for the drawbacks of data augmentation methods. First, we propose a new biased model that works well with an existing loss function to mitigate a newly discovered shortcut cue, relative answer positions, that extractive QA models can exploit. However, such a method can't be applied to unknown shortcut solutions. To cope with unknown shortcut solutions, we next focus on the insensitivity of QA models to large perturbations unlike humans. We hypothesize that QA models are not sensitive to large perturbations because they use spurious correlations that still exist in perturbed inputs. We show a negative result of our proposed loss function to make QA models sensitive to several types of perturbations to inputs, providing a direction of future research on perturbations and OOD generalization.
- In Chapter 5, we claim that the learnability of shortcut solutions is useful to design shortcut mitigation methods. We show the correlation between the learnability of shortcut solutions and the requirement of data balancing for unlearning shortcut solutions in extractive and multiple-choice QA. The implication of this chapter is that whether a shortcut solution can be unlearned via only data augmentation or not depends on the learnability of the shortcut solution.
- In Chapter 6, we focus on a relatively understudied direction of shortcut mitigation, changing model architectures. In the interactive instruction following task, where agents are required to follow language instructions and interact with objects in 3D simulated environments, we reveal that an end-to-end neural model lacks robustness to variations of objects and language instructions. We assume that the lack of robustness to variations of vision-and-language inputs is partly caused by the entirely continuous representations of end-to-end neural models. We show that utilizing discrete representations of inputs in the computation process and task-specific rules for selecting objects mitigate the lack of robustness.
- In Chapter 7, we discuss the possible directions of future work based on the existing studies and our findings in this thesis.

- In Chapter 8, we summarize this thesis.

Chapter 2

An Overview of Shortcut Learning of NLU Models

In this chapter, we give an overview of how NLU models have been diagnosed as learning shortcuts, how shortcut learning of NLU models has been mitigated, and how and why neural models learn shortcuts.

2.1 How to Diagnose NLU Models as Learning Shortcut Solutions

2.1.1 Lack of Generalization to Challenge Test Sets

If models fail to generalize to a test set where a specific shortcut solution is not available, they are probable to learn the solution. A series of studies has proposed such test sets to reveal the weakness of NLU models. The Adversarial SQuAD test set was constructed to fool question answering models that learn to find answers from sentences that are lexically similar to questions (Jia and Liang, 2017). Similarly, Mudrakarta et al. (2018) found that QA models ignore important terms in questions using integrated gradients (Sundararajan et al., 2017) and perturbed questions and successfully dropped the accuracy of QA models. Lexical overlap between input sentences can be exploited as shortcut cues by NLU models. The HANS test set was constructed to test whether NLI models rely solely on lexical overlap between premises and hypotheses (McCoy et al., 2019). The PAWS test set was proposed to see if paraphrase identification models exploit only lexical overlap between input sentences (Zhang et al., 2019). The FeverSymetric test set was designed to penalize fact verification models that rely only on cues from claims (Schuster et al., 2019). Adversarial test sets in multi-hop QA showed that QA models do not learn multi-hop reasoning but a simple word-matching shortcut (Jiang and Bansal, 2019). For the Story Cloze Test, Sharma et al. (2018) carefully collected a challenge test set so that both right and wrong endings have a similar number of tokens, similar distributions of token n-grams and char-ngrams, and the same likelihood to occur as standalone events.

2.1.2 Partial-input Baseline

If models can correctly predict the labels even from partial inputs (e.g., the first token of a question) from which intended features are omitted, models should exploit shortcut cues (annotation artifacts) in the partial inputs. Gururangan et al. (2018); Poliak et al. (2018) and Tsuchiya (2018) found that only hypothesis is enough to predict the label in existing NLI datasets. The analysis revealed that certain phrases are highly correlated with certain labels, which serve as shortcut cues. Sugawara et al. (2018) revealed that two partial input baselines, reading only the most similar sentence and using only the first k token of a question, are sufficient to correctly predict answers in most cases in QA, which can be used to split a test set into easy and hard subsets. They also showed that hard subsets often require not only word matching skills but also other reasoning skills.

In multiple-choice QA, Yu et al. (2020) found that correct options can be predicted with only looking at options. Partial-input baselines have also achieved good performance in commonsense reasoning (Branco et al., 2021), story cloze task (Schwartz et al., 2017), math word problems (Yang et al., 2022), and multi-hop QA (Min et al., 2019). The high accuracy of these partial-input baselines suggests that the examined NLU datasets contain some spurious correlation between inputs and labels. These datasets may produce full-input models that exploit such undesirable correlations as discussed in (Gururangan et al., 2018; Poliak et al., 2018). However, Srikanth and Rudinger (2022) investigated whether full-input models ignore context in NLI or not, and showed that it is not true. What full-input models learn from datasets can not be easily inferred only from partial-input baselines.

2.1.3 Generalization from Biased Training Sets

If the primary difference between training and test sets is the availability of a shortcut and models fail to generalize to the test set, models should exploit it. In this context, biased training sets indicate examples most or all of which are solvable with some shortcut solution. Biased training sets are useful not only for revealing the possible weakness of deep learning models but also for posing challenging evaluation setting to researchers who develop debiasing methods. For NLU, Ko et al. (2020) found that when training QA models on examples that are intentionally filtered so that answers exist in the first sentence lack the generalization to examples where answers exist in unseen positions. Lewis and Fan (2019) showed that QA models trained on biased examples where only matching types of questions and answers is enough to predict correct answers exhibit the degraded generalization to anti-biased examples to which the heuristic can not be applied. They used this train-test split to evaluate the proposed method. In computer vision, biased MNIST, where the colors of images intentionally correlates strongly but spuriously with the class labels, has been also used as a challenging testbed to evaluate debiasing algorithms (Arjovsky et al., 2019; Bahng et al., 2020).

2.1.4 Showing the Difference between the Behavior of NLU Models and Humans

NLU models have been shown to exhibit different reading behavior from humans in NLU tasks, revealed by carefully designed input modifications. For example, QA models trained on SQuAD are not robust to paraphrased questions (Gan and Ng, 2019), implications derived from SQuAD (Ribeiro et al., 2019), questions with low lexical overlap (Sugawara et al., 2018), and other QA datasets (Yogatama et al., 2019; Talmor and Berant, 2019; Sen and Saffari, 2020). Insensitivity of models to perturbations to inputs also has attracted attention from the community. Sugawara et al. (2020) showed that perturbing inputs to delete features indispensable for human reading surprisingly do not hurt the performance of QA models on test sets, thereby questioning the benchmarking capacity of existing QA datasets. Gardner et al. (2020) proposed to manually perturb test instances in small but meaningful ways that change the ground truths to test a model's local decision boundary. Schlegel et al. (2021) found that extractive QA models struggle to correctly process semantics altering modifications. Word order shuffling do not degrade the performance of NLU models (Sugawara et al., 2020; Hessel and Schofield, 2021; Gupta et al., 2021; Sinha et al., 2021). Similarly deleting words in inputs do not affect the accuracy of NLU models (Feng et al., 2018; Longpre et al., 2021). These observations imply that models rely on more shallow shortcut solutions instead of the intended ones.

Task	Analysis Method	Example
NLI	Anti-biased Test Set	• Lexical Overlap (McCoy et al., 2019)
	Partial-input Baseline	• Hypothesis-only baseline (Poliak et al., 2018)
	Biased Training Set	
	Others	• Word order shuffling (Gupta et al., 2021)
EX-QA	Anti-biased Test Set	• Lexical Overlap (Jia and Liang, 2017)
	Partial-input Baseline	• First k tokens & The most similar sentence (Sugawara et al., 2018)
	Biased Training Set	• Absolute Answer Position (Ko et al., 2020) • Question-answer Type Matching (Lewis and Fan, 2019) • Relative Answer Position (**Chap. 4**)
	Others	• Input Ablation (Sugawara et al., 2020) • Adversarial Attacks (Wu et al., 2021)
MC-QA	Anti-biased Test Set	
	Partial-input Baseline	• Option-only baseline (Sugawara et al., 2020; Yu et al., 2020)
	Biased Training Set	
	Others	• Magnet options (Lin et al., 2021)
V&L	Anti-biased Test Set	• VQA-CP (Agrawal et al., 2018) • VQA-CE (Dancette et al., 2021) • VQA-VS (Si et al., 2022)
	Partial-input Baseline	• Language prior (Agrawal et al., 2016)
	Biased Training Set	
	Others	

Table 2.1: Categories of analyses for finding shortcut cues.

2.2 How to Mitigate Shortcut Learning of Neural Models

We give the overview of existing shortcut mitigation methods in Table 2.2. The listed methods have been applied to NLP or computer vision tasks. We provide more detailed descriptions of each method in this section. We refer to the standard training as updating the parameters of end-to-end neural models with a standard optimizer such as Adam (Kingma and Ba, 2014) to minimize a standard loss function such as the cross entropy for a classification task.

	Bias is	
Data-centric	Known	Adversarial Examples (Jia and Liang, 2017), Reducing Lexical Overlap (**Chap. 3**)
	Unknown	Back Translation (Sennrich et al., 2016), Diversity Promoting QAG (**Chap. 3**)
Loss-centric	Known	IRM (Arjovsky et al., 2019), Group DRO (Sagawa* et al., 2020), LearnedMixin (Clark et al., 2019), Confidence Regularization (Utama et al., 2020a), Counterfactual Inference (Niu et al., 2021), Binarized Context (**Chap. 4**)
	Unknown	self-debias (Utama et al., 2020b), JTT (Liu et al., 2021), Diversity Loss (Teney et al., 2022), Entrpoy Maximization (**Chap. 4**)
Model-centric	Known	-
	Unknown	Generative Classifier (Lewis and Fan, 2019), Discrete representation (**Chap. 6**))

Table 2.2: Categories of representative approaches for improving the OOD generalization.

2.2.1 Loss-centric Approach

For loss-centric approaches, debiasing loss functions instead of standard loss functions are minimized. They are designed for learning non-shortcut solutions from training sets. The loss-centric approaches are further categorized into two: bias-aware and bias-agnostic as described below.

Bias-aware Methods

First, bias-aware methods have been designed for unlearning shortcut solutions that are found easy to be learned with standard loss functions.

Invariant risk minimization (IRM) (Arjovsky et al., 2019) relies on the fact that spurious correlations do not hold true across different distributions but causal relationships do. Namely, IRM aims to train an invariant classifier simultaneously optimal for all predefined environments, the effectiveness of which was verified on synthetic image datasets. However, Dranker et al. (2021) showed that IRM can not work well in naturalistic settings of NLI, calling for a more naturalistic setup for OOD generalization. Group DRO (Sagawa* et al., 2020) adopted a two-step approach similar to IRM: grouping training sets by bias information, and minimizing the worst-case loss among the groups. Predict then Interpolate (Bao et al., 2021) mitigated the manual annotation of groups used in Group DRO by using a set of biased models trained on different environments. Whereas, ensemble approaches has been proposed (Clark et al., 2019; Cadene et al., 2019), where some type of an ensemble of biased and main models is used for loss computation to make main models learn a generalizable solution that the biased model do not learn. Example re-weighting using the output of biased models was used as a baseline (Clark et al., 2019). Biased models used in these studies are generally partial-input models. E.g., question-only models for VQA, and hypothesis-only models for NLI. These methods have improved OOD accuracy at the cost of IID accuracy. To mitigate the trade-off between accuracies on IID and OOD test sets, Utama et al. (2020a) regularized models' confidence on biased examples where shortcut solutions are valid. Du et al. (2021)

defined shortcut degrees for each example using local mutual information between a word and a label in NLI and used the information to penalize overconfident predictions for examples with high shortcut degrees via knowledge distillation. Their method also preserved the IID accuracy. From the perspective of causal inference, Niu et al. (2021) tackled the language prior problem in VQA by excluding the direct impact of language from VQA models when making inference at the test stage. Tian et al. (2022) also formulated biases from a causal view and proposed a counterfactual reasoning framework in NLI and fact verification.

Bias-agnostic Methods

The bias-aware methods in §2.2.1 rely on prior information about biases that are found by researchers' careful analyses. However, bias information is not always accessible in real-world applications. Therefore, several studies have proposed bias-agnostic methods of loss function modification. Utama et al. (2020b) proposed self-debias, which uses a biased model with the same size as the main model. The biased model was trained without knowledge about biases to automatically identify biased examples. Next, the information was used to train the main model with the existing debiasing loss functions such as example re-weighting, ensembles (Clark et al., 2019), and confidence regularization (Utama et al., 2020a). Sanh et al. (2021) used TinyBERT (Turc et al., 2019) as a biased model that has fewer parameters than a main model. Nam et al. (2020) also employed a similar approach that amplifies the failure of a biased model. Ghaddar et al. (2021) refined self-debias by training both biased and main models in an end-to-end manner. Just Train Twice (JTT) (Liu et al., 2021) similarly adopted a two-stage training for improving the worst-group accuracy: first training a model with the standard training, and second upweighting examples that are misclassified by the first model to train the second model. Whereas, Teney et al. (2022) trained a diverse set of classifiers penalized with a diversity loss using each classifier's input gradient, thereby obtaining a generalizable model. Wang et al. (2021a) improved the robustness to adversarial attacks with an information theoretic perspective.

2.2.2 Data-centric Approach

Data augmentation Data augmentation for general NLP tasks has been studied. To name a few, Wei and Zou (2019) studied simple rules for data augmentation such as synonym replacement, random insertion, random swap, and random deletion. Back translation (Sennrich et al., 2016) is also a useful data augmentation approach using machine translation models. We refer the readers to Feng et al. (2021) for a more detailed list of data augmentation approaches for general purposes in NLP. Hereinafter, we focus on data augmentation approaches designed for question answering or debiasing.

Data augmentation with question-answer pair generation Automatic QAG from textual contexts has been proposed to mitigate the scarcity of training sets for extractive QA. Question-answer pair generation is generally done by a two-step approach: answer extraction (AE) and question generation (QG). AE is to extract possible answer spans from textual contexts, and QG is to generate questions from context-answer pairs. Yang et al. (2017a) first extracted possible answer spans using linguistic tags, and then generated questions with an RNN-based sequence-to-sequence model from contexts and extracted answers. They also used domain adaptation techniques to generate question-answer pairs from unlabeled text, thereby improving the QA performance in low-resource settings. Successive studies have improved the generation process in terms of quality and diversity. Du and Cardie (2018) framed AE as a sequence labeling task

and used BiLSTM-CRF (Huang et al., 2015). Subramanian et al. (2018) used a pointer network (Vinyals et al., 2015) to predict sequences of answer positions. Alberti et al. (2019) made use of BERT (Devlin et al., 2019) for both AE and QG. They also proposed a heuristic, called roundtrip consistency, to keep only synthetic triples where QA models can correctly predict the answers from the contexts and questions. The heuristic was shown to be effective to enhance the generalization of QA models. Liu et al. (2020a) proposed answer-clue-style-aware QG, in which answer-related chunks are used as clues and what type of questions to ask as inputs to generate diverse questions in a controlled manner. Lee et al. (2020) introduced an information-maximizing term to improve the consistency of question-answer pairs, while employing a conditional variational auto-encoder to improve the diversity. Bartolo et al. (2021) refined the QAG pipeline to make QA models robust to human adversaries.

Dataset Construction Several works have carefully designed dataset construction methods towards constructing datasets with less biases. Adversarial Filtering (Zellers et al., 2019) is a data collection paradigm where a set of classifiers iteratively select an adversarial set of machine-generated wrong answers. Bartolo et al. (2020) and Nie et al. (2020) proposed dynamic data collection, where humans are asked to fool models in the annotation loop. Ashida and Sugawara (2022) proposed to ask multiple questions with the same set of possible endings as candidate answers, thereby reducing word-label spurious correlations in multiple-choice QA.

Data Balancing Sakaguchi et al. (2020) proposed AFLite, a data filtering method to make datasets less likely to contain spurious correlations. Bras et al. (2020) provided a theoretical justification for AFLite and showed that it is useful to yield better OOD generalization. Idrissi et al. (2022) showed that subsampling or re-weighting examples are effective to improve the worst-group accuracy Kirichenko et al. (2022) showed that robustness to spurious correlations are improved by just retraining the last linear layer on a few examples where the backgrounds are not spuriously correlated with the foreground in image classification. On the other hand, Schwartz and Stanovsky (2022) claimed that the existence of spurious correlations should not be solved by dataset balancing because they often convey world knowledge facts or common sense knowledge.

Auxiliary Information Tu et al. (2020) empirically showed that multi-task learning can help improve the robustness to spurious correlations. Stacey et al. (2022) showed that using natural language explanations for regularizing attentions in NLU models improved the OOD generalization. Wen et al. (2022) used coarse estimates of outputs based on domain knowledge to encourage models to find solutions that exploit the intended cues.

2.2.3 Model-centric Approach

There exists a few studies that change model architectures to mitigate shortcut learning, i.e., model-centric approach. Unlike loss- and data-centric approaches, only bias-agnostic methods have been proposed. Lewis and Fan (2019) proposed the generative question-answering model, which learns to generate questions given answers and contexts. At inference time, the model select an answer that can serve as as input that maximize the probability of the question. In this way, they aimed to make the model look into the whole question words and avoid shallow understanding. They showed that the model is robust to spurious correlations with respect to question and answer types. Shrestha et al. (2022) proposed OccamNets, which are biased toward simpler solutions, and can

learn more complex solutions if necessary by adapting the network depth for each example. Their method is complementary to other loss- and data-centric approaches, so OccamNets can be combined with those existing methods. Model architecture modification tends to be more effective to improve OOD generalization than other types of approaches. Intuitively, this may be because changes of model architectures are more drastic than those of training sets and loss functions.

2.3 How and Why Neural Models Learn Shortcuts

Existing studies have struggled to understand how and why neural models learn shortcut solutions rather than the intended one. Neural models were shown to learn simple solutions at the early stage of the training empirically (Utama et al., 2020b; Du et al., 2021) and theoretically (Hu et al., 2020). Shah et al. (2020) showed that neural models can exclusively rely on the simplest feature among predictive features and remain invariant to the change of predictive complex features. Pezeshki et al. (2021) studied gradient starvation, a phenomenon that learning spurious feature may prevent learning generalizable features. Lovering et al. (2021) studied the relationship between the extractability of shortcut cues and the required frequency of anti-shortcut examples on synthetic and natural text datasets. Scimeca et al. (2022) investigated the preference of neural image classification models over different shortcut cues that are equally predictive of labels.

Chapter 3

Data Augmentation for Debiasing Reading Comprehension Models

3.1 Introduction

Manipulating training sets has the potential to mitigate dataset bias and shortcut learning of NLU models. This type of approaches includes increasing the sizes of training sets or filtering out biased examples manually or automatically. Such methods have the advantage of being controllable because each example can be added to and removed from training sets by analyzing the effect on dataset bias with respect to some attributes (e.g., question–context lexical overlap (§3.5)) of examples compared to other classes of mitigation approaches. In this chapter, we focus on improving the diversity of training sets and the drawbacks of data augmentation based on generative models.

Training a QA model requires a substantial number of question-answer (QA) pairs. To reduce the considerable manual cost of dataset creation, there has been a resurgence of studies on automatic QA pair generation (QAG), consisting of a pipeline of answer extraction (AE) and question generation (QG), to augment QA datasets (Yang et al., 2017a; Du and Cardie, 2018; Subramanian et al., 2018; Alberti et al., 2019).

For the downstream QA task, most existing studies have evaluated QAG methods using a test set from the same distribution as a training set (Yang et al., 2017a; Zhang and Bansal, 2019; Liu et al., 2020a). However, when a QA model is evaluated only on an in-distribution test set, it is difficult to verify that the model is not exploiting unintended biases in a dataset (Geirhos et al., 2020). Exploiting an unintended bias can degrade the robustness of a QA model, which is problematic in real-world applications. For example, recent studies have observed that a QA model does not generalize to other QA datasets (Yogatama et al., 2019; Talmor and Berant, 2019; Sen and Saffari, 2020). Other studies have found a lack of robustness to challenge sets, such as paraphrased questions (Gan and Ng, 2019), adversarial examples (Jia and Liang, 2017), questions with low lexical overlap (Sugawara et al., 2018), and questions that include noise (Ravichander et al., 2021).

While existing studies have proposed data augmentation methods targeting a particular challenge set, they are only effective at the expense of the in-distribution accuracy (Gan and Ng, 2019; Ribeiro et al., 2019; Ravichander et al., 2021). These methods assume that the target distribution is given a priori. However, identifying the type of samples that a QA model cannot handle in advance is difficult in real-world applications.

We conjecture that increasing the diversity of a training set with data augmentation, rather than augmenting QA pairs similar to the original training set, can improve the robustness of QA models. Poor diversity in QA datasets has been shown to result in the poor robustness of QA models (Lewis and Fan, 2019; Geva et al., 2019; Ko et al., 2020), supporting our hypothesis. To this end, we propose a variational QAG model

(VQAG). We introduce two independent latent random variables into our model to learn the two one-to-many relationships in AE and QG by utilizing neural variational inference (Kingma and Welling, 2013). Incorporating the randomness of these two latent variables enables our model to generate diverse answers and questions separately. We also study the effect of controlling the Kullback–Leibler (KL) term in the variational lower bound for mitigating the posterior collapse issue (Bowman et al., 2016), where the model ignores latent variables and generates outputs that are almost the same. We evaluate our approach on 12 challenge sets that are unseen during training to assess the improved robustness of the QA model.

In our experiments and existing studies, it is demonstrated that QG can improve not only the in-distribution generalization but also the out-of-distribution generalization capability of QA models (Zhang and Bansal, 2019; Lee et al., 2020; Shinoda et al., 2021a). In other areas, data augmentation techniques have been successfully used to reduce dataset biases and increase the performance of machine learning models on under-represented samples in vision (McLaughlin et al., 2015; Wong et al., 2016) and language (Zhao et al., 2018a; Zhou and Bansal, 2020). Thus, we assume that QG is useful to debias QA models and improve its robustness by augmenting QA datasets. However, it has not been fully studied whether existing QG models can contribute to debiasing QA models (i.e., improve the robustness of QA models to under-represented questions).

C					
Besides earning a reputation as a respected entertainment device, the iPod has also been accepted as a business device. Government departments, major institutions and international organisations have turned to the iPod line as a delivery mechanism for business communication and training, such as the Royal and Western Infirmaries in Glasgow, Scotland, where iPods are used to train new staff.					
Q	A	$\frac{\lvert Q \cap C \rvert}{\lvert Q \rvert}$	$C, Q \rightarrow A'$	$C, A \rightarrow Q'$	$\frac{\lvert Q' \cap C \rvert}{\lvert Q' \rvert}$
Where is Royal and Western Infirmaries located?	Glasgow, Scotland	5/8 = 0.62	Glasgow, Scotland (✓)	Where is the Royal and Western Infirmaries located? (✓)	6/9 = 0.67
Aside from recreational use, in what other arena have iPods found use?	business	4/14 = 0.29	entertainment (✗)	The iPod has been accepted as what kind of device? (✗)	7/11 = 0.64

Table 3.1: Examples of ground-truth question–answer pairs and predictions of question answering (BERT-base (Devlin et al., 2019)) and generation (SemanticQG (Zhang and Bansal, 2019)) models. C: context, Q: question, A: answer, $\frac{\lvert Q \cap C \rvert}{\lvert Q \rvert}$: question–context lexical overlap, A': predicted answer, Q': generated question. Overlapping words in the questions are underlined.

In this study, we also focus on question–context lexical overlap, inspired by the findings presented in Sugawara et al. (2018). Their work revealed that questions having low lexical overlap with context tend to require reasoning skills rather than superficial word matching, and existing QA models are not robust to these questions (Table 3.1). To see

if data augmentation with recent neural QG models can improve the robustness to those questions, we analyze the performance of BERT (Devlin et al., 2019) trained on SQuAD v1.1 (Rajpurkar et al., 2016) augmented with generated questions. Our analysis reveals that data augmentation with neural QG models frequently sacrifices the QA performance of the BERT-base model on questions with low lexical overlap, while improving that on questions with high lexical overlap. We conjecture that this is because neural QG models frequently generate questions with high lexical overlap as indicated in Table 3.1. This behavior can be interpreted as a consequence of the recent QG models pursuing higher average BLEU scores on SQuAD, which inherently contains reference questions with high lexical overlap, by copying many words from contexts to generate questions. By doing so, QG models can amplify the lexical overlap bias in the original dataset.

To address the performance degradation, we use a simple data augmentation approach using synonym replacement to generate questions with low question–context lexical overlap. We found that the proposed approach not only debiases the dataset but also improves the QA performance on questions with low lexical overlap with only 70k synthetic examples, whereas conventional neural QG approaches use more than one million synthetic examples.

In summary, our contributions in this chapter are as follows:

- We propose a variational question-answer pair generation model with explicit KL control to generate significantly diverse answers and questions (§3.2).
- We construct synthetic QA datasets using our model (§3.3.7) to boost the QA performance in an in-distribution test set, achieving comparable scores with existing QAG methods (§3.3.9).
- We discover that our method achieves meaningful improvements in unseen challenge sets, which are further boosted using a simple ensemble method (§3.3.9).
- We find that not only QA but also QG models are biased in terms of question–context lexical overlap; that is, QG models fail to generate questions with low lexical overlap (§3.5).
- We discover that data augmentation using recent neural QG models does not contribute to debias QA datasets; rather, it frequently degrades the QA performance on questions with low lexical overlap, while improving that on questions with high lexical overlap (§3.7).
- We demonstrate that the proposed simple data augmentation approach using synonym replacement (§3.6) for augmenting questions with low lexical overlap is effective to improve QA performance on questions with low lexical overlap with only 70k synthetic examples (§3.7), while preserving or slightly hurting the overall accuracy.

3.2 Method I: Variational Question-Answer Pair Generation Model

3.2.1 Problem Definition

Our problem is to generate QA pairs from textual contexts. We focus on extractive QA in which an answer is a text span in context. We use c, q, and a to represent the context, question, and answer, respectively. We assume that every QA pair is sampled independently given a context. Thus, the problem is defined as maximizing the conditional log likelihood $\log p(q, a|c)$ averaged over all samples in a dataset.

3.2.2 Variational Lower Bound with Explicit KL Control

Generating questions and answers from different latent spaces makes sense because multiple questions can be created from a context-answer pair and multiple answer spans can be extracted from a context. Thus, we introduce two independent latent random variables to assign the roles of diversifying AE and QG to z and y, respectively.

VAEs often suffer from *posterior collapse*, where the model learns to ignore latent variables and generates outputs that are almost the same (Bowman et al., 2016). Many approaches have been proposed to mitigate this issue, such as weakening the generators (Bowman et al., 2016; Yang et al., 2017b; Semeniuta et al., 2017), or modifying the objective functions (Tolstikhin et al., 2018; Zhao et al., 2017a; Higgins et al., 2017).

To mitigate this problem, we use a variant of the modified β-VAE (Higgins et al., 2017) proposed by Burgess et al. (2018), which uses two hyperparameters to control the KL terms. Our modified objective function is:

$$\begin{aligned}\mathcal{L} = \mathbb{E}_{q_\phi(z,y|q,a,c)}&[\log p_\theta(q|y,a,c) \\ &+ \log p_\theta(a|z,c)] \\ &- |D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|c)) - \mathrm{C_a}| \\ &- |D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|c)) - \mathrm{C_q}|, \end{aligned} \tag{3.1}$$

where D_{KL} is the KL divergence, θ (ϕ) is the parameters of the generative (inference) model, and $\mathrm{C_a}, \mathrm{C_q} \geq 0$. See §3.2.4 for the derivation of the objective. Tuning $\mathrm{C_a}$ and $\mathrm{C_q}$ was enough to regularize the KL terms in our case (see §3.3.6). $\mathrm{C_a}$ and $\mathrm{C_q}$ can explicitly control the KL values because the KL terms are forced to get closer to these values during training.

3.2.3 Information Theoretic Interpretation of the KL control

We mathematically show that the KL control can be interpreted as controlling the conditional mutual information $I(Z;A|C)$ and $I(Y;Q|C)$. This is the major difference between our model and Lee et al. (2020), where $I(Q;A)$ is maximized to improve consistency of QA pairs.[1]

When training our models, we miximized the variational lower bound in Eq. 3.1 is averaged over the training samples. In other words, the expectation with respect to the data distribution is maximized. In the ideal case, the approximated posterior $q_\phi(z|a,c)$ is equal to the true posterior $p_\theta(z|a,c)$. Then, the expectation of the KL terms with respect to the data distribution is equivalent to the conditional mutual information $I(Z;A|C)$.

Mathematically, when the approximated posterior q_ϕ is equal to the true posterior p_θ, the expectation of the KL terms in Eq. 3.1 with respect to the data distribution is:

$$\begin{aligned}&\mathbb{E}_{p(q,a,c)}[D_{\mathrm{KL}}(p(z|a,c)||p(z|c))] \\ &= \sum_{a,c} p(a,c) \int_z p(z|a,c) \log \frac{p(z|a,c)}{p(z|c)} dz \\ &= \sum_{a,c} \int_z p(a,c,z) \log \frac{p(a,z|c)}{p(z|c)p(a|c)} dz \\ &= I(Z;A|C).\end{aligned}$$

Thus, controlling the KL term in Eq. 3.1 is equivalent to controlling the conditional mutual information. The same is true for question $I(Y;Q|C)$.

[1] The upper cases represent the random variables.

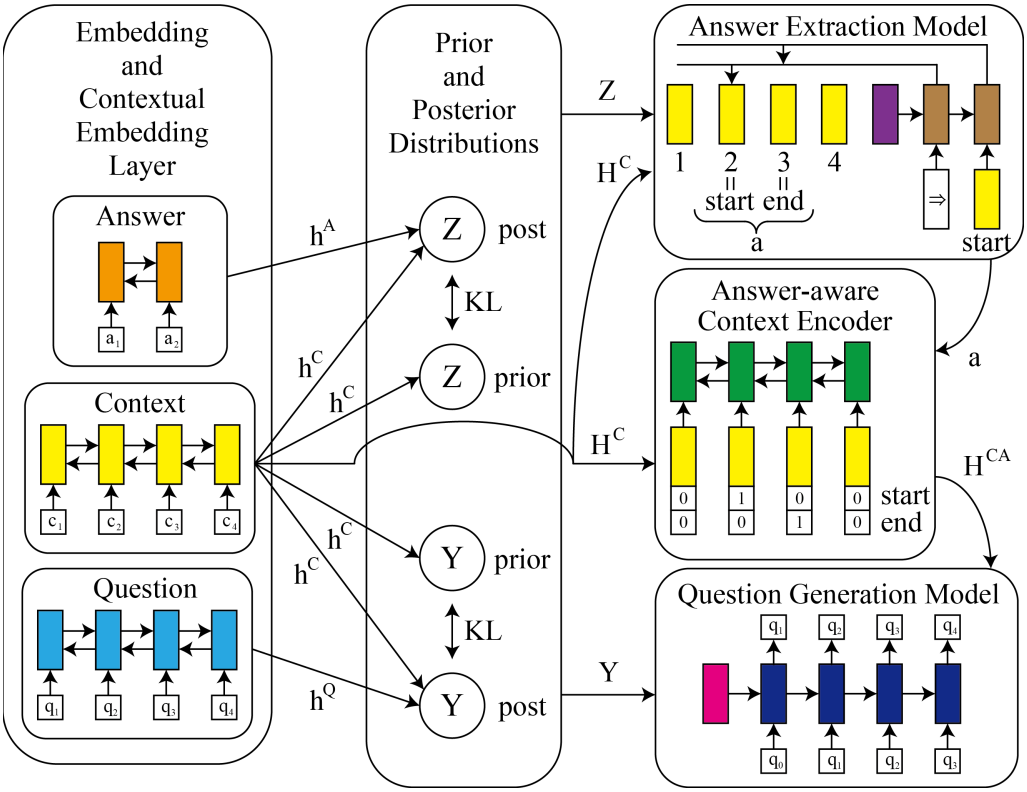


Figure 3.1: Overview of the model architecture. The latent variables z and y are sampled from the posteriors when computing the variational lower bound and from the priors during generation. See §3.2.5 for the details.

3.2.4 Derivations of the Variational Lower Bound

The variational lower bound, Eq. 3.1, without the KL control is derived as follows:

$$
\begin{aligned}
&\log p_\theta(q, a|c) \\
&= \mathbb{E}_{z,y\sim q_\phi(z,y|q,a,c)}\left[\log p_\theta(q, a|c)\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q, a|z, y, c)p_\theta(z, y|c)}{p_\theta(z, y|q, a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q, a|z, y, c)p_\theta(z, y|c)}{p_\theta(z, y|q, a, c)}\right. \\
&\quad \left. + \log \frac{q_\phi(z, y|q, a, c)}{q_\phi(z, y|q, a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log \frac{p_\theta(q|y, a, c)p_\theta(y|c)}{p_\theta(y|q, c)}\right. \\
&\quad + \log \frac{p_\theta(a|z, c)p_\theta(z|c)}{p_\theta(z|a, c)} \\
&\quad \left. + \log \frac{q_\phi(y|q, c)}{q_\phi(y|q, c)} + \log \frac{q_\phi(z|a, c)}{q_\phi(z|a, c)}\right] \\
&= \mathbb{E}_{z,y}\left[\log p_\theta(q|y, a, c) + \log p_\theta(a|z, c)\right. \\
&\quad \left. + \log \frac{p_\theta(y|c)}{q_\phi(y|q, c)} + \log \frac{q_\phi(y|q, c)}{p_\theta(y|q, c)}\right.
\end{aligned}
$$

$$
\begin{aligned}
& + \log \frac{p_\theta(z|c)}{q_\phi(z|a,c)} + \log \frac{q_\phi(z|a,c)}{p_\theta(z|a,c)} \Big] \\
= \; & \mathbb{E}_{z,y} \left[\log p_\theta(q|y,a,c) + \log p_\theta(a|z,c) \right] \\
& - D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|c)) \\
& + D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|q,c)) \\
& - D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|c)) \\
& + D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|a,c)) \\
\geq \; & \mathbb{E}_{z,y} \left[\log p_\theta(q|y,a,c) + \log p_\theta(a|z,c) \right] \\
& - D_{\mathrm{KL}}(q_\phi(y|q,c)||p_\theta(y|c)) \\
& - D_{\mathrm{KL}}(q_\phi(z|a,c)||p_\theta(z|c)).
\end{aligned}
$$

3.2.5 Model Architecture

An overview of VQAG is given in Figure 3.1. We denote c_i, q_i, and a_i as the i-th word in context, question, and answer, respectively.

Embedding and Contextual Embedding Layer First, in the embedding layer, the i-th word, w_i, of a sequence of length L is simultaneously converted into word- and character-level embedding vectors by using a CNN based on Kim (2014). Then, we concatenate the embedding vectors. After that, we pass the embedding vectors to the contextual embedding layer consisting of bidirectional LSTMs (BiLSTM). We obtain $H \in \mathbb{R}^{L \times 2d}$, which is the concatenated outputs from the LSTMs in each direction at each time step, and $h \in \mathbb{R}^{2d}$, which is the concatenated last hidden state vectors in each direction. The superscripts of the outputs H and h shown in Figure 3.1 indicate where they come from. C, Q, and A denote the context, question, and answer, respectively.

Prior and Posterior Distribution Following Zhao et al. (2017b), we hypothesized that the prior and posterior distributions of the latent variables follow multivariate Gaussian distributions with diagonal covariance. The distributions are described as follows:

$$
z|a,c \sim \mathcal{N}(\mu_{post_Z}, diag(\sigma^2_{post_Z})) \tag{3.2}
$$

$$
z|c \sim \mathcal{N}(\mu_{prior_Z}, diag(\sigma^2_{prior_Z})) \tag{3.3}
$$

$$
y|q,c \sim \mathcal{N}(\mu_{post_Y}, diag(\sigma^2_{post_Y})) \tag{3.4}
$$

$$
y|c \sim \mathcal{N}(\mu_{prior_Y}, diag(\sigma^2_{prior_Y}). \tag{3.5}
$$

The mean μ and log variance $\log \sigma^2$ of these prior and posterior distributions of z and y are computed with linear transformation from h^C, h^A, and h^Q as follows:

$$
\begin{bmatrix} \mu_{post_Z} \\ \log(\sigma^2_{post_Z}) \end{bmatrix} = W_{post_Z} \begin{bmatrix} h^C \\ h^A \end{bmatrix} + b_{post_Z} \tag{3.6}
$$

$$
\begin{bmatrix} \mu_{prior_Z} \\ \log(\sigma^2_{prior_Z}) \end{bmatrix} = W_{prior_Z} h^C + b_{prior_Z} \tag{3.7}
$$

$$
\begin{bmatrix} \mu_{post_Y} \\ \log(\sigma^2_{post_Y}) \end{bmatrix} = W_{post_Y} \begin{bmatrix} h^C \\ h^Q \end{bmatrix} + b_{post_Y} \tag{3.8}
$$

$$
\begin{bmatrix} \mu_{prior_Y} \\ \log(\sigma^2_{prior_Y}) \end{bmatrix} = W_{prior_Y} h^C + b_{prior_Y}. \tag{3.9}
$$

Then, latent variable z (and y) is obtained using the reparameterization trick (Kingma and Welling, 2013): $z = \mu + \sigma \odot \epsilon$, where $\odot$ represents the Hadamard product, and $\epsilon \sim \mathcal{N}(0, I)$. Then, z and y is passed to the AE and QG models, respectively. z and y are sampled from the posteriors during training and the priors during testing.

Answer Extraction Model We regard answer extraction as two-step sequential decoding, i.e.,

$$p(a|c) = p(c_{end}|c_{start}, c)p(c_{start}|c), \tag{3.10}$$

which predicts the start and end positions of an answer span in this order. For AE, we modify a pointer network (Vinyals et al., 2015) to take into account the initial hidden state $h_0^{AE} = W_1 z + b_1$, which in the end diversify AE by learning the mappings from z to a. The decoding process is as follows:

$$h_i^{IN} = \begin{cases} e(\Rightarrow) & \text{if } i = 1 \\ H_{t_{i-1}}^C & \text{if } i = 2 \end{cases} \tag{3.11}$$

$$h_i^{AE} = \mathrm{LSTM}(h_{i-1}^{AE}, h_i^{IN}) \tag{3.12}$$

$$u_{ij}^{AE} = (v^{AE})^T \tanh(W_2 H_j^C + W_3 h_i^{AE} + b_2) \tag{3.13}$$

$$p(c_{t_i}|c_{t_{i-1}}, c) = \mathrm{softmax}(u_i) \tag{3.14}$$

where $1 \leq i \leq 2$, $1 \leq j \leq L_C$, h_i^{AE} is the hidden state vector of the LSTM, h_i^{IN} is the i-th input, t_i denotes the start (i=1) or end (i=2) positions in c, and v, W_n and b_n are learnable parameters. We learn the embedding of the special token "⇒" as the initial input h_1^{IN}.

When we used the embedding vector e_{t_i} as h_{i+1}^{IN}, instead of $H_{t_i}^C$, following Subramanian et al. (2018), we observed that the extracted spans tended to be long and unreasonable. We assume that this is because the decoder cannot get the positional information from the input in each step.

Answer-aware Context Encoder To compute answer-aware context information for QG, we use another BiLSTM. We concatenate H^C and one hot vectors of start and end positions of answer as in Figure 3.1, which are fed to the BiLSTM. We use ground-truth answers during training and predicted answers when generating QA pairs from contexts. We obtain $H^{CA} \in \mathbb{R}^{L \times 2d}$, which is the concatenated outputs from the LSTMs in each direction. H^{CA} is used as the source for attention and copying in QG.

Question Generation Model For QG, we modify an LSTM decoder with attention and copying mechanisms to take the initial hidden state $h_0^{QG} = W_4 y + b_3$ as input to diversify QG. In detail, at each time step, the probability distribution of generating words from vocabulary using attention (Bahdanau et al., 2014) is computed as:

$$h_i^{QG} = \mathrm{LSTM}(h_{i-1}^{QG}, q_{t-1}) \tag{3.15}$$

$$u_{ij}^{att} = (v^{att})^T \tanh(W_5 h_i^{QG} + W_6 H_j^{CA} + b_4) \tag{3.16}$$

$$a_i^{att} = \mathrm{softmax}(u_i^{att}) \tag{3.17}$$

$$\hat{h}_i = \textstyle\sum_j a_{ij}^{att} H_j^{CA} \tag{3.18}$$

$$\tilde{h}_i = \tanh(W_7([\hat{h}_i; h_i^{QG}] + b_5)) \tag{3.19}$$

$$P_{vocab} = \mathrm{softmax}(W_8(\tilde{h}_i) + b_6), \tag{3.20}$$

and the probability distributions of copying (Gulcehre et al., 2016; Gu et al., 2016) from context are computed as:

$$u_{ij}^{copy} = (v^{copy})^T \tanh(W_9 h_i^{QG} + W_{10} H_j^{CA} + b_7) \tag{3.21}$$

$$a_i^{copy} = \mathrm{softmax}(u_i^{copy}) \tag{3.22}$$

In parallel, the switching probability p_s is linearly estimated from the hidden state vector. Accordingly, the probability of outputting q_i is:

$$p_g = \sigma(W_{11} h_i^{QG}) \tag{3.23}$$

$$p(q_i|q_{1:i-1}, a, c) \tag{3.24}$$

$$= p_g P_{vocab}(q_i) + (1 - p_g) \sum_{j:c_j=q_i} a_{ij}^{copy} \tag{3.25}$$

where σ is the sigmoid function.

3.3 Experiments I: Question-Answer Pair Generation

3.3.1 Dataset

We used SQuAD v1.1 (Rajpurkar et al., 2016), a large scale QA dataset consisting of documents collected from Wikipedia and 100k QA pairs created by crowdworkers, as a source dataset for QAG. Answers to questions in SQuAD can be extracted from textual contexts. Since the SQuAD test set has not been released, we use the split of the dataset, SQuAD-Du (Du et al., 2017), where the original training set is split into the training set ($\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{train}}$) and the test set ($\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$), and the original development set is used as the dev set ($\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{dev}}$). The sizes of $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{train}}$, $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{dev}}$, and $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$ are 75,722, 10,570, and 11,877, respectively.

3.3.2 Training Details

We use pretrained GloVe (Pennington et al., 2014) vectors with 300 dimensions and freeze them during training. The pretrained word embeddings were shared by the input layer of the context encoder, the input and output layers of the question decoder. The vocabulary has most frequent 45k words in our training set. The dimension of character-level embedding vectors is 32. The number of windows is 100. The dimension of hidden vectors is 300. The dimension of latent variables is 200. All LSTMs used in this paper have one layer. We used Adam (Kingma and Ba, 2014) for optimization with initial learning rate 0.001. All the parameters were initialized with Xavier Initialization (Glorot and Bengio, 2010). Models were trained for 16 epochs with a batch size of 32. We used a dropout (Srivastava et al., 2014) rate of 0.2 for all the LSTM layers and attention modules.

3.3.3 Answer Extraction

First, we conducted the AE experiment, where inputs were contexts and outputs were a set of multiple answer spans. The objective of this experiment is to measure the diversity and the extent to which our extracted answers cover the ground truths. We also study the effect of $\mathrm{C_a}$ in Eq. 3.1.

Metrics To measure the accuracy of multi-span extraction, we computed Proportional Overlap (Prop.) and Exact Match (Exact) metrics (Breck et al., 2007; Johansson and Moschitti, 2010; Du and Cardie, 2018) for each pair of extracted and ground truth answer

	Relevance				Diversity
	Precision		Recall		Dist
	Prop.	Exact	Prop.	Exact	
NER	34.44	19.61	64.60	45.39	30.0k
HarQG	45.96	33.90	41.05	28.37	-
InfoHCVAE	31.59	16.18	78.75	59.32	70.1k
VQAG					
$\mathrm{C_a} = 0$	**58.39**	**47.15**	21.82	16.38	3.1k
$\mathrm{C_a} = 5$	30.16	13.41	**83.13**	**60.88**	71.2k
$\mathrm{C_a} = 20$	21.95	5.75	72.26	42.15	**103.3k**

Table 3.2: Results of answer extraction on $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$. Prop.: Proportional Overlap, Exact: Exact Match, Dist: the number of distinct context-answer pairs. $\mathrm{C_a}$ is the hyperparameter in Eq. 3.1.

	Relevance				Diversity		
	B1-R	ME-R	RL-R	Token	D1	E4	SB4
SemQG	**62.32**	**36.77**	**62.87**	7.0M	15.8k	18.28	91.44
VQAG							
$\mathrm{C_q} = 0$	35.57	18.31	33.92	7.6M	14.4k	17.33	97.61
$\mathrm{C_q} = 5$	44.19	25.84	45.18	11.5M	19.0k	19.71	82.59
$\mathrm{C_q} = 20$	48.19	25.29	48.26	4.9M	**22.4k**	**19.72**	**44.41**

Table 3.3: Results of answer-aware question generation on $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$. 50 questions for each context-answer pair are generated and evaluated to assess their diversity. B1-R, ME-R, RL-R is the recall of BLEU-1, METEOR, and Rouge-L. D1: Dist-1, E4: Ent-4, SB4: Self-BLEU-4. $\mathrm{C_q}$ is the hyperparameter in Eq. 3.1.

spans, and then we report their precision and recall.[2] Prop. is proportional to the amount of overlap between two phrases. Our models extracted 50 answers from each context. To measure the diversity, we defined a Dist score, which is the the total number of distinct context-answer pairs.

Baselines We used three baselines: named entity recognition (NER), Harvesting QG (HarQG) (Du and Cardie, 2018), and InfoHCVAE (Lee et al., 2020). For NER, we used spaCy (Honnibal et al., 2020). For HarQG, we directly copied the scores from Du and Cardie (2018). For InfoHCVAE, we trained the model on the training set, and extracted 50 answers randomly from each context for a fair comparison.

Result Table 3.2 shows the result. While we tested various values of $\mathrm{C_a}$ ranging from 0 to 100, we only report the selected values here for brevity. When using $\mathrm{C_a}$ larger than 20, the scores did not get improved. Our model with $\mathrm{C_a} = 5$ performed the best in terms of the recall scores, while surpassing the diversity of NER. The highest Dist scores did not occur together with the highest recall scores. When $\mathrm{C_a}$ is 0, the Dist score is fairly low. This implies the posterior collapse issue, though the precision scores are the best.

[2]We exclude Binary Overlap, which assigns higher scores to systems that extract the entire input context and is not a reliable metric as Breck et al. (2007) discussed.

We assert that low precision scores do not necessarily mean poor performance in our experiment because even the original test set does not cover all the valid answer spans.

3.3.4 Answer-aware Question Generation

We also conducted answer-aware QG experiments where the contexts and ground truth answer spans were the inputs to assess diversity and relevance to the gold questions.

Metrics To evaluate the diversity of the generated questions, our models generated 50 questions from each context-answer pair. We reported the recall scores (denoted as "-R") of BLEU-1 (B1), METEOR (ME), and ROUGE-L (RL) per reference question. We do not report precision scores here because our motivation is to improve diversity. To measure diversity, we reported Dist-1 (D1), Entropy-4 (E4) (Serban et al., 2017; Zhang et al., 2018), and Self-BLEU-4 (SB4) (Zhu et al., 2018).[3]

Baselines We compared our models with SemQG (Zhang and Bansal, 2019).[4] We used diverse beam search (Li et al., 2016b), sampled the top 50 questions per answer from SemQG, and used them to calculate the metrics as the baseline for a fair comparison.

Result The results in Table 3.3 show that our model can improve diversity while degrading the recall scores compared to SemQG. Using $\mathrm{C_q}$ larger than 20 did not lead to improved diversity. More detailed exploration of $\mathrm{C_a}$ and $\mathrm{C_q}$ is provided in §3.3.5.

3.3.5 Detailed Results of Answer Extraction and Answer-aware Question Generation

Tables 3.4 and 3.5 show the detailed results of AE and QG. Various values of $\mathrm{C_a}$ and $\mathrm{C_q}$ are explored.

[3]We computed Dist-1 following the definition of Xu et al. (2018), wherein Dist-1 is the number of distinct unigrams. Dist-1 is often defined as the ratio of distinct unigrams (Li et al., 2016a) but this is not fair when the number of generated sentences differs among models, so we did not use this. SB4 was calculated per 50 questions generated from each input.

[4]We reran the ELMo+QPP&QAP model, which is available at `https://github.com/ZhangShiyue/QGforQA`.

	Relevance				Diversity
	Precision		Recall		Dist
	Prop.	Exact	Prop.	Exact	
NER	34.44	19.61	64.60	45.39	30.0k
BiLSTM-CRF	45.96	33.90	41.05	28.37	-
InfoHCVAE	31.59	16.18	78.75	59.32	70.1k
VQAG					
$C_a = 0$	**58.39**	**47.15**	21.82	16.38	3.1k
$C_a = 3$	34.09	19.22	78.94	59.09	47.5k
$C_a = 5$	30.16	13.41	**83.13**	**60.88**	71.2k
$C_a = 10$	26.17	8.83	79.70	53.02	92.3k
$C_a = 15$	22.42	6.11	76.18	44.80	99.9k
$C_a = 20$	21.95	5.75	72.26	42.15	**103.3k**
$C_a = 25$	21.60	5.37	71.55	40.48	101.6k
$C_a = 30$	23.88	6.75	74.08	44.59	99.5k
$C_a = 40$	24.58	7.90	74.86	43.33	88.1k
$C_a = 50$	25.05	7.83	76.56	44.67	88.9k
$C_a = 100$	23.32	7.48	71.74	39.70	84.6k

Table 3.4: Detailed results of AE on $\text{SQuAD}^{\text{Du}}_{\text{test}}$.

	Relevance							Diversity			
	B1	B2	B3	B4	ME	RL	Token	D1	D2	E4	SB4
SemQG	48.59	32.83	24.21	18.40	24.86	46.66	133.8k	10.2k	46.4k	15.78	-

	B1-R	B2-R	B3-R	B4-R	ME-R	RL-R	Token	D1	D2	E4	SB4
SemQG	**62.32**	**47.77**	**37.96**	**30.05**	**36.77**	**62.87**	7.0M	15.8k	218.9k	18.28	91.44
VQAG											
$C_q = 0$	35.57	18.75	10.79	6.35	18.31	33.92	7.6M	14.4k	155.3k	17.33	97.61
$C_q = 3$	44.05	26.74	16.08	9.26	24.61	44.10	9.0M	17.8k	394.2k	19.14	85.88
$C_q = 5$	44.19	27.09	16.33	9.71	25.84	45.18	11.5M	19.0k	481.1k	19.71	82.59
$C_q = 10$	44.00	27.15	16.78	10.24	25.64	44.78	10.2M	18.8k	461.5k	19.69	80.39
$C_q = 15$	45.23	27.91	16.67	10.11	26.12	45.41	11.3M	19.5k	381.5k	19.40	84.56
$C_q = 20$	48.19	32.87	22.96	14.94	25.29	48.26	4.9M	22.4k	549.2k	19.72	44.41
$C_q = 25$	47.20	31.16	21.15	13.66	25.30	45.97	6.8M	22.3k	706.9k	**20.34**	47.00
$C_q = 30$	47.96	31.69	21.26	13.83	24.95	47.07	7.3M	**22.9k**	**732.8k**	18.54	50.32
$C_q = 40$	46.31	31.29	21.52	13.94	23.73	46.46	5.4M	21.0k	487.8k	19.39	55.95
$C_q = 50$	43.92	25.95	15.54	9.61	23.61	43.18	10.8M	22.2k	527.2k	19.29	73.78
$C_q = 100$	35.22	19.88	13.25	9.20	22.27	37.55	8.2M	22.1k	508.8k	19.74	**44.22**

Table 3.5: Detailed results of answer-aware QG on $\text{SQuAD}^{\text{Du}}_{\text{test}}$. Paragraph-level contexts and answer spans are used as input. The baseline model is ELMo+QPP&QAP (Zhang and Bansal, 2019) with diverse beam search (Li et al., 2016b) with a beam size 50. Bn: BLEU-n, ME: METEOR, RL: ROUGE-L, Token: the total number of the generated words, Dn: Dist-n, E4: Ent-4 (entropy of 4-grams), SB4: Self-BLEU-4. "-R" represents recall. (e.g. B1-R is the recall of B1.) One question per answer-context pair is evaluated in the upper part, while 50 questions per answer-context pair are evaluated in the lower part to assess their diversity.

beyoncé 's vocal range spans [four octaves]. [jody rosen] highlights her tone and timbre as particularly distinctive , describing her voice as " one of the most compelling instruments in popular music " . while another critic says she is a " vocal acrobat , being able to sing long and complex melismas and vocal runs effortlessly , and in key . [her vocal abilities] mean she is identified as the centerpiece of destiny 's child . [the daily mail] calls beyoncé 's voice " [versatile] " , capable of exploring power ballads , soul , rock belting , operatic flourishes , and [hip hop] . jon pareles of the new york times commented that her voice is " velvety yet [tart], with an insistent flutter and reserves of soul belting " .	
Q: how can one find her vocal abilities in key music ?	A: she is identified as the centerpiece of destiny 's child
Q: how many octaves spans beyoncé 's vocal range ?	A: spans four
Q: how many octaves 's vocal range spans the beyoncé hop vocal range ?	A: four
Q: who commented that her voice is tart yet tart ?	A: jon pareles

Table 3.6: Heatmap of extracted answer spans and generated samples using our model. The darker the color is, the more often the word is extracted. The phrases surrounded by black boxes are the ground truth answers in SQuAD.

3.3.6 Distribution Modeling Capacity

We originally developed a QA pair modeling task to evaluate and compare QA pair generation models. We compared models based on the probability they assigned to the ground truth QA pairs. We used the negative log likelihood (NLL) of QA pairs as the metric, namely, $-\log p(q, a|c)$. Since variational models can not directly compute NLL, we estimate NLL with importance sampling. We also estimate each term in decomposed NLL, i.e.,$\mathrm{NLL_a} = -\log p(a|c)$ and $\mathrm{NLL_q} = -\log p(q|a, c)$. The better a model performs in this task, the better it fits the test set. As a baseline, to assess the effect of incorporating latent random variables, we implemented a pipeline model similar to Subramanian et al. (2018) using a deterministic pointer network.

Result Table 3.7 shows the result of QA pair modeling. First, our models with $\mathrm{C} = 0$ are superior to the pipeline model, which means that introducing latent random variables aid QA pair modeling capacity. However, the KL terms converge to zero with $\mathrm{C} = 0$. When we set $C > 0$, KL values are greater than 0, which implies that latent variables have non-trivial information about questions and answers. Also, we observe that the target value of KL C can control the KL values, showing the potential to avoid the posterior collapse issue.

	NLL	NLL_a	NLL_q	D_{KL_z}	D_{KL_y}
Pipeline	36.26	3.99	32.50	-	-
VQAG					
$\mathrm{C} = 0$	**34.46**	4.46	30.00	0.027	0.036
$\mathrm{C} = 5$	37.00	5.15	31.51	4.862	4.745
$\mathrm{C} = 20$	59.66	14.38	43.56	17.821	17.038
$\mathrm{C} = 100$	199.43	81.01	112.37	92.342	91.635

Table 3.7: QA pair modeling capacity measured on $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$. We used the same value C for the target values of KL $\mathrm{C_a}$ and $\mathrm{C_q}$ for simplicity. NLL: negative log likelihood of QA pairs. NLL_a (NLL_q): NLL of answers (questions). D_{KL_z} and D_{KL_y} are Kullback–Leibler divergence in Eq. 3.1. NLL for our models are estimated with importance sampling using 300 samples.

Synthetic Datasets	$(\mathrm{C_a}, \mathrm{C_q})$
$\mathcal{D}_{5,5}$	(5, 5)
$\mathcal{D}_{5,20}$	(5, 20)
$\mathcal{D}_{20,20}$	(20, 20)

3.3.7 Synthetic Dataset Construction

We created three synthetic QA datasets, denoted as $\mathcal{D}_{5,5}$, $\mathcal{D}_{20,20}$, and $\mathcal{D}_{5,20}$, using VQAG with the different configurations, $(\mathrm{C_a}, \mathrm{C_q}) = (5, 5), (20, 20), (5, 20)$ respectively. These configurations are chosen based on the recall-based metrics and diversity scores in the AE and QG results.

VQAG generated 50 QA pairs from each paragraph in $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{train}}$ to construct each $\mathcal{D}$. It is generally known that VAEs generate diverse but low-quality data unlike GANs. We used heuristics to filter out low-quality generated QA pairs, dropping questions that are longer than 20 words or shorter than 5 words and answers that are longer than 10 words, keeping questions that have at least one interrogative word, and removing n-gram repetition in questions. While some existing works used the BERT QA model or an entailment model as a data filter (Alberti et al., 2019; Zhang and Bansal, 2019; Liu et al., 2020a), our heuristics are enough to obtain improvement in the downstream QA task as shown in §3.3.9. Some samples in our datasets are given in Table 3.6, showing that the diverse QA pairs are generated. See §3.4 to see how VQAG maps the latent variables to the QA pairs.

3.3.8 Human Evaluation

We assess the quality of the synthetic QA pairs by conducting human evaluation on Amazon Mechanical Turk. For human evaluation, we randomly chose 200 samples from synthetic QA pairs generated by Zhang and Bansal (2019) and our model with $(\mathrm{C_a}, \mathrm{C_q}) = (5, 5), (20, 20)$ from the paragraphs in $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$. We also chose 100 samples from $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$. In addition to the three items proposed by Liu et al. (2020a), we asked annotators if an given answer is important, i.e., it is worth being asked about. We showed the workers a triple (passage, question, answer) and asked them to answer the four questions as shown below.

1. **Is the question well-formed in itself?** The workers are asked to select *yes* if a given question is both grammatical and meaningful. The workers select *understandable* if a question is not grammatical but meaningful.
2. **Is the question relevant to the passage?** This is to check whether a question is relevant to the content of a passage.
3. **Is the answer a correct answer to the question?** If a given answer partially overlaps with the true answer in a passage, the workers select *partially*.
4. **Is the meaning of the answer in itself related to the main topic of the passage?** This is to check the importance of an answer. We designed this question to assess the question-worthiness of an answer.

Each triple is evaluated by three crowdworkers. Each task costs 0.08 USD. We report the responses obtained using the majority vote.

According to the results in Table 3.8, nearly 25% of our questions are not understandable or meaningful, and 30% of our answers are incorrect for the generated questions. This result indicates that our synthetic datasets contain a considerable number of noisy QA pairs in these two aspects. However, 90% of the generated questions are relevant to

Experiments		SemQG	$(C_a, C_q) = (5, 5)$	$(C_a, C_q) = (20, 20)$	SQuAD
Question is well-formed	No	2.9%	23.1%	27.8%	2.3%
	Understandable	34.5%	16.0%	17.0%	10.5%
	Yes	62.6%	60.9%	55.1%	87.2%
Question is relevant	No	2.5%	9.5%	11.5%	4.0%
	Yes	97.5%	90.5%	88.5%	96.0%
Answer is correct	No	2.8%	28.8%	30.5%	7.5%
	Partially	21.8%	28.1%	26.6%	11.8%
	Yes	75.4%	43.2%	42.9%	80.6%
Answer is important	No	1.5%	10.0%	5.0%	6.0%
	Yes	98.5%	90.0%	95.0%	94.0%

Table 3.8: Human evaluation of the quality of QA pairs. C_a and C_q are the hyperparameters in Eq. 3.1.

the passages, and 90% of the answers extracted by our models are question-worthy. As we will verify in §3.3.9, our noisy but diverse synthetic datasets are effective in enhancing the QA performance in the in- and out-of-distribution test sets.

3.3.9 Question Answering

We evaluated QAG methods on the downstream QA task. We evaluated our method on 12 challenge sets in addition to the in-distribution test set.

Baselines

We compared our method with the following baselines.

- **SQuAD$^{\text{Du}}_{\text{train}}$** BERT-base model trained on SQuAD$^{\text{Du}}_{\text{train}}$ without data augmentation.
- **HarQG** (Du and Cardie, 2018) uses neural AE and QG models and generates over one million QA pairs from top ranking Wikipedia articles not included in SQuAD. We used the publicly available dataset.[5]
- **SemQG** (Zhang and Bansal, 2019) uses reinforcement learning to generate more SQuAD-like questions. We reran the trained model, and generated questions from the same context-answer pairs as HarQG.
- **InfoHCVAE** (Lee et al., 2020) uses a variational QAG model with an information-maximizing term. We trained this model[6] on SQuAD$^{\text{Du}}_{\text{train}}$, and then generated 50 QA pairs from each context in SQuAD$^{\text{Du}}_{\text{train}}$ for a fair comparison with VQAG.

Training Details of Question Answering Models

We trained pretrained BERT-base models (Devlin et al., 2019) on each synthetic dataset, and then fine-tuned it on SQuAD$^{\text{Du}}_{\text{train}}$. We adopted this procedure following existing data augmentation approach for QA (Dhingra et al., 2018; Zhang and Bansal, 2019). In our study, the order in which our synthetic datasets $\mathcal{D}$ were given to a QA model was tuned on the dev set.

We used the Hugging Face's implementation of BERT (Wolf et al., 2020). We used Adam (Kingma and Ba, 2014) with epsilon as 1e-8 for the optimizer. The batch size

[5] https://github.com/xinyadu/harvestingQA

[6] https://github.com/seanie12/Info-HCVAE

was 32. In both the pretraining and fine-tuning procedure, the learning rate decreased linearly from 3e-5 to zero. We conducted the training for one epoch using a synthetic dataset and two epochs using the original training set.

In addition to the performance of *Single* models, we reported the performance of *Ensemble* models, where the output probabilities of three different QA models are simply averaged. In practice, the top 20 candidate answer spans predicted by each QA model were used for the final prediction.

	Training Data (Size)	Challenge Sets												
		$\text{SQuAD}^{\text{Du}}_{\text{test}}$	News	NQ	Quo	Para	APara	Hard	Imp	Add	AddO	MT	ASR	KB
Single	$\text{SQuAD}^{\text{Du}}_{\text{train}}$ (76k)	83.5	49.2	67.7	30.1	85.7	50.2	75.6	64.7	62.9	71.8	79.7	67.5	80.1
	+HarQG (1,205k)	*83.3*	*48.5*	*66.2*	31.3	*85.2*	56.5	*73.0*	*63.5*	65.1	73.1	*78.6*	70.0	80.3
	+SemQG (1,204k)	84.7	50.5	69.8	***34.5***	86.2	51.8	*75.0*	65.1	***66.5***	74.3	*79.5*	71.0	80.7
	+InfoHCVAE (824k)	**84.8**	**51.3**	**71.2**	33.8	85.6	53.3	**77.7**	64.8	66.1	**74.5**	**81.3**	**71.6**	**82.8**
	+VQAG (432k)	84.5	49.2	70.1	32.0	**86.7**	**59.0**	76.1	**66.3**	64.8	73.9	79.9	70.5	81.1
Ensemble	{$\text{SQuAD}^{\text{Du}}_{\text{train}}$}*3	84.2	50.4	69.4	31.3	86.4	53.2	76.6	65.7	63.6	72.6	80.3	68.7	81.2
	{+SemQG}*3	85.5	51.8	71.3	**35.1**	87.5	57.8	78.2	66.5	67.0	75.1	80.8	72.9	82.5
	{+InfoHCVAE}*3	85.3	52.0	**72.2**	34.0	88.0	56.9	**79.0**	65.7	**67.7**	**75.9**	81.4	73.1	**83.2**
	{+VQAG}*3	84.9	50.9	70.1	32.3	**88.1**	**58.6**	77.3	**67.5**	64.9	73.9	80.8	71.2	81.6
	{+Sem,+Info,+V}	**85.8**	**52.1**	72.0	34.2	88.0	55.1	78.8	67.0	66.3	74.7	**82.2**	**73.5**	83.0
	If challenge set is known	-	62.9	83.0	66.9	88.6	73.9	-	-	-	-	80.8	75.9	82.6

Table 3.9: QA performance (F1 score) on $\text{SQuAD}^{\text{Du}}_{\text{test}}$ and the 12 challenge sets. The abbreviations of the challenge sets are explained in §3.3.9. Curly brackets denote an ensemble of different models (e.g., {+VQAG}*3 denotes the ensemble of three QA models, trained with different random seeds after data augmentation with VQAG). The best scores for each of the *Single* and *Ensemble* models are **boldfaced**. The degraded scores compared to the no data augmentation baseline (the 1st line) are in *red*. Sem: SemQG, Info: InfoHCVAE, V: VQAG.

Challenge Sets

We assessed the robustness of the QA models to the following 12 challenge sets, as well as $\text{SQuAD}^{\text{Du}}_{\text{test}}$.

- **NewsQA (News)** (Trischler et al., 2017): 5,166 QA pairs created from CNN articles by crowdworkers, transformed into the SQuAD format following Sen and Saffari (2020).
- **Natural Questions (NQ)** (Kwiatkowski et al., 2019): 2,356 questions from real users for Wikipedia articles. We reframed NQ as extractive QA by using long answers in NQ as contexts following Sen and Saffari (2020).[7]
- **Non-Adversarial Paraphrased Test Set (Para)** (Gan and Ng, 2019): 1,062 questions paraphrased with slight perturbations from SQuAD using a trained paraphrased model.
- **Adversarial Paraphrased Test Set (APara)** (Gan and Ng, 2019): 56 questions manually paraphrased using context words near a confusing answer from SQuAD.
- **Hard Subset (Hard)** (Sugawara et al., 2018): A subset of the SQuAD dev set, which consists of 1,661 questions that require less word matching and more knowledge inference and multiple sentence reasoning.
- **Implications (Imp)** (Ribeiro et al., 2019): 13,371 QA pairs automatically derived from the SQuAD dev set with a linguistic rule-based method.[8]
- **AddSent (Add) & AddOneSent (AddO)** (Jia and Liang, 2017): Adversarial SQuAD dataset created using handcrafted rules designed for fooling a QA model. The sizes of Add and AddO are 3,560 and 1,787, respectively.
- **Quoref (Quo)** (Dasigi et al., 2019): 2,418 questions requiring coreference resolution created by humans. We used the dev set.
- **Natural Machine Translation Noise (MT)** (Ravichander et al., 2021): A subset of NoiseQA, consisting of 1,190 English translated questions produced by Google's commercial translation system from the XQuAD dataset (Artetxe et al., 2020). This creation introduces naturally occurring noise caused by machine translation.
- **Natural Automatic Speech Recognition Noise (ASR)** (Ravichander et al., 2021): Another subset of NoiseQA, consisting of 1,190 questions that include automatic speech recognition error.
- **Natural Keyboard Noise (KB)** (Ravichander et al., 2021): Another subset of NoiseQA, consisting of 1,190 questions that include natural character-level typos introduced by typing questions on a keyboard.

These challenge sets enable us to evaluate the QA models' robustness to other domain corpora, variations in questions, adversarial examples, and noise that may occur in real-world applications.

Results

The overall results are given in Table 3.9. First, we discovered that the QA model without data augmentation degraded the performance on the 12 challenge sets, showing a lack of the robustness to the natural and adversarial distribution shifts in contexts, questions, and answers.[9]

[7] We used answerable questions for NewsQA and NQ provided by Sen and Saffari (2020). We did not use the MRQA shared task version as Lee et al. (2020) did.

[8] For example, "Q: *Who died in 1285?* A: *Zhenjin*" is derived from "Q: *When did Zhenjin die?* A: *1285*"

[9] The score on Para—85.7 F1 is degraded when compared to the score on the SQuAD dev set—87.9 F1, which is the source for creating Para. This means the lack of robustness to paraphrased questions as shown in Gan and Ng (2019).

With data augmentation using QAG, the in-distribution scores were generally improved, except for HarQG. In the *Single* model setting on the challenge sets, SemQG achieved the best performance on Quo and Add. InfoHCVAE achieved the best performance on News, NQ, Hard, AddO, MT, ASR, and KB. VQAG achieved the best performance on Para, APara, and Imp. These results imply that different QAG methods have different benefits. In the *Ensemble* setting, taking the best of the three, the scores on $\text{SQuAD}^{\text{Du}}_{\text{test}}$, News, MT, and ASR were further improved with {+Sem,+Info,+V}.

We also attached scores that are obtained *if challenge set is known* in Table 3.9; that is, natural or synthetic samples from the same distributions as the challenge sets are available during training. For News, NQ, and Quo, we trained the BERT-base model on the corresponding training sets, which are annotated by humans. For paraphrased questions (Para, APara) and NoiseQA (MT, ASR, and KB), the scores were taken from Gan and Ng (2019) and Ravichander et al. (2021), respectively. These scores can be considered as the upper bounds. In NoiseQA, the QAG methods consistently improved the scores, even though they were not designed for the noise. This may be because the lack of quality in synthetic datasets, as shown in Table 3.8, unintentionally improved the robustness to the noise. However, the most significant performance gap (> 30 F1) between the upper bound and the no data augmentation baseline was observed in Quo. This result indicates that a QA model does not acquire coreference resolution from SQuAD, even though approximately 18% of SQuAD questions require coreference resolution (Sugawara et al., 2018). The QAG methods mitigated this gap to some extent, but there is a significant room for improvement.

The improvement in NQ is generally more prominent than that in News. This may be because both SQuAD and NQ contain paragraphs in Wikipedia. Utilizing unlabeled documents in domains such as news articles may improve the generalization to other domains such as News.

In paraphrased questions (Gan and Ng, 2019), implications (Ribeiro et al., 2019), and NoiseQA (Ravichander et al., 2021), augmenting questions that are similar to the corresponding challenge sets, that is, generating paraphrases, implications, and questions including the noise, successfully improved the robustness to these perturbations. While these methods slightly degraded or maintained the in-distribution score, we showed that QAG methods are less likely to exhibit a trade-off between the in- and out-of-distribution accuracies. Notably, VQAG did not degrade the scores on all the 12 challenge sets while improving the in-distribution score. In contrast, SemQG degraded the scores on Hard and MT, and InfoHCVAE degraded the score on Para. This property of VQAG may be because it can significantly improve the diversity by combining the different configurations.

Moreover, the size of the synthetic dataset created by VQAG was the smallest among the QAG methods as shown in Table 3.9. If the diversity is assured sufficiently, significantly increasing the quantity may not be necessary. In Add and AddO, we showed that the QAG methods consistently improved adversarial robustness, which has not been studied in the QAG literature.

Ablation Study

To assess the usefulness of each dataset $\mathcal{D}$ in VQAG, we conducted an ablation study. As shown in Table 3.10, each dataset $\mathcal{D}$ has meaningful effect on the performance. This result implies that creating more synthetic datasets using different configurations may further improve the performance.

To understand the differences in each dataset in terms of diversity, we conducted a simple analysis on the question type. As shown in Table 3.11, VQAG with different configurations corresponds to different distributions of question types, while more than

Training Data (Size)	EM	F1
VQAG (432k)	81.49	88.61
$-\mathcal{D}_{5,5}$ (251k)	81.04	88.39
$-\mathcal{D}_{5,20}$ (113k)	81.00	88.48
$-\mathcal{D}_{20,20}$ (68k)	81.14	88.52

Table 3.10: Ablation study on $\text{SQuAD}^{\text{Du}}_{\text{dev}}$. Each synthetic dataset is shown to be useful to improve the scores.

Dataset	what	how	who	which	when	where	why
$\text{SQuAD}^{\text{Du}}_{\text{train}}$	<u>58.3</u>	10.4	10.3	6.7	6.7	4.2	1.5
$\text{SQuAD}^{\text{Du}}_{\text{test}}$	<u>56.5</u>	12.1	11.5	8.6	6.0	3.8	0.8
HarQG	<u>61.3</u>	7.8	13.8	0.7	10.1	5.8	0.5
SemQG	<u>71.1</u>	8.1	12.8	1.3	3.6	2.7	0.2
InfoHCVAE	<u>77.1</u>	6.6	5.0	1.6	5.6	3.3	0.5
VQAG							
$\mathcal{D}_{5,5}$	36.6	<u>54.9</u>	4.9	0.5	0.3	0.5	2.3
$\mathcal{D}_{5,20}$	9.5	35.5	3.6	<u>49.2</u>	1.2	0.9	0.0
$\mathcal{D}_{20,20}$	28.2	<u>36.7</u>	6.3	23.2	0.2	1.6	3.9
							(%)

Table 3.11: Percentages (%) of each question type in each dataset. The largest number in each line is <u>underlined</u>. VQAG is less likely to contain "what" and more likely to contain "which" and "how" than other data sets.

50% of the questions in the other datasets contain "what". Among the QAG methods, this point is unique to VQAG.

3.4 Analysis: Latent Interpolation

To intuitively understand what VQAG learns, we conducte latent interpolation. Table 3.12 shows the latent interpolation between two ground-truth QA pairs using VQAG with $(\mathrm{C_a}, \mathrm{C_q}) = (5, 20)$. This result shows that z controls answer and y controls question.

3.5 Revisiting the QA and QG Performance in Terms of Question–Context Lexical Overlap

In this paper, we denote question–context lexical overlap as QCLO. We define QCLO as the ratio of the overlapping words between question Q and context C to the total number of words in question.[10] Precisely, QCLO is calculated as

$$\text{QCLO} = \frac{|\boldsymbol{Q} \cap \boldsymbol{C}|}{|\boldsymbol{Q}|}. \tag{3.26}$$

The second example in Table 3.1 indicates a question with lower QCLO is neither answered nor generated correctly by neural models. To investigate this phenomenon, we first analyze the QA and QG performance in terms of QCLO.

[10]When computing lexical overlap, we do not exclude stop words because even overlapping stop words are important cues to determine the correct answer.

	z_1	z_2	z_3	z_4	z_5
y_1	**in what city and state did beyonce grow up ?—houston , texas**	how do competitions performed a child child ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé album album ?—dangerously in love
y_2	the album born and raised ?—houston , texas	how do competitions enovid ?—dancing	how is actress - carter ?—songwriter	how did beyoncé 's album album ?—dangerously in love	how did beyoncé album album ?—dangerously in love
y_3	the album born and raised ?—houston , texas	how do competitions performed a child child ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé dobruja to ?—dangerously in love
y_4	the album born and raised ?—houston , texas	how many competitions does texas child perform ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	how did beyoncé dobruja to ?—dangerously in love
y_5	the album born and raised ?—houston , texas	how many competitions did texas child perform ?—dancing	the american singer born what american singer ?—songwriter	how did beyoncé dobruja to ?—dangerously in love	**what was the name of beyoncé 's first solo album ?—dangerously in love**

Table 3.12: Latent interpolation with VQAG with $(\mathrm{C_a}, \mathrm{C_q}) = (5, 20)$. The samples in the upper left and lower right are the ground truth QA pairs from the same paragraph as Table 3.6. The linearly interpolated samples show how our generative model learns mapping from latent space to QA pairs.

Experimental setups For QA, we use the fine-tuned BERT-base and -large models (Devlin et al., 2019). For QG, we use SemanticQG (Zhang and Bansal, 2019).[11] For the dataset, we use the SQuAD-Du dataset; the train, dev, and test split of SQuAD v1.1 (Rajpurkar et al., 2016) proposed by Du et al. (2017), which we denote as $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{train}}$, $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{dev}}$, $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$, respectively. This split is commonly used in the QG literature (Du and Cardie, 2018; Zhang and Bansal, 2019) because the original test set is not released. The numbers of question, answer and context triples in $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{train}}$, $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{dev}}$, and $\mathrm{SQuAD}^{\mathrm{Du}}_{\mathrm{test}}$, are 76k, 11k, and 12k, respectively.

Results We show the result in Figure 3.2. This indicates that the performance of the BERT models on the questions with lower QCLO is degraded compared to the questions with higher QCLO. For QG, the BLEU-4 score (Papineni et al., 2002) is highly correlated with QCLO, which means that the model fails to generate questions with low QCLO accurately.

We also show the distributions in terms of QCLO of questions generated by recent neural QG models (HarvestingQG (Du and Cardie, 2018), SemanticQG (Zhang and Bansal, 2019), InfoHCVAE (Lee et al., 2020), and VQAG (Shinoda et al., 2021a)) in Figure 3.3. This indicates that all the QG models are biased towards generating ques-

[11] We used the ELMo+QPP&QAP (Zhang and Bansal, 2019) model for QG.

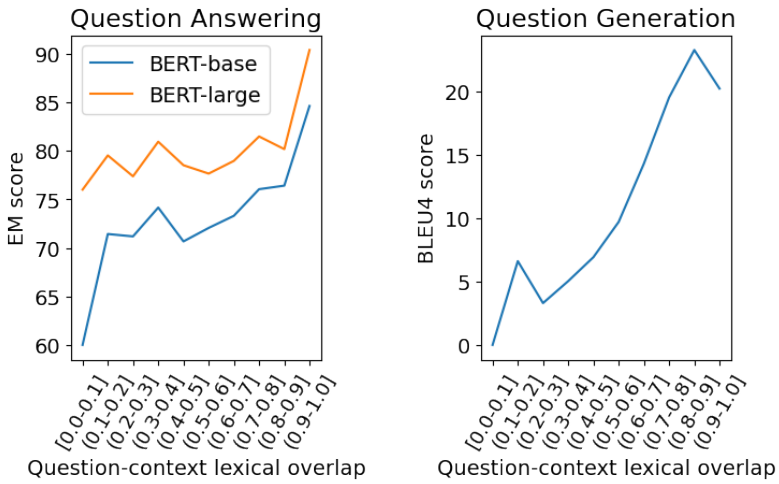


Figure 3.2: Exact match (EM) score of BERT models and BLEU-4 score of SemanticQG (Zhang and Bansal, 2019) on the test set of SQuAD-Du for each range of Question–Context Lexical Overlap (QCLO). See Eq. 3.26 for the definition of QCLO. Both the QA and QG models degrade the scores on questions with low QCLO.

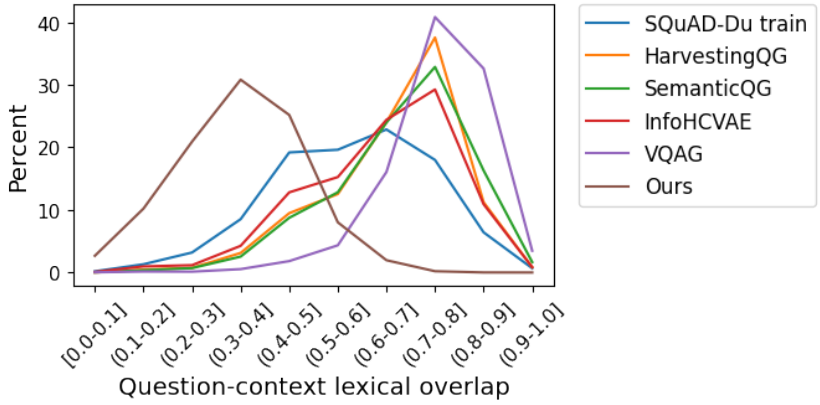


Figure 3.3: The percentages of questions in the datasets, SQuAD-Du (Du et al., 2017), HarvestingQG (Du and Cardie, 2018), SemanticQG (Zhang and Bansal, 2019), InfoHCVAE (Lee et al., 2020), VQAG (Shinoda et al., 2021a), and ours (§3.6), for each range of QCLO. While neural question generation models are biased towards generating questions with high QCLO, ours can generate questions with low QCLO.

tions with higher QCLO than $\text{SQuAD}^{\text{Du}}_{\text{train}}$, which is used to train those QG models.

Based on the result, we suspect that when neural QG is used to augment a QA dataset, the degraded QG performance on questions with low QCLO could exacerbate the degraded QA performance. Our experiments in §3.7 show that this is often true. We hypothesize that this is caused by the strong tendency of neural QG models to generate questions with high QCLO as shown in Figure 3.3.

3.6 Method II: Synonym Replacement for Reducing Lexical Overlap Bias

We assume that if we augment questions with low QCLO unlike existing neural QG approaches, the robustness of QA models to questions with low QCLO can be improved. In this section, we describe the proposed method for generating questions with low QCLO. We extend the idea of synonym replacement used in Wei and Zou (2019) to reduce the

lexical overlap. The proposed method is as follows:

1. List all the overlapping words between question and context.
2. Replace every word in the listed words other than predefined stop words with one of its synonyms chosen randomly from WordNet (Miller, 1995), and obtain a synthetic question.
3. If the lexical overlap decreases after synonym replacement, add the synthetic question to our dataset; if not, discard the question.

After repeating this procedure once for every ground-truth question in the training set, we obtain 70k synthetic questions with significantly lower lexical overlap, as indicated in Figure 3.3 (ours). For example, *What is heresy mainly at odds with?* is converted into *What is heterodoxy mainly at odds with?*, and *How many documents remain classified?* is converted into *How many text file remain classified?*. Because *heterodoxy*, *text*, and *file* do not appear in the contexts, the lexical overlap is reduced in each example.

It is worth mentioning a couple of limitations of our method. First, synonym replacement may slightly change the meaning of questions depending on the context. Second, our approach relies on the assumption that annotated questions are available, which makes it impossible to apply to unlabeled passages.

3.7 Experiments II: Robustness to Questions with Low Lexical Overlap

To determine the effect of data augmentation on improving the QA model robustness to questions with low QCLO, we conducted experiments with several QG approaches.

3.7.1 Dataset

We used the SQuAD-Du dataset as in §3.5. Considering the QCLO statistics of SQuAD displayed in Figure 3.3, we split $\text{SQuAD}^{\text{Du}}_{\text{dev}}$ and $\text{SQuAD}^{\text{Du}}_{\text{test}}$ into **Easy** and **Hard** subsets that contain questions with QCLO greater than 0.3, and the others, respectively. Our Easy and Hard subsets offered concise, yet sufficient, evaluation in terms of QCLO.

3.7.2 Baselines

We adopted the following four baselines that use neural QG models for data augmentation.

- **HarvestingQG** (Du and Cardie, 2018) generates question–answer pairs from 10,000 top-ranking Wikipedia articles with neural answer extraction and question generation.[12] The size is 1.2 million.
- **SemanticQG** (Zhang and Bansal, 2019) is a QG model that uses reinforcement learning to generate semantically valid questions. Following this work, we generated questions using the publicly available model[13] from the same context–answer pairs as HarvestingQG. The size is 1.2 million.
- **InfoHCVAE** (Lee et al., 2020) is a question–answer pair generation model based on conditional variational autoencoder with mutual information maximization. We trained this model on $\text{SQuAD}^{\text{Du}}_{\text{train}}$, and then generated 50 questions and answers from each context in $\text{SQuAD}^{\text{Du}}_{\text{train}}$. The size is 824k.

[12] https://github.com/xinyadu/harvestingQA
[13] https://github.com/ZhangShiyue/QGforQA

- **VQAG** (Shinoda et al., 2021a) is a question–answer pair generation model based on conditional variational autoencoder with explicit KL control. We used the publicly available dataset.[14] The size is 432k.

The distributions of the lexical overlap of these datasets are presented in Figure 3.3. We indicate that these methods are more biased towards high lexical overlap than $\text{SQuAD}^{\text{Du}}_{\text{train}}$, which was used as the training set for these QG models.

3.7.3 Experimental Setups

As in our previous experiment (§3.5), we used BERT-base and -large models, whose total number of parameters are 110M and 340M, respectively. Dhingra et al. (2018) proposed to pretrain a QA model using synthetic data composed of cloze-style questions and then fine-tune it on the ground-truth data. We adopted the pretrain-and-fine-tune approach for the neural QG approaches, which generated over 1.2 million questions. However, as discussed by Zhang and Bansal (2019), we observed that when the size of the synthetic data was small or similar to the ground-truth data, a performance gain could not be obtained by the pretrain-and-fine-tune approach. Thus, for the proposed approach, which generated 70k questions, we fine-tuned QA models on the ground-truth data randomly mixed with the generated data. We used the Hugging Face's implementation of BERT (Wolf et al., 2019). We use the Adam (Kingma and Ba, 2014) optimizer with epsilon set to 1e-8. The batch size was 32 for all the settings. In both the pretraining and fine-tuning procedure, the learning rate decreased linearly from 3e-5 to zero. We train the QA models for one epoch for pretraining with synthetic data and two epochs for fine-tuning with $\text{SQuAD}^{\text{Du}}_{\text{train}}$.

[14] https://github.com/KazutoshiShinoda/VQAG

		$\text{SQuAD}^{\text{Du}}_{\text{dev}}$ (EM/F1)			$\text{SQuAD}^{\text{Du}}_{\text{test}}$ (EM/F1)		
Model	Train Source	Hard	Easy	ALL	Hard	Easy	ALL
base	$\text{SQuAD}^{\text{Du}}_{\text{train}}$	72.31/81.11	80.74/88.39	80.35/88.05	70.88/81.99	73.22/84.75	73.06/84.57
	+ HarvestingQG	70.25/78.27	80.06/87.62	79.60/87.19	69.28/79.92	73.15/84.20	72.90/83.93
	+ SemanticQG	70.45/80.25	81.70/89.08	81.17/88.67	71.68/82.49	**74.39/85.59**	**74.21/85.39**
	+ InfoHCVAE	72.05/80.66	81.79/**89.35**	81.34/**88.95**	73.47/**83.91**	73.50/85.08	73.48/84.99
	+ VQAG	73.29/82.04	**81.88**/88.93	**81.48**/88.62	71.60/83.07	73.79/85.23	73.63/85.08
	+ Ours	**73.50/82.81**	80.34/87.81	80.02/87.58	**73.60**/83.49	73.08/84.41	73.11/84.34
large	$\text{SQuAD}^{\text{Du}}_{\text{train}}$	78.72/87.71	**87.06/93.23**	**86.67/92.98**	77.93/87.84	**79.33/89.88**	**79.24/89.74**
	+ HarvestingQG	79.13/86.92	85.55/92.12	85.26/91.88	76.99/86.61	77.58/88.28	77.54/88.17
	+ SemanticQG	79.96/87.73	85.90/92.57	85.62/92.35	76.99/87.29	77.82/88.68	77.77/88.59
	+ InfoHCVAE	77.85/86.44	85.25/92.15	84.91/91.89	76.00/87.55	78.02/88.90	77.87/88.80
	+ VQAG	79.50/87.55	86.68/93.01	86.35/92.76	77.33/87.70	78.98/89.36	78.86/89.25
	+ Ours	**81.37/88.33**	86.49/92.78	86.25/92.57	**78.40/88.52**	77.94/89.00	77.96/88.97

Table 3.13: QA performance with data augmentation. EM/F1 scores on the Hard (where $\text{QCLO} \leq 0.3$) and Easy (where $\text{QCLO} > 0.3$) subsets, and the whole set of $\text{SQuAD}^{\text{Du}}_{\text{dev}}$ and $\text{SQuAD}^{\text{Du}}_{\text{test}}$ are reported.

3.7.4 Results

The results of the data augmentation are displayed in Table 3.13. In all the settings, the proposed approach achieved the best EM score on the Hard subset. Notably, the proposed method significantly improved the performance by **2.72 (EM) / 1.50 (F1)** points using BERT-base on the Hard subset in the test set, while maintaining the overall scores compared to the no data augmentation baseline. This improvement indicates that the proposed approach for debiasing the dataset in terms of QCLO is helpful for addressing the performance degradation. However, the proposed approach degraded the scores on the Easy subsets when using BERT-large. Addressing the trade-off between the scores in the Hard and Easy subsets using BERT-large is future work.

When using BERT-base, the neural QG baselines except for HarvestingQG improved the scores on the Easy subset; however, the baselines except for InfoHCVAE often degraded the scores on the Hard subset. This could be due to the tendency to generate questions with high QCLO (Figure 3.3).

When using BERT-large, the QG approaches often fail to improve the scores in both the Hard and Easy subsets. Generating useful examples for a larger model is more challenging than for a smaller one according to these results. Utilizing pretrained language models for QG may be useful given the fact that only RNNs are used in all the baseline QG methods in our experiments.

HarvesingQG was not effective in almost all the settings. Comparing its scores with those of SemanticQG, which used the same context–answer pairs as HarvestingQG, some feature of generated questions other than lexical overlap appeared to be critical in improving the QA scores on the Easy subset, because the distributions of QCLO of two synthetic datasets were similar to each other (see Figure 3.3).

For further boosting the overall average score, we can make an ensemble prediction using the best performing models in the Easy and Hard subsets, although improving the overall scores is not the main focus in this paper. The performance gains were positive but not very significant in our case. We leave utilizing the ensemble prediction to address the performance trade-off to future work.

3.8 Analysis: Case Study

To demonstrate the effect of the baseline QG models and proposed method qualitatively, we present examples in both the Hard (QCLO ≤ 0.3) and Easy (QCLO > 0.3) subsets in Table 3.14. The first two examples show that only the QA model trained with the proposed method could correctly answer the questions. Answering the questions in these examples required a knowledge of synonyms, such as "recreational" vs. "entertainment," "besides" vs. "aside from," "employees" vs. "workers," and "kill oneself" vs. "commit suicide." These examples imply that the proposed data augmentation method based on synonym replacement enabled the QA model to acquire knowledge regarding synonyms. This kind of reasoning beyond superficial word matching is indispensable for QA systems to achieve human-level language understanding.

The third example in Table 3.14 displays an example where data augmentation using the neural QG models made the original prediction incorrect. This example implies that current QG models may harm the robustness of QA models to questions with low QCLO. As Geirhos et al. (2020) discussed, if QG models just amplify the dataset bias, QA models could learn dataset-specific solutions (i.e., shortcuts) and fail to generalize to challenge test sets.

In contrast, the fourth and fifth examples in Table 3.14 display examples in the Easy subset where data augmentation with neural QG models is beneficial, while the original and proposed models fail to answer them correctly. These examples require multiple-

Besides earning a reputation as a respected entertainment *(Original, InfoHCVAE)* device, the iPod has also been accepted as a **business** *(Ours)* device. Government departments, major institutions and international organisations have turned to the iPod line as a delivery mechanism for business communication and training, such as the Royal and Western Infirmaries *(HarvestingQG, SemanticQG, VQAG)* in Glasgow, Scotland, where iPods are used to train new staff. — Aside from recreational use, in what other arena have iPods found use? (QCLO: 0.29)
In **2010** *(Ours)*, a number of workers committed suicide at a Foxconn operations in China. Apple, HP, and others stated that they were investigating the situation. Foxconn guards have been videotaped beating employees. Another employee killed himself in 2009 *(Original, HarvestingQG, SemanticQG, VQAG)* when an Apple prototype went missing, and claimed in messages to friends, that he had been beaten and interrogated. — In what year did Chinese Foxconn emplyees* kill themselves? (*: annotator's typo) (QCLO: 0.2)
The BBC began its own regular television programming from the basement of Broadcasting House, London, on 22 August 1932 *(HarvestingQG, SemanticQG)*. The studio moved to larger quarters in 16 Portland Place, London, in **February 1934** *(Original, Ours)*, and continued broadcasting the 30-line images, carried by telephone line to the medium wave transmitter at Brookmans Park, until 11 September 1935, by which time advances in all-electronic television systems made the electromechanical broadcasts obsolete. — When did the BBC first change studios? (QCLO: 0.25)
Peyton Manning *(VQAG)* became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39 *(Original, Ours)*. The past record was held by **John Elway** *(HarvestingQG, SemanticQG, InfoHCVAE)*, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. — Prior to Manning, who was the oldest quarterback to play in a Super Bowl? (QCLO: 0.88)
Despite being relatively unaffected by the embargo *(Original, HarvestingQG, VQAG, Ours)*, the UK nonetheless faced an oil crisis of its own - **a series of strikes by coal miners and railroad workers** *(SemanticQG, InfoHCVAE)* over the winter of 1973–74 became a major factor in the change of government. Heath asked the British to heat only one room in their houses over the winter. The UK, Germany, Italy, Switzerland and Norway banned flying, driving and boating on Sundays. Sweden rationed gasoline and heating oil. The Netherlands imposed prison sentences for those who used more than their ration of electricity. — What caused UK to have an oil crisis in its own country? (QCLO: 0.62)

Table 3.14: Illustrative predictions on $\text{SQuAD}^{\text{Du}}_{\text{dev}}$ and $\text{SQuAD}^{\text{Du}}_{\text{test}}$ by a BERT-base model trained on $\text{SQuAD}^{\text{Du}}_{\text{train}}$ (Original), +HarvestingQG, +SemanticQG, +InfoHCVAE, +VQAG, and +Ours. The ground truth answers are in **bold**. The incorrectly predicted answers are written in red. The QA models that predict them are written in *italics*. The overlapping words in the questions are underlined. Question–context lexical overlap (QCLO) is given in parentheses.

sentence reasoning, i.e., one has to read and understand multiple sentences to answer these questions. This observation implies that some under-represented features (e.g., multiple-sentence reasoning (Rajpurkar et al., 2016)) exist even in the Easy subset, and the existing neural QG models might amplify such features (possibly by copying many words from multiple sentences to formulate questions) and make it easy to capture them. Investigating what kind of features are learned by using data augmentation with neural QG models in more detail is future work.

3.9 Related Work: Question-Answer Pair Generation

3.9.1 Robustness of QA models

Pretrained language models such as BERT (Devlin et al., 2019) have surpassed the human score on the SQuAD leaderboard.[15] However, such powerful QA models have been shown to exhibit the lack of robustness. A QA model that is trained on SQuAD is not robust to paraphrased questions (Gan and Ng, 2019), implications derived from SQuAD (Ribeiro et al., 2019), questions with low lexical overlap (Sugawara et al., 2018), and other QA datasets (Yogatama et al., 2019; Talmor and Berant, 2019; Sen and Saffari, 2020). Ko et al. (2020) showed that extractive QA model can suffer from positional bias and fail to generalize to different answer positions.

The lack of robustness demonstrated in these studies can be explained by shortcut learning of deep neural networks (Geirhos et al., 2020). A high score on an in-distribution test set can be achieved by just exploiting unintended dataset biases (Levesque, 2014). Therefore, evaluating QA models only on an in-distribution test set is not enough to evaluate the robustness of the QA models.

3.9.2 Data Augmentation and Dataset Bias

Data augmentation has been widely used in other domains to reduce dataset biases such as the background bias in person re-identification (McLaughlin et al., 2015), the gender bias in coreference resolution (Zhao et al., 2018a), and the lexical bias in natural language inference (Zhou and Bansal, 2020). These works repeated training examples or added synthetic data to increase under-represented samples and reduce the imbalance in a training set. Our proposed approach has the same motivation as these works.

On the other hand, data augmentation can unintentionally introduce or amplify dataset bias. Back-translation (Sennrich et al., 2016), which is the common data augmentation approach for machine translation, can introduce the translationese bias. That is, machine translation systems trained with back-translation, compared to ones without back-translation, can enhance the BLEU scores when the input is translationese (i.e., human-translated texts) but harm the BLEU scores when the input is naturally occurring texts (Edunov et al., 2020; Marie et al., 2020). This phenomenon is analogous to the observation in our work, where we demonstrated that SQuAD QG models are biased towards generating questions with high QCLO, and this tendency can harm the QA performance on questions with low QCLO while improving that on questions with high QCLO.

3.9.3 Question Generation for Question Answering

QG has been studied extensively in order to augment QA datasets and boost the QA performance, which has been evaluated primarily on SQuAD (Du et al., 2017; Zhou et al., 2018; Yang et al., 2017a; Zhang and Bansal, 2019). Question answer pair generation, which consists of answer candidate extraction and QG, has been also received attention

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

because question-worthy answers for the input of QG are not freely available (Du and Cardie, 2018; Lee et al., 2020; Shinoda et al., 2021a). The de facto standard of QG models is to utilize a copy mechanism (Gu et al., 2016; Gulcehre et al., 2016). The tendency of QG models to copy words from textual contexts as indicated in Figure 3.3 is partially due to this copy mechanism. While the existing QG works have increased the BLEU scores on SQuAD[16] and successfully generated fluent questions in terms of human scores, the bias regarding lexical overlap in QG has not received sufficient attention.

3.9.4 Answer Extraction

AE aims to extract question-worthy phrases, which are worth being asked about, from each textual context without looking at the questions. AE has been performed mainly in two ways: rule-based and neural methods. (Yang et al., 2017a) extracted candidate phrases using rule-based methods such as named entity recognition (NER). However, not all the named entities, noun phrases, verb phrases, adjectives, or clauses in the given documents are used as gold answer spans. As such, these rule-based methods are likely to extract many trivial phrases.

Therefore, there have been studies on training neural models to identify question-worthy phrases. Du and Cardie (2018) framed AE as a sequence labeling task and used BiLSTM-CRF (Huang et al., 2015). Subramanian et al. (2018) treated the positions of answers as a sequence and used a pointer network (Vinyals et al., 2015). Wang et al. (2019b) used a pointer network and Match-LSTM (Wang and Jiang, 2016, 2017). Alberti et al. (2019) made use of pretrained BERT (Devlin et al., 2019).

However, these neural AE models are trained with maximum likelihood estimation; that is, each model is optimized to produce an answer set closest to the gold answers. In contrast, our model incorporates a latent random variable and is trained by maximizing the lower bound of the likelihood to extract diverse answers. In this study, we assume that there should be question-worthy phrases that are not used as the gold answers in a manually created dataset. We aim to extract such phrases.

3.9.5 Question Generation

Traditionally, QG was studied using rule-based methods (Mostow and Chen, 2009; Heilman and Smith, 2010; Lindberg et al., 2013; Labutov et al., 2015). After Du et al. (2017) proposed a neural sequence-to-sequence model (Sutskever et al., 2014) for QG, neural models that take context and answer as inputs have started to be used to improve question quality with attention (Bahdanau et al., 2014) and copying (Gulcehre et al., 2016; Gu et al., 2016) mechanisms. Most works focused on generating relevant questions from context-answer pairs (Zhou et al., 2018; Song et al., 2018; Zhao et al., 2018b; Sun et al., 2018; Kim et al., 2019; Liu et al., 2019a; Qiu and Xiong, 2019). These works showed the importance of answers as input features for QG. Other works studied predicting question types (Zhou et al., 2019; Kang et al., 2019), modeling a structured answer-relevant relation (Li et al., 2019), and refining generated questions (Nema et al., 2019). To further improve question quality, policy gradient techniques have been used (Yuan et al., 2017; Yang et al., 2017a; Yao et al., 2018; Kumar et al., 2019). Dong et al. (2019) used a pretrained language model.

The diversity of questions has been tackled using variational attention (Bahuleyan et al., 2018), a conditional variational autoencoder (CVAE) (Yao et al., 2018), and top p nucleus sampling (Sultan et al., 2020). Our study is different from these studies wherein we study QAG by introducing variational methods into both AE and QG. Lee et al.

[16] `http://aqleaderboard.tomhosking.co.uk/squad`

(2020) is the closest to our study in terms of the modeling choice. While Lee et al. (2020) introduced an information-maximizing term to improve the consistency of QA pairs, our study uniquely controls the diversity by explicitly controlling KL values.

Despite the potential of data augmentation with QAG to mitigate the sparsity of QA datasets and avoid overfitting, not much is known about the robustness of QA models reinforced with QAG to more challenging test sets. We comprehensively evaluate QAG methods on challenging QA test sets, such as hard questions (Sugawara et al., 2018), implications (Ribeiro et al., 2019), and paraphrased questions (Gan and Ng, 2019).

3.9.6 Variational Autoencoder

The variational autoencoder (VAE) (Kingma and Welling, 2013) is a deep generative model consisting of a neural encoder (inference model) and decoder (generative model). The encoder learns to map from an observed variable to a latent random variable and the decoder works vice versa. The techniques of VAE have been widely applied to NLP tasks such as text generation (Bowman et al., 2016), machine translation (Zhang et al., 2016), and sequence labeling (Chen et al., 2018).

The CVAE is an extension of the VAE, in which the distribution of a latent variable is explicitly conditioned on certain variables and enables generation processes to be more diverse than a VAE (Li et al., 2018b; Zhao et al., 2017b; Shen et al., 2017). The CVAE is trained by maximizing the variational lower bound of the log likelihood.

3.10 Conclusion

We presented a variational QAG model, incorporating two independent latent random variables. We showed that an explicit KL control can enable our model to significantly improve the diversity of QA pairs. The QA pairs generated by our VQAG were shown to be noisy in terms of the grammaticality and answerability of questions, but effective in improving the QA performance in the in- and out-of-distribution test sets. While our synthetic datasets are noisy, they may unintentionally improve the robustness to the noise that can occur in real applications. However, we should pay attention to the negative effect of using our noisy dataset. For example, the lack of the answerability of our synthetic questions may lead to the poor performance in handling unanswerable questions such as SQuAD v2.0.

Moreover, the QAG methods led to improvements in most of the 12 challenge sets while being agnostic to the target distributions during training. We need to pursue such a target-unaware method to improve the robustness of QA models, because it is quite difficult for developers to know the types of questions a QA model cannot handle in advance.

In summary, our experimental results showed that the diversity of QA datasets plays a non-negligible role in improving its robustness, which can be improved with QAG. We will consider using unlabeled documents in other domains to further improve the robustness to other domain corpora in future work.

In addition, we demonstrated that not only QA models but also QG models are biased in terms of the question–context lexical overlap. To determine the influence of the bias, we analyzed the QA performance with data augmentation using the recent QG models. We demonstrated that they frequently degraded the QA performance on questions with low lexical overlap, while improving that on questions with high lexical overlap when using BERT-base. To address this problem, we designed a simple approach using synonym replacement to debias a QA dataset. We demonstrated that the proposed approach improved the QA performance on questions with low lexical overlap while maintaining or slightly degrading the overall scores with only 70k synthetic examples.

Our results suggest that future research in QG for data augmentation should exercise caution to prevent the amplification of dataset bias in terms of lexical overlap. In addition, what features are learned by data augmentation with neural QG models is worth to be explored in more detail to clarify what is improved and what is not improved by QG. It is also worth investigating whether our findings still hold in other QA datasets where annotated questions have lower lexical overlap than those in SQuAD.

Chapter 4

Loss Function Modification for Debiasing Reading Comprehension Models

4.1 Introduction

Pretrained language models (Devlin et al., 2019; Liu et al., 2019b; Lewis et al., 2020; Radford et al., 2019; Brown et al., 2020) have achieved human-level performance on natural language understanding (NLU) tasks, such as question answering (QA) (Rajpurkar et al., 2016), and natural language inference (NLI) (Williams et al., 2018). Additionally, recent studies have shown that pretrained language models capture linguistic features based on an analysis with probing classifiers (Tenney et al., 2019; Hewitt and Manning, 2019; Hewitt and Liang, 2019; Belinkov, 2022).

However, whether language models fine-tuned on NLU tasks have human-like language understanding abilities remains debatable. Because NLU models are prone to use unintended biases in the training set, they have poor generalization ability to out-of-distribution test sets (McCoy et al., 2019; Geirhos et al., 2020), which is a significant challenge in the field. Particularly, QA models trained on intentionally biased training sets are more likely to learn solutions based on spurious correlations rather than causality between inputs and outputs. For example, QA models can learn question–answer type matching heuristics (Lewis and Fan, 2019), and absolute positional heuristics (Ko et al., 2020) especially when a training set is biased towards subsets where corresponding spurious correlations entirely hold. Practically, it is difficult to collect fully unbiased dataset especially when the dataset size is not large. Therefore, it is important to develop a method to learn solutions that do not rely on spurious correlations even when training on unintentionally biased dataset.

To avoid learning spurious correlations in training sets, loss function modification has been extensively studied in this context (Arjovsky et al., 2019; Sagawa* et al., 2020; Clark et al., 2019; Utama et al., 2020a). Such methods have advantages over data-centric approaches in the point that they do not require newly annotated data or data augmentation, so computational costs do not increase significantly. In addition, we need to understand why standard training with standard loss functions leads to shortcut learning through studying new loss functions to mitigate this problem. Pretrained language models are also shown to learn social biases such as gender and race (Kurita et al., 2019; Dev et al., 2020; Kaneko and Bollegala, 2021). We conjecture that, if learning social biases and shortcut solutions arises from standard loss functions, the implications of studying debiasing loss functions are not only improving OOD generalization but also fairer language models. In this chapter, we aim to study the methods for unlearning relative position biases and making models sensitive to large perturbations based on loss function modification.

In extractive QA (Rajpurkar et al., 2016), in which answers to questions are spans in textual contexts, we find that when the relative position of an answer, which is defined

Context	...This changed in **1924** with formal requirements developed for graduate degrees, including offering Doctorate (PhD) degrees...
Question	The granting of Doctorate degrees first occurred in what year at Notre Dame?
Relative Position	-1
Context	...The other magazine, The Juggler, is released **twice** a year and focuses on student literature and artwork...
Question	How often is Notre Dame's the Juggler published?
Relative Position	-2

Table 4.1: Examples taken from SQuAD. Underlined words are contained in both the context and question. **Bold** spans are the answers to the questions. In both the examples, answers are found by *looking to the right* from the overlapping words. See §4.2.1 for the definition of the relative position.

as the relative distance from an answer span to the closest word that appears in both a context and a question, can be exploited as superficial cues by QA models. See Table 4.1 for the examples. Specifically, we find that when the relative positions are intentionally biased in a training set, a QA model tends to degrade the performance on examples where answers are located in relative positions unseen during training. For example, when a QA model is trained on examples where relative positions are negative as shown in Table 4.1, the QA performance on examples with positive relative positions is degraded by 10 $\sim$ 20 points. (See §4.2.1 for the detailed definition of the relative position.) Figure 4.1 shows the performance degradation on examples with unseen relative positions. This can be interpreted as the model preferentially learning to find answers from seen relative positions. The similar phenomena were observed when the distribution of the relative positions in the training set were biased differently as shown in Figure 4.1.

We aim to develop a method for mitigating the performance degradation on subsets with unseen relative positions while maintaining the scores on subsets with seen relative positions, even when the training set is biased with respect to relative positions. To achieve this, we propose debiasing methods based on an ensemble (Hinton, 2002) of an intentionally biased model and a main model. The biased model makes predictions relying on relative positions, which promotes the main model not to rely solely on relative positions. Our experiments on SQuAD (Rajpurkar et al., 2016) using BERT-base (Devlin et al., 2019) as the main model show that the proposed methods achieve accuracies for unseen relative positions are improved by 0$\sim$10 points. We demonstrate that the proposed method is effective in four settings where the training set is differently filtered to be biased with respect to relative positions (§4.4.1). Furthermore, when applied to the full training set, our method improves the generalization to examples where questions and contexts have no lexical overlap (§4.4.2).

Meanwhile, QA models often maintain high accuracy and confidence scores even when the inputs are transformed by large perturbations (e.g., word deletion (Feng et al., 2018; Sugawara et al., 2018), word order shuffling, sentence deletion, and sentence order

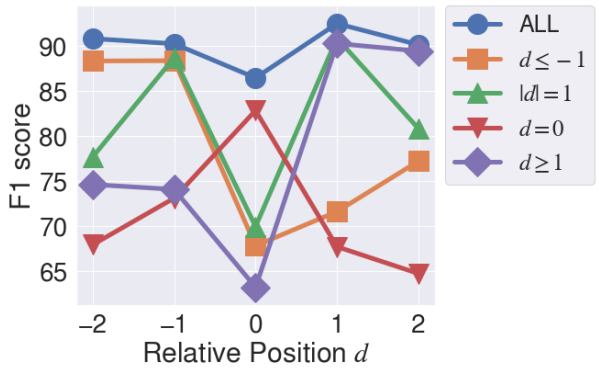


Figure 4.1: F1 score for each relative position d in the SQuAD development set. "ALL" in the legend refers to a QA model trained on all the examples in the SQuAD training set, while the other terms refer to models trained only on examples for which the respective conditionals are satisfied. BERT-base was used for the QA models. The accuracy is comparable to ALL for examples with seen relative positions, but worse for other examples. Please refer to §4.2.1 for the definition of d.

shuffling (Sugawara et al., 2020)). See Table 4.2 for the examples of largely perturbed inputs in extractive QA. Similar phenomena have been observed in NLI (Sinha et al., 2021), and other NLU tasks (Gupta et al., 2021). Our human evaluation (§4.6.2) indicates that humans cannot correctly derive answers from such invalid inputs. We argue that such phenomena imply that models do not adequately use semantic and syntactic features removed by perturbations to make predictions, while these features are indispensable for humans to understand language.

Given the insensitivity to large perturbations (Feng et al., 2018; Sugawara et al., 2018) and the lack of OOD generalization ability (Jia and Liang, 2017; Yogatama et al., 2019; Talmor and Berant, 2019; Sen and Saffari, 2020) of QA models, we also aim to answer the following question: *If the overconfident predictions of QA models for various types of perturbations are penalized, will the OOD generalization and adversarial robustness be improved?* Previous studies have shown that maximizing the entropy (Shannon, 1948) of the output probability for perturbed inputs can successfully reduce model confidence for such perturbed inputs (Feng et al., 2018; Sinha et al., 2021; Gupta et al., 2021). We adopt this method to make QA models sensitive to the perturbations listed in Table 4.3. From the perspective of the representation levels of the construction–integration model (McNamara and Magliano, 2009; Sugawara et al., 2021), which is the most well-formulated model of comprehension in psychology, the word-level perturbations remove features classified as the surface structure, and the sentence-level perturbations remove features classified as the propositional textbase. Thus, we expect that the features removed from inputs by the examined perturbations in Table 4.3 cover the ones necessary for human reading comprehensively.

However, we observe that entropy maximization for a certain perturbation type often fails to transfer to unseen perturbation types. For example, after maximizing the entropy for question deletion, models are not sensitive to function word deletion. To mitigate this lack of transferability, we propose to simultaneously maximize the entropy for the predefined perturbation types. We show that this approach is effective to make models recognize all the predefined perturbations while maintaining in-domain accuracy.

Contrary to our expectations, even though models become sensitive to the four types of perturbations, we find that the OOD generalization or adversarial robustness is not improved. As discussed in Hase et al. (2021), intentionally perturbed inputs become unnatural and are unlikely to appear in a dataset. Therefore, making models sensitive

Perturbed with function word deletion	
Context	...American Football Conference AFC champion **Denver Broncos** defeated National Football Conference NFC champion Carolina Panthers 24 earn third Super Bowl title...
Question	NFL team represented AFC Super Bowl 50?
Perurbed with question deletion	
Context	...The American Football Conference (AFC) champion **Denver Broncos** defeated the National Football Conference (NFC) champion Carolina Panthers 24 to earn their third Super Bowl title...
Question	-
Perturbed with word order shuffling	
Context	...an Carolina the Super 10 American The National third their defeated NFC Conference champion Football to **Denver Broncos** 24 AFC (Panthers (champion...
Question	at represented NFL team the AFC 50 Which Bowl Super?

Table 4.2: Examples of largely perturbed inputs taken from SQuAD. In word order shuffling, we ensure that the answer spans indicated by **bold** remain as they are.

Perturbation σ	Description	Intended feature removed by perturbation
$\mathrm{Del}_{\mathrm{func}}$	Delete all the function words	Function words
$\mathrm{Del}_{\mathrm{que}}$	Delete the question	Question words
$\mathrm{Shuf}_{\mathrm{word}}$	Shuffle the word order in each sentence	Syntactic information
$\mathrm{Shuf}_{\mathrm{sent}}$	Shuffle the sentence order in a context	Discourse relations

Table 4.3: Four types of perturbations σ studied in this work. Different perturbations remove different types of intended features necessary for human reading from the inputs.

to largely perturbed inputs may a have negative impact on out-of-distribution generalization. While becoming sensitive to unnatural inputs with entropy maximization like humans can gain trust from humans, our results suggest that researchers should pay attention to the side effect of entropy maximization.

Our main contributions in this chapter are as follows:

- We find that QA models can exploit relative positional cues as shortcuts when training sets are intentionally biased in terms of relative positions (Figure 4.1).
- We demonstrate that the proposed ensemble-based method is effective in four settings where the training set is differently filtered to be biased with respect to rela-

tive positions (§4.4.1.

- When applied to the full training set, we show that our method improves the generalization to examples where questions and contexts have no lexical overlap (§4.4.2).
- We find that entropy maximization can mitigate the insensitivity to seen perturbation types, but fail to transfer to unseen perturbation types in QA (§4.6.3).
- We show that simply maximizing the entropy for the four perturbation types (§4.5.4), including word- and sentence-level ones, can mitigate this issue (§4.6.3).
- We show that even though QA models become sensitive to the four types of perturbations, the OOD generalization or adversarial robustness is not improved but rather sometimes degraded (§4.6.4).

4.2 Relative Position Bias

4.2.1 Definition

We call a word that is contained in both the question and the context as overlapping word in this study. Then, let d be the relative position of the nearest overlapping word to the answer span in extractive QA. Let w be a word, $c = \{w_i^c\}_{i=0}^N$ for the sentence, $q = \{w_i^q\}_{i=0}^M$ for the question, and $a = \{w_i^c\}_{i=s}^e$ $(0 \leq s \leq e \leq N)$ for the answer, the relative position d is defined as follows:

$$f(j,s,e) = \begin{cases} j-s, & \text{for } j < s \\ 0, & \text{for } s \leq j \leq e \\ j-e, & \text{for } j > e \end{cases} \tag{4.1}$$

$$D = \{f(j,s,e) | w_j^c \in q\} \tag{4.2}$$

$$d = \text{argmin}_{d' \in D} |d'| \tag{4.3}$$

where $0 \leq j \leq N$ denotes the position of the word w_j^c in the sentence, $f(i,s,e)$ denotes the relative position of w_i^c from a, and D denotes the set of relative positions of all overlapping words.[1] Because QA models favor spans that are located close to the overlapping words (Jia and Liang, 2017) and accuracy deteriorates when the absolute distance between the answer span and the overlapping word is large (Sugawara et al., 2018), the one with the lowest absolute value in Equation 4.3 is used as the relative position.[2]

4.2.2 Distribution of Relative Position d

Figure 4.2 shows the distribution of relative position d in the SQuAD (Rajpurkar et al., 2016) training set. This demonstrates that the d values are biased around zero. Although the tendency to bias around zero is consistent for the other QA datasets, there are differences in the distribution between the datasets. See Figure 4.3 for the frequency distributions of relative positions in other datasets. This difference may be caused by the way the datasets were collected or to the domain of the sentences. Therefore, it is necessary to build a QA model that does not overfit to a specific distribution of relative positions.

[1] Because function words as well as content words are important clues for reading comprehension, D in Equation 4.2 can contain function and content words.

[2] There are few cases where d in Equation 4.3 is not fixed to one, but such examples are excluded from the training and evaluation sets for brevity.

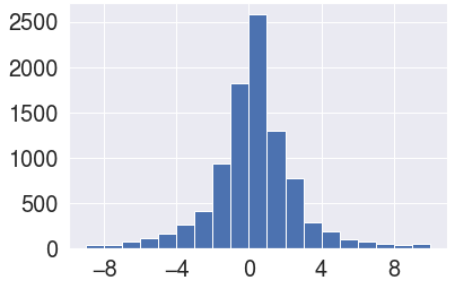


Figure 4.2: Histogram of relative position d in the SQuAD training set.

4.3 Method I: Ensemble-based Debiasing with Biased Models for Relative Position Bias

4.3.1 Debiasing Algorithm

For debiasing algorithms, we employ BiasProduct and LearnedMixin (Clark et al., 2019; He et al., 2019), which are based on product-of-experts (Hinton, 2002). In both methods, after an intentionally biased model is prepared, the loss function is calculated using a product-of-experts mixture of the biased and main models. When training the main model, the biased model is fixed, and the loss function is minimized. Only the main model is used to make predictions during testing. Following existing research (Seo et al., 2017; Devlin et al., 2019), the model $\hat{p}$ outputs the probabilities $\hat{p}(s)$ and $\hat{p}(e)$ of the start position s and the end position e of the answer span, and the loss function is the sum of the cross-entropy of the start and end positions. For simplicity, $\hat{p}(s)$ and $\hat{p}(e)$ will be denoted $\hat{p}$.

BiasProduct

In BiasProduct, the sum of the logarithms of the output probability b of the biased model and the output probability p of the main model is given to the softmax function to obtain $\hat{p}$, as shown below.

$$\hat{p} = \text{softmax}(\log p + \log b) \tag{4.4}$$

This encourages the learning of the main model on examples that do not contain biases that would cause the biased model to make incorrect predictions, rather than on examples that contain biases that would allow the biased model to make correct predictions.

LearnedMixin

BiasProduct strongly depends on the output probability of the biased model. The main model can be made more robust by using LearnedMixin, which predicts whether the main model can trust the prediction of the biased model for each example.

$$\hat{p} = \text{softmax}(\log p + g(c, q) \log b) \tag{4.5}$$

where $g(\geq 0)$ is a learnable function that takes q and c as inputs.

4.3.2 Biased Model

We describe how to construct the biased model described in §4.3.1. The first model is a fully rule-based model with a prior probability of answer span, Answer prior. The second is a question-answer model trained with only absolute position and overlapping vocabulary position as input.

Answer Prior (AnsPrior)

We first use a simple heuristic as a biased model, called AnsPrior. AnsPrior empirically defines the prior probabilities of the start and end positions of the answer span a according to the distribution of the relative position d in a training set. The prior probability b_i that a word w_i^c in a sentence is a start or end of the answer is defined as follows for each of the subsets of the training set that satisfies one of the four conditions shown in the legend of Figure 4.1.

$$b_i = \begin{cases} \mathbb{1}\left[w_{i+1}^c \in q\right]/Z, & \text{for } d \leq -1 \\ \mathbb{1}\left[(w_{i+1}^c \in q) \vee (w_{i-1}^c \in q)\right]/Z, & \text{for } |d| = 1 \\ \mathbb{1}\left[w_i^c \in q\right]/Z, & \text{for } d = 0 \\ \mathbb{1}\left[w_{i-1}^c \in q\right]/Z, & \text{for } d \geq 1 \end{cases} \tag{4.6}$$

where Z denotes a normalizing constant. These prior probabilities are based on a heuristic that assigns equal probabilities to the possible answers in the training set. Therefore, they are inflexible in that they are prior probabilities specific to the distribution of the relative position d of a particular training set.

Position-only model (PosOnly)

We propose PosOnly as a biased model that can be applied to any distributions of relative position d in training sets. PosOnly only accepts as input the sequence of binary variables that indicate if a word in context is overlapped with question. Because the only information available to PosOnly to predict answer spans is the relative distances from the overlapping words, PosOnly is expected to learn solutions using the relative positions regardless of how biased the relative positions of the training set are. The illustration of LearnedMixin with PosOnly is shown in Figure 4.4.

4.4 Experiments I: Mitigating Relative Position Bias

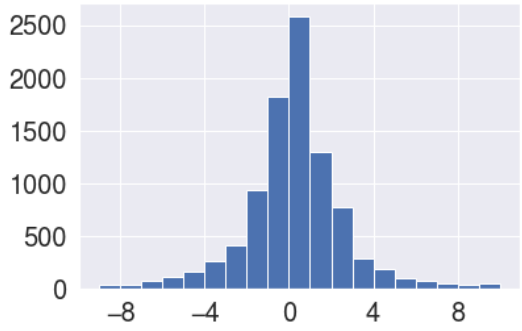
4.4.1 Generalization to Unseen Relative Positions

Dataset

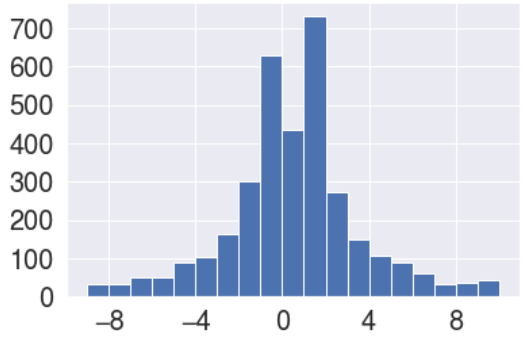
SQuAD 1.1 Rajpurkar et al. (2016) was used as the dataset. To assess the applicability of our methods, we prepare four different ways to make the training set biased. The four subsets were constructed by extracting only examples whose relative positions d satisfied the conditions $d \leq -1$, $|d| = 1$, $d = 0$, and $d \geq 1$, respectively. The sizes are 33,256, 30,003, 21,266, and 25,191, respectively. The scores when trained on the original training set are also reported for comparison. For evaluation, we reported the F1 scores on subsets of the SQuAD development set stratified by the relative positions.

Trainig Details

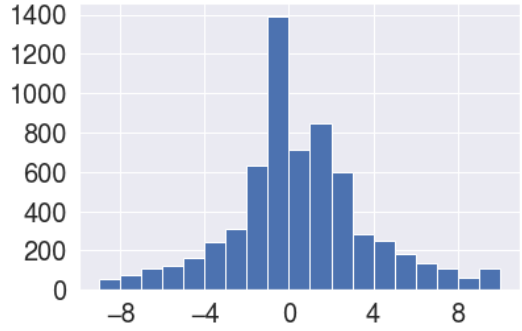
The number of training epochs for each model was 2, the batch size was 32, the learning rate decreased linearly from 3e-5 to 0, and Adam (Kingma and Ba, 2014) was used for optimization.



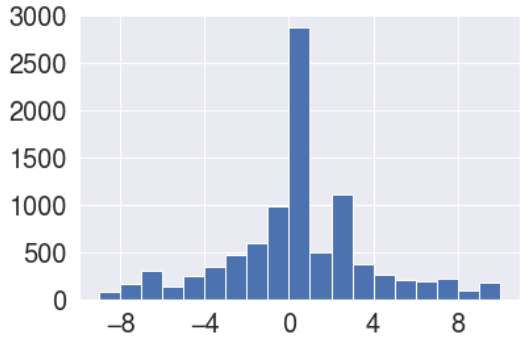
(a) SQuAD



(b) NewsQA



(c) TriviaQA



(d) NaturalQuestions

Figure 4.3: Histograms of relative position d in the SQuAD, NewsQA, TriviaQA, and NaturalQuestions development sets.

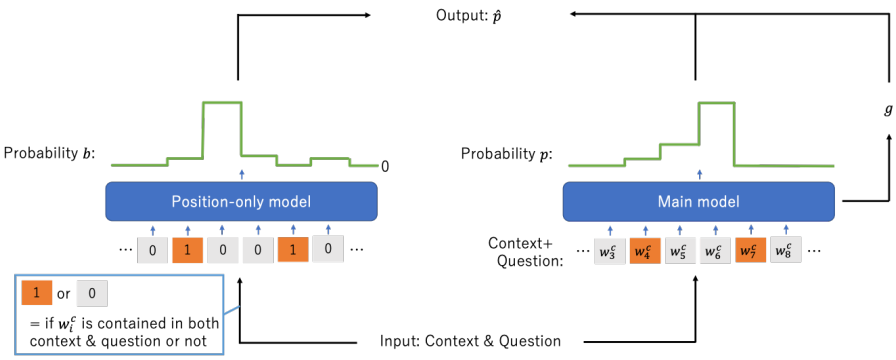


Figure 4.4: Illustration of LearnedMixin with PosOnly as a biased model.

Trained on	Model	Evaluated on						
		$d \leq -3$	$d = -2$	$d = -1$	$d = 0$	$d = 1$	$d = 2$	$d \geq 3$
ALL	BERT-base	82.19	90.82	90.25	86.47	92.49	90.14	81.43
$d \leq -1$	BERT-base	78.17	88.34	88.38	67.82	71.62	77.22	69.54
$d \leq -1$	BiasProduct-AnsPrior	73.00	84.34	85.61	46.32	25.23	64.91	59.06
$d \leq -1$	LearnedMixin-AnsPrior	79.07	89.27	89.01	68.52	72.35	80.43	70.31
$d \leq -1$	BiasProduct-PosOnly	75.04	83.90	83.22	73.80	81.35	81.79	73.27
$d \leq -1$	LearnedMixin-PosOnly	77.00	86.72	86.25	74.26	82.66	82.81	75.94
$\|d\| = 1$	BERT-base	65.62	77.69	88.70	69.96	90.88	80.84	66.42
$\|d\| = 1$	BiasProduct-AnsPrior	60.44	75.07	56.44	49.32	52.37	72.85	57.98
$\|d\| = 1$	LearnedMixin-AnsPrior	73.42	83.39	88.70	74.24	90.47	85.51	73.52
$\|d\| = 1$	BiasProduct-PosOnly	72.41	80.59	84.01	73.34	87.61	83.11	72.09
$\|d\| = 1$	LearnedMixin-PosOnly	73.76	80.63	86.10	74.50	89.64	82.98	72.04
$d = 0$	BERT-base	60.75	67.94	73.11	82.85	67.72	64.74	52.88
$d = 0$	BiasProduct-AnsPrior	56.25	65.15	69.05	81.07	65.10	62.95	49.43
$d = 0$	LearnedMixin-AnsPrior	59.66	69.62	72.53	83.06	68.04	66.03	53.29
$d = 0$	BiasProduct-PosOnly	62.97	67.88	70.22	78.66	66.69	69.12	59.88
$d = 0$	LearnedMixin-PosOnly	65.09	70.47	72.51	81.32	68.29	68.47	59.54
$d \geq 1$	BERT-base	68.03	74.63	74.08	63.21	90.28	89.44	75.42
$d \geq 1$	BiasProduct-AnsPrior	58.63	63.13	29.08	39.22	88.53	88.34	72.29
$d \geq 1$	LearnedMixin-AnsPrior	70.71	77.22	76.82	66.67	90.87	89.75	76.31
$d \geq 1$	BiasProduct-PosOnly	68.54	78.13	78.58	70.72	85.17	81.59	72.90
$d \geq 1$	LearnedMixin-PosOnly	71.17	80.41	79.97	71.33	87.53	84.33	74.24

Table 4.4: F1 scores for each subset of the SQuAD development set. The cells with relative position d seen during training are indicated by gray. In the case of gray cells, the scores tend to remain close to those in the case where the original training set is used (ALL). Conversely, the scores for the other white cells tend to be lower than ALL.

Method

We comparing four combinations of two learning methods for debiasing, BiasProduct and LearnedMixin, and two biased models, AnsPrior and PosOnly. BERT-base (Devlin et al., 2019) was used for both the main model and PosOnly. We also evaluate the BERT-base with standard training as the baseline.

Results

Table 4.4 shows the results. First, when the original training set is used (ALL), the performance exceeds 90 points when $|d| = 1, 2$, while it drops by about eight points when $|d| \geq 3$. Since the distribution of relative position d in the SQuAD training set was biased around zero as shown in the §4.2.2, accuracy may be high for d with high frequency in the training set, and conversely be low for d with low frequency.

The results of the BERT-base baseline trained on subsets where the distribution of relative position d was intentionally biased strengthened the credibility of this hypothesis. For example, when the standard training was performed only on examples with $|d| = 1$ relative positions, the F1 score decreased by less than two points for $|d| = 1$ compared to ALL, whereas the F1 score decreased by 10∼15 points for $|d| \neq 1$. A similar trend was observed in the BERT-base baseline trained on other subsets. This suggests that the model used spurious correlations in the biased subsets to make predictions.

We compare the four proposed debiasing methods under the same conditions of relative positions in the training sets. For the debiasing algorithms, LearnedMixin produced higher F1 scores than BiasProduct in most cases. This result shows the effectiveness of learning the degree to which the predictions of a biased model should be utilized for training the main model. For the biased models, PosOnly was superior to AnsPrior for improving the generalization ability to examples with relative positions unseen during training, i.e., the scores in white cells in Table 4.4. LearnedMixin-PosOnly outperformed LearnedMixin-AnsPrior by about 5 points when trained on $d = 0$ and tested on $d \leq -3$, and when trained on $d \leq -1$ and tested on $d \geq 3$.

In contrast, regarding the scores on examples with relative positions seen during training (i.e., the scores in the gray cells in Table 4.4), LearnedMixin-AnsPrior was superior to LearnedMixin-PosOnly. As pointed out in Utama et al. (2020a), the trade-off between accuracies on in- and out-of-distribution test sets was observed in our cases. Mitigating the trade-off for relative positions is future work.

		Evaluated on	
Trained on	Model	$c \cap q \neq \phi$	$c \cap q = \phi$
ALL	BERT-base	87.94	67.11
ALL	BiasProduct-PosOnly	84.83	59.88
ALL	LearnedMixin-PosOnly	87.37	80.44

Table 4.5: F1 scores for two subsets of the SQuAD development set. Each model is trained on the full SQuAD training set. c indicates the context, and q indicates the question. ϕ indicates the empty set.

4.4.2 Effect of Mitigating Relative Position Bias in Normal Settings

Although we verified the effectiveness of our methods on intentionally biased datasets in §4.4.1, the proposed biased model, PosOnly, can also be applied to training on standard datasets because it does not require prior information about the distribution of relative

positions, unlike AnsPrior. To investigate the effect of our method in a normal setting, we first trained PosOnly on the full training set. We then trained BERT-base as the main model on the full training set using BiasProduct or LearnedMixin with PosOnly as the biased model.

The results are shown in Table 4.5. LearnedMixin-PosOnly improved the generalization to a subset where questions and contexts have no common words (i.e., $c \cap q = \phi$), with little performance degradation on the other subset (i.e., $c \cap q \neq \phi$). This observation implies that our method might mitigate the reliance on overlapping words in a normal setting. However, the size of the subset $c \cap q = \phi$ is only 15, which makes the above conclusion unreliable. Future work should increase the size of this subset and verify its effectiveness.

4.5 Method II: Entropy Maximization for Perturbations

4.5.1 Perturbation Types

We list the examined perturbations in Table 4.3. We adopt two word-level perturbations, function word deletion ($\mathrm{Del}_{\mathrm{func}}$) and word order shuffling ($\mathrm{Shuf}_{\mathrm{word}}$), and two sentence-level perturbations, question deletion ($\mathrm{Del}_{\mathrm{que}}$) and sentence order shuffling ($\mathrm{Shuf}_{\mathrm{sent}}$) to comprehensively assess the sensitivity of QA models to the intended features necessary for humans to understand language, which cover the surface structure and the textbase of the construction–integration model comprehensively. We adopted these perturbations because a QA model is relatively insensitive to them compared to other types of perturbations as found in Sugawara et al. (2020). We expect that entropy maximization with these perturbations make models learn to recognize the intended features as shown in Table 4.3. The detailed motivation of the expectation is described in §4.5.5.

4.5.2 Entropy Maximization

To penalize the confident predictions of models on perturbed inputs, we adopt entropy maximization used by Feng et al. (2018) and Gupta et al. (2021). Namely, we minimize the cross-entropy loss while maximizing the entropy of the output probabilities given perturbed inputs.

When the dataset $\mathcal{D}$ consists of pairs of input x and output y, and the model parameters are θ, the cross-entropy loss is given by

$$\mathcal{L}_{ce} = -\frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \log p_\theta(y|x). \tag{4.7}$$

When a perturbed input x_σ is obtained from x by applying a perturbation σ, the entropy of the model output given perturbed input is given by[3]:

$$H(Y|X_\sigma) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} -p_\theta(y|x_\sigma) \log p_\theta(y|x_\sigma). \tag{4.8}$$

The loss function to be minimized is computed as follows:

$$\mathcal{L} = \mathcal{L}_{ce} - \lambda_\sigma H(Y|X_\sigma). \tag{4.9}$$

where the entropy term is scaled by the factor λ_σ (> 0).

[3]Uppercase letters (e.g., X) represent random variables and lowercase letters (e.g., x) represent actual values.

4.5.3 Conditional Independence Assumption for Extractive QA

In extractive QA tasks, such as SQuAD (Rajpurkar et al., 2016), the models need to specify the start and end positions of the predicted answer span in the context for the given question. When computing the conditional probability $p_\theta(y|x)$ in Equations 4.7 and 4.8, as most existing studies implicitly did during training (Seo et al., 2017; Devlin et al., 2019), we assume that the start and end positions of answers, i.e., Y_{start} and Y_{end}, are conditionally independent given the context and question for brevity. Namely, we assume that $p(Y|X) = p(Y_{start}|X)p(Y_{end}|X)$. Based on this assumption, the entropy term in Equations 4.9 and 4.11 can be computed as follows:

$$H(Y|X_\sigma) = H(Y_{start}|X_\sigma) + H(Y_{end}|X_\sigma). \tag{4.10}$$

We adopt this relaxation because it is costly to raise all the possible answer spans meeting the condition that the start position is lower than or equal to the end position. Our experiments show that this does not degrade the in-distribution accuracy.

4.5.4 Recognizing Multiple Types of Perturbations

Our experiments in §4.6.3 show that maximizing entropy for a certain perturbation type does not transfer to unseen perturbation types. We need to mitigate this problem because our aim is to investigate whether making models sensitive to the four types of perturbations in Table 4.3 improves out-of-distribution generalization.

To mitigate the lack of transferabiliity, we propose to maximize the entropy term for the four type of perturbations to make models recognize those features as follows:

$$\mathcal{L} = \mathcal{L}_{ce} - \sum_{\sigma} \lambda_\sigma H(Y|X_\sigma). \tag{4.11}$$

Model	Perturbation train ↓ / test →	None	Del_{func}	Del_{que}	$\text{Shuf}_{\text{word}}$	$\text{Shuf}_{\text{sent}}$
BERT-base	None	1.38±0.00	3.43±0.16	7.03±1.10	3.53±0.16	1.43±0.00
	Del_{func}	1.37±0.00	**11.9**±0.00	7.1±0.20	10.48±0.36	1.41±0.01
	Del_{que}	1.37±0.01	3.69±0.28	**11.9**±0.00	4.31±0.21	1.41±0.01
	$\text{Shuf}_{\text{word}}$	1.37±0.00	11.87±0.01	7.43±0.27	**11.9**±0.00	1.41±0.00
	$\text{Shuf}_{\text{sent}}$	1.43±0.01	3.65±0.04	7.94±0.35	3.82±0.23	**1.48**±0.01
	ALL	1.41±0.01	**11.9**±0.00	**11.9**±0.00	**11.9**±0.00	1.47±0.01
RoBERTa-base	None	1.13±0.05	3.17±0.85	8.29±0.39	2.76±0.48	1.45±0.02
	Del_{func}	1.10±0.02	**11.9**±0.0	9.05±0.19	11.89±0.00	1.44±0.04
	Del_{que}	1.12±0.03	2.94±0.47	**11.9**±0.00	2.59±0.32	1.42±0.01
	$\text{Shuf}_{\text{word}}$	1.13±0.03	8.95±2.74	8.51±0.31	**11.9**±0.00	2.38±0.28
	$\text{Shuf}_{\text{sent}}$	1.09±0.01	2.27±0.59	9.08±0.23	**11.9**±0.00	**11.9**±0.00
	ALL	1.14±0.04	**11.9**±0.00	**11.9**±0.00	**11.9**±0.00	**11.9**±0.00

Table 4.6: Entropy of the model predictions on the original and perturbed SQuAD 1.1 development set. The more confident predictions models make, the lower entropy is.

Model	Perturbation train ↓ / test →	None	Del_{func}	Del_{que}	$\text{Shuf}_{\text{word}}$	$\text{Shuf}_{\text{sent}}$
BERT-base	None	88.0±0.03	54.2±0.06	10.2±0.41	26.5±0.14	83.9±0.06
	Del_{func}	88.1±0.02	**22.2**±3.83	10.2±0.28	24.2±0.73	83.8±0.26
	Del_{que}	88.1±0.12	53.9±0.91	**5.9**±0.74	26.4±0.36	84.1±0.14
	$\text{Shuf}_{\text{word}}$	88.1±0.07	36.4±0.32	10.0±0.37	**16.2**±1.53	83.8±0.25
	$\text{Shuf}_{\text{sent}}$	88.0±0.09	54.3±0.79	9.9±0.53	26.8±0.29	83.9±0.18
	ALL	88.0±0.10	31.1±2.61	7.9±1.81	19.1±0.41	83.9±0.14
RoBERTa-base	None	91.2±0.04	61.0±0.72	11.3±0.33	29.3±0.06	87.3±0.21
	Del_{func}	91.4±0.01	**14.5**±2.21	11.0±0.21	19.2±0.88	87.4±0.12
	Del_{que}	91.2±0.13	60.9±0.53	**7.0**±2.44	28.9±0.41	87.5±0.12
	$\text{Shuf}_{\text{word}}$	91.2±0.17	47.8±4.34	11.2±0.12	12.1±2.05	86.8±0.30
	$\text{Shuf}_{\text{sent}}$	91.3±0.05	59.9±0.50	10.2±0.70	10.0±1.87	**17.0**±5.06
	ALL	91.3±0.08	19.6±3.74	8.9±2.46	**9.7**±1.56	34.8±7.65
Human Score		91.2†	28.1	0.1	10.8	53.2

Table 4.7: F1 scores on the original and perturbed SQuAD dev set. See Table 4.3 for details of perturbation types. †Copied from the SQuAD 1.1 Leaderboard.

4.5.5 Interpretation from the Perspective of Causality

When the maximum predicted probability (confidence score) of model θ for original input x is $p_\theta(\hat{y}|x) = \max p_\theta(y|x)$, the difference in probabilities that the model assigns to $\hat{y}$,

$$d_\theta(x, \hat{y}, \sigma) = p(\hat{y}|x) - p(\hat{y}|x_\sigma), \tag{4.12}$$

can be regarded as quantifying how much feature s is used by the model for making prediction $\hat{y}$. Quantities of similar definitions have been used as feature importance (Li et al., 2016c; DeYoung et al., 2020; Hase et al., 2021) to increase the interpretability of the model, or the degree to which a cause affects an outcome in the context of causality (Pearl, 2000).

By minimizing the cross entropy while maximizing the entropy in Equation 4.9, $p(\hat{y}|x)$ is increased while $p(\hat{y}|x_\sigma)$ is decreased. Thus, minimizing $\mathcal{L}$ in Equation 4.9 is expected to indirectly increase $d_\theta(x, \hat{y}, \sigma)$ in Equation 4.12. When $d_\theta(x, \hat{y}, \sigma)$ is larger than non-zero values, features in x removed by perturbation σ have causal effects on prediction $\hat{y}$ made by QA model θ. Based on this interpretation, we assume that training with entropy maximization causes QA models to use intended features as listed in Table 4.3, and have positive impact on out-of-distribution generalization.

However, our experiments show that the results are opposite to the assumption. We will discuss why the out-of-generalization is not improved with the approach in §4.6.4.

4.6 Experiments II: Sensitivity to Multiple Types of Perturbations and Out-of-Distribution Generalization

In this study, we considered a QA task because the inputs of QA datasets consist of questions and contexts, and the contexts often consist of multiple sentences. This enables examination of broader types of perturbations, such as sentence order shuffling, that are absent in other NLU tasks such as NLI and paraphrase identification (Dolan and Brockett, 2005), where the inputs are only two sentences.

4.6.1 Experimental Setups

Model We used BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019b) for QA models because they are often adopted in QA.

Dataset We used SQuAD 1.1 (Rajpurkar et al., 2016) for training and evaluation. To evaluate the OOD generalization, we employed the dev set of NewsQA (Trischler et al., 2017), TriviaQA (Joshi et al., 2017), SearchQA (Dunn et al., 2017), HotpotQA (Yang et al., 2018), and NaturalQuestions (Kwiatkowski et al., 2019) from MRQA 2019 shared task (Fisch et al., 2019). To evaluate adversarial robustness, we used AddSent and AddOneSent from Adversarial SQuAD (Jia and Liang, 2017).

Training Details We used the Adam (Kingma and Ba, 2014) optimizer with epsilon 1e-8. The models were trained for two epochs with the learning rate being linearly decreased from 3e-5 to zero. The batch size was set to 32. For other hyperparameters, we generally used the default hyperparameters in the example code provided by Huggingface. We tuned the scaling factor λ_σ in Equation 4.9 in $\{0.01, 0.1, 1.0, 5.0\}$ for each perturbation σ on the SQuAD dev set based on the F1 scores. When computing the loss in Equation 4.9, we sampled exactly one perturbed example from each original example in the training set. The means and standard deviations of the F1 scores over three random seeds are reported.

Model	Perturbation	SearchQA	HotpotQA	NQ	NewsQA	TriviaQA
BERT-base	None	27.3±0.60	60.6±0.44	59.1±0.50	55.8±0.26	58.5±0.27
	Del_{func}	27.2±0.98	60.0±0.37	56.2±0.39	55.9±0.37	58.6±0.19
	Del_{que}	27.4±0.71	60.0±0.21	58.7±0.27	55.5±0.43	58.6±0.46
	$\text{Shuf}_{\text{word}}$	27.8±0.29	60.1±0.02	56.7±0.42	55.9±0.52	58.7±0.16
	$\text{Shuf}_{\text{sent}}$	27.6±1.83	60.2±0.20	58.8±0.45	56.0±0.15	58.6±0.08
	ALL	28.0±1.08	60.7±0.45	56.9±0.81	55.3±0.44	57.9±0.49
RoBERTa-base	None	30.7±1.90	66.5±0.73	61.8±0.39	64.3±0.11	62.7±0.30
	Del_{func}	26.8±2.49	66.6±0.25	61.4±0.22	64.5±0.21	62.1±0.58
	Del_{que}	31.5±0.99	66.2±0.31	62.0±0.45	64.7±0.34	62.8±0.36
	$\text{Shuf}_{\text{word}}$	23.6±1.65	66.7±0.48	57.0±2.39	64.6±0.07	62.2±0.37
	$\text{Shuf}_{\text{sent}}$	28.6±1.03	66.2±0.42	17.5±3.90	64.7±0.27	61.4±0.43
	ALL	14.4±3.51	66.5±0.81	25.3±4.56	63.6±0.24	60.6±0.38

Table 4.8: F1 scores on out-of-domain test sets. The means±standard deviations over three random seeds are reported.

Model	Perturbation	AddSent	AddOneSent
BERT-base	None	50.8±0.40	62.1±0.89
	Del_{func}	49.6±0.54	61.4±1.26
	Del_{que}	50.7±0.88	62.2±0.39
	$\text{Shuf}_{\text{word}}$	49.9±0.63	61.8±1.14
	$\text{Shuf}_{\text{sent}}$	49.9±0.87	61.6±1.08
	ALL	50.7±0.71	62.2±0.74
RoBERTa-base	None	62.6±0.90	72.0±0.95
	Del_{func}	62.4±1.00	71.6±1.33
	Del_{que}	61.9±0.94	71.6±0.84
	$\text{Shuf}_{\text{word}}$	61.5±0.37	70.8±0.65
	$\text{Shuf}_{\text{sent}}$	62.2±1.61	71.6±1.10
	ALL	61.9±1.11	71.4±0.41

Table 4.9: F1 scores on adversarial test sets. The means±standard deviations over three random seeds are reported.

4.6.2 Human Evaluation

To see whether humans can derive correct answers from inputs with the examined perturbations, we conducted a human evaluation. We asked human annotators to answer a question by extracting an answer span from a given context. The annotators are allowed to submit empty answers when they cannot find plausible answers. The input is transformed by one of the four perturbation types. We randomly chose 200 examples for each perturbation from the SQuAD dev set. Three annotators on Amazon Mechanical Turk were assigned to each example. Annotators were paid 0.04$ per example.

4.6.3 Cross-Perturbation Evaluation

Previous studies have shown that maximizing entropy for input with certain perturbations, such as word deletion (Feng et al., 2018) and word order shuffling (Sinha et al., 2021), can make models sensitive (i.e., less confident) to the *same* type of perturbations at test time. Intuitively, the word orders and words themselves should convey different types of information. Then, we can ask the question: Can maximizing the entropy for word order shuffling transfer to that for word deletion?

To answer this question, we investigated the transferability of entropy maximization across four types of perturbations. That is, we trained the QA models with entropy maximization for one of the four perturbations and evaluated the models with seen and unseen perturbations. To evaluate the sensitivity of the predictions to the perturbations, we used the entropy of the predicted answers and the F1 score.[4] The entropies and F1 scores for the in-distribution dev set are shown in Tables 4.6 and 4.7, respectively.

QA Models Are More Insensitive to Large Perturbations Than Humans

First, without entropy maximization, models can somehow correctly answer decent portions of perturbed inputs compared with humans. Notably, QA models trained without entropy maximization were most robust to $\mathrm{Shuf}_{\mathrm{sent}}$ among the four perturbation types. This result is consistent with the findings of Sugawara et al. (2020). While the F1 scores decreased substantially and the entropy is relatively high when the inputs in the dev set are perturbed with $\mathrm{Del}_{\mathrm{que}}$, the scores of the models are still higher than the human score.

These relatively high F1 scores imply that models can not recognize the intended features removed by the perturbations as humans do. Moreover, the confident predictions of the models on invalid data may harm the reliability of model predictions in real-world applications.

Entropy Maximization Can Penalize Confident and Correct Predictions for Seen Perturbations

Entropy maximization make models sensitive to seen perturbation types, as shown by the diagonals of Tables 4.6 and 4.7, except for BERT-base with $\mathrm{Shuf}_{\mathrm{sent}}$. The F1 scores decreased and the entropies increased along the diagonal cells. Moreover, the entropies for the seen perturbations almost reached the maximum value of 11.9 without hurting the F1 scores on the original dev set (None).

Entropy Maximization Fails to Transfer to Unseen Perturbations

However, maximizing entropy for a certain perturbation type often cannot transfer to unseen perturbation types. For example, maximizing entropy for $\mathrm{Del}_{\mathrm{que}}$ have little impact on the sensitiveness of models to $\mathrm{Del}_{\mathrm{func}}$. To mitigate the lack of cross-perturbation transferability, we simply maximize the entropy terms for all the perturbation types as in Equation 4.11, which is denoted as ALL. We show that this approach can successfully make models less confident on all the four perturbations except for BERT-base with $\mathrm{Shuf}_{\mathrm{sent}}$. On the other hand, we observed that there are some perturbation types where entropy maximization can transfer to some extent (e.g., from $\mathrm{Del}_{\mathrm{func}}$ to $\mathrm{Shuf}_{\mathrm{word}}$ and from $\mathrm{Shuf}_{\mathrm{sent}}$ to $\mathrm{Shuf}_{\mathrm{word}}$).

Influence of the Scaling Factor

Among the perturbation types, BERT-base failed to become less confident on $\mathrm{Shuf}_{\mathrm{sent}}$. To determine the cause, we examined the influence of scaling factors. First, the chosen scaling factor was 0.01, which was insufficient to increase the entropy. Second, we found that the F1 scores of BERT-base on the clean dev set and the dev set perturbed with $\mathrm{Shuf}_{\mathrm{sent}}$ are strongly correlated. This implies that BERT-base cannot distinguish the original sentence order and the shuffled sentence order inherently. Given that this

[4] In this experiment, because the maximum length of the context is set to 384, the maximum of the entropy is 11.9 ($= -1/384 * \log(1/384) * 384 * 2$), and the minimum is 0.0 based on Equations 4.8 and 4.10.

tendency is not consistent with RoBERTa-base, the next sentence prediction task for pretraining BERT-base may cause this difference.

4.6.4 Out-of-Distribution Generalization

If models can recognize the intended features described in Table 4.3 after entropy maximization, it is possible that the way models process language becomes closer to the way humans do, and thereby generalize to OOD test sets (Talmor and Berant, 2019; Jia and Liang, 2017) better than before.

The F1 scores on OOD test sets are shown in Tables 4.8 and 4.9, respectively. Based on these results, entropy maximization for any perturbations did not improve the OOD generalization nor the adversarial robustness, but rather sometimes degraded them.

As discussed in Hase et al. (2021), intentionally perturbed inputs can be out-of-distribution for models, and do not naturally appear in a dataset. Therefore, regularizing the predictions on largely perturbed inputs may not have a positive effect on the generalization to natural examples. Making QA models recognize natural changes in inputs using carefully designed perturbations is future work.

4.7 Related Work

4.7.1 Bias in QA and Debiasing Loss Functions

Recent studies have pointed out that natural language understanding models tend to learn shortcut solutions specific to the same distribution of the training set. The problem was pointed out that in natural language inference, models use the spurious correlation regarding lexical items (Gururangan et al., 2018) and lexical overlap (McCoy et al., 2019). In extractive QA, existing studies have shown that substantial number of questions can be answered only by type matching between questions and answers (Weissenborn et al., 2017), or looking at few tokens in questions (Sugawara et al., 2018), and that models can easily learn the absolute position bias of the answers (Ko et al., 2020). One of the most effective ways to assess whether a model is learning human-like reading skills is to use data with different distributions at training and test time. In this study, we analyzed this problem from a new perspective of relative position, and developed an effective debiasing method. The loss function we used is the same as Ko et al. (2020), and the biased model for debiasing the relative position bias was newly proposed in this study.

4.7.2 Insensitivity to Large Perturbations

Recently, there has been a surge of interest in the insensitivity of NLU models to large perturbations (Sugawara et al., 2018, 2020; Hessel and Schofield, 2021). These studies showed that NLU models can correctly predict the output even when the inputs is largely perturbed with some transformations such as word deletion and word order shuffling at test time. To penalize the insensitivity of NLU models, entropy maximization has been used (Feng et al., 2018; Gupta et al., 2021; Sinha et al., 2021). However, the effect of entropy maximization for large perturbations has been studied in isolation. Given the hypothesis that different perturbation removes different features from input as shown in Table 4.3, only regularizing entropy for single type of perturbation may not be enough to make models' predictions more human-like.

Making dialog models recognize dialog history with perturbations has been shown to improve the performance of dialog systems (Zhou et al., 2021). This work is most relevant to our motivation.

4.7.3 Sensitivity to Small Perturbations

In contrast, deep learning models can misclassify slightly perturbed inputs (Szegedy et al., 2013; Jia and Liang, 2017; Mudrakarta et al., 2018), which are called adversarial examples. To mitigate this issue, adversarial training and its variants have been proven to be effective (Goodfellow et al., 2015; Zhu et al., 2020; Jiang et al., 2020; Liu et al., 2020b) in maintaining the model prediction when the input is slightly perturbed, which is the exact opposite of entropy maximization used in our work. Wang et al. (2021b) have studied the connection between these two contrasting phenomena.

4.8 Conclusion

We showed that models tend to exploit relative position biases in extractive question answering, causing the performance degradation for relative positions unseen during training. To mitigate this problem, we proposed effective debiasing method. When whole training sets are used, we showed that out method improved the model robustness to examples where questions and answers have no lexical overlap. Future work includes extracting subsets from training sets that makes it easier for QA models to learn the relative position bias, mitigating the trade-off between the accuracies for seen and unseen relative positions, and analyzing the intermediate representations of QA models.

We also showed that entropy maximization often fails to transfer to unseen perturbations. Maximizing the entropy terms for various types of perturbations is effective in mitigating this problem. The failure of entropy maximization to improve out-of-distribution generalization may be caused by the unnaturalness of the perturbed inputs. Modifying the perturbation functions to effectively improve out-of-distribution generalization is future work.

Chapter 5

Investigating the Learnability of Shortcut Solutions in Machine Reading Comprehension

5.1 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 et al., 2019; Gardner et al., 2020).

To date, question answering (QA) models for reading comprehension have been reported to learn several types of shortcut solutions, degrading the OOD generalization (Jia and Liang, 2017; Sugawara et al., 2018; Ko et al., 2020). The high performance of QA models with partial inputs also indicates learning shortcut solutions (Sugawara et al., 2020; Yu et al., 2020). Various approaches have been proposed to mitigate these problems in QA, such as data augmentation methods (Shinoda et al., 2021b) and debiasing loss functions (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 5.1. 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

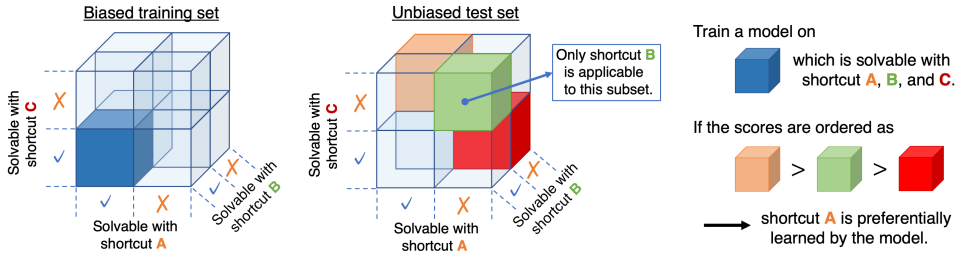


Figure 5.1: An illustration of the behavioral test to reveal which shortcut solution QA models prefer to learn.

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 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.

Our contributions in this chapter are as follows.

- We designed and conducted a behavioral analysis to reveal which shortcut solution QA models prefer to learn. For extractive QA, shortcut solutions based on answer positions are more preferred to the word and type matching shortcuts. For multiple-choice QA, shortcut solutions based on word-label correlations are more preferred to the lexical overlap shortcut. (5.3.2)

- We found that more preferred shortcut solutions tend to lie in flatter and deeper loss surfaces in the parameter space. The orders of the flatness and depth of the loss surfaces are roughly correlated with the preferential order of learning shortcuts. (5.3.3)

- We designed and conducted an information-theoretic analysis (Rissanen Shortcut Analysis; RSA) to quantitatively compare the learnabilities of shortcut solutions. We found that The availability of the preferred shortcuts (based on answer positions and word-label correlations) tends to make the task easier to learn. (5.3.4)

- We showed that the requirements of the proportion of examples where shortcut solutions are not available are correlated with the learnabilies of the shortcut solutions. (5.3.5)

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

5.2 Shortcut Solutions

5.2.1 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.

5.2.2 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. ($\mathrm{k} = \mathrm{Word}$)

Type Matching Matching question and answer types (Weissenborn et al., 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 (Honnibal et al., 2020) for named entity recognition. ($\mathrm{k} = \mathrm{Type}$)

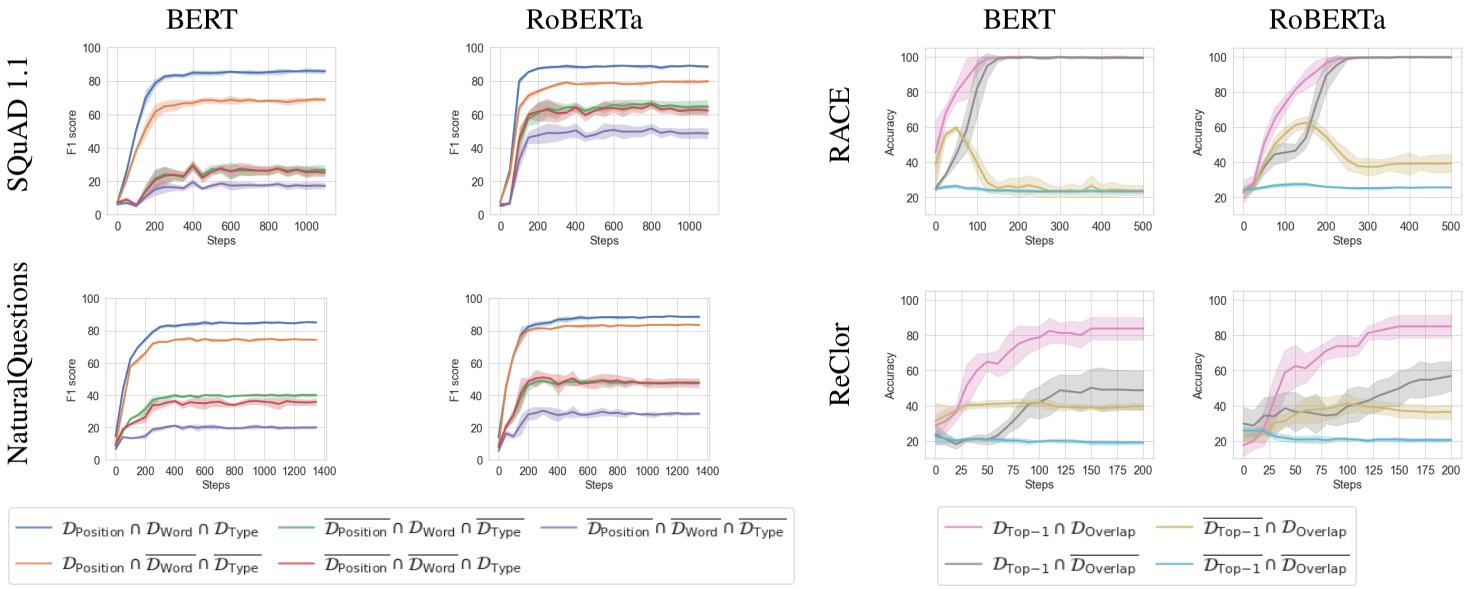


Figure 5.2: Left: F1 score on each subset of the SQuAD 1.1 and NaturalQuestions evaluation sets during training. Right: Accuracy on each subset of the RACE and ReClor test sets during training. The mean±standard deviations over 5 random seeds are displayed.

5.2.3 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 et al., 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., 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

$$z^* = \frac{p(y|w)}{\sqrt{p_0(1-p_0)/n}}, \tag{5.1}$$

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 5.1. We choose the top-1 word for the analysis of the word-label correlation shortcut for simplicity. ($\mathrm{k} = \mathrm{Top\text{-}1}$)

Lexical Overlap NLI models exploit the lexical overlap between premise and hypothesis to make predictions (McCoy et al., 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 uni-grams contained in both sequences to the number of words in an option. ($\mathrm{k} = \mathrm{Overlap}$)

5.3 Experiments

5.3.1 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 5.1. 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., 2019b) 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.[1]

Evaluation Metrics

For extractive QA, we used the F1 score as the evaluation metric, where as for multiple-choice QA, we used accuracy.

[1]The codes are publicly available here `https://github.com/KazutoshiShinoda/ShortcutLearnability`.

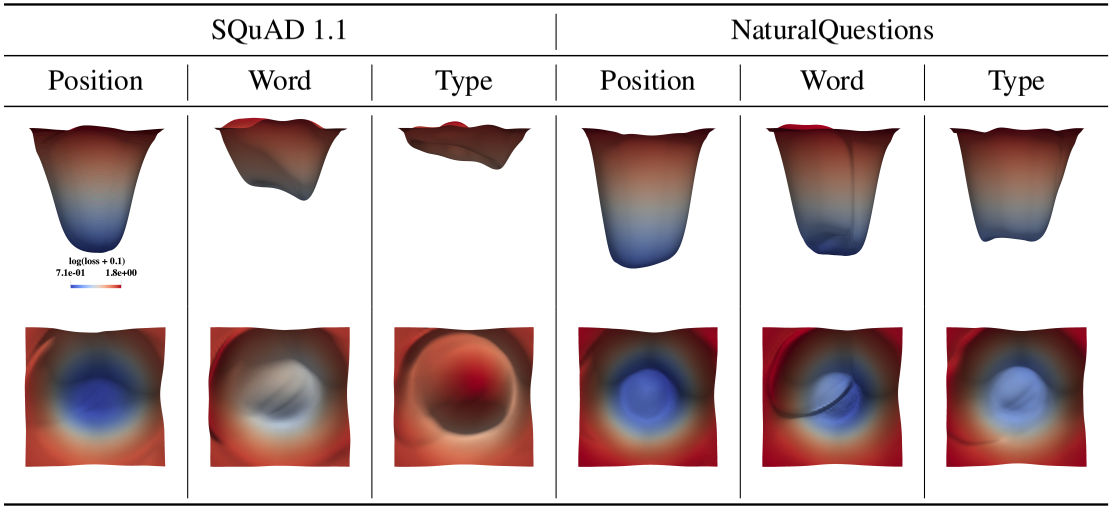


Figure 5.3: Visualization of loss landscapes around each shortcut in extractive QA datasets. The x and y directions are randomly selected in the parameter space. The center of the surface corresponds to the model that uses each shortcut.

5.3.2 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 5.1.

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 5.2 (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 et al., 2020b; 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 5.2 (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

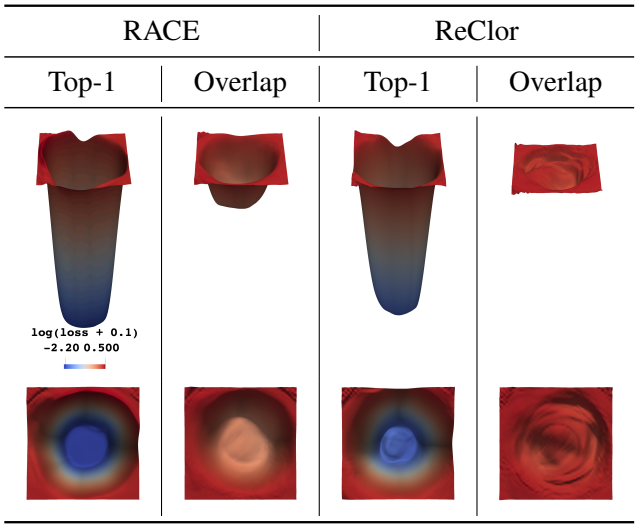


Figure 5.4: Visualization of loss landscapes around each shortcut in multiple-choice QA datasets.

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).

5.3.3 Visualizing the Loss Landscape

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

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. (2018a).

Results The visualization results for extractive and multiple-choice QA are displayed in Figures 5.3 and 5.4. 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.[2] 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

[2]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, 1997a).

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

Table 5.2: Minimum description lengths (kbits) on biased datasets where only one of the examined shortcut solutions is valid. The means$\pm$standard deviations over five random seeds are reported.

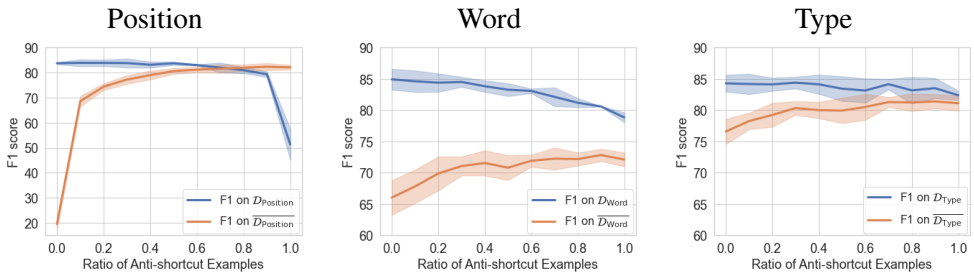


Figure 5.5: F1 scores on shortcut and anti-shortcut examples from SQuAD with different proportions of anti-shortcut examples in the training set, with the size set to 5k. The mean$\pm$standard deviations over 5 random seeds are displayed.

$\mathcal{D}_{\text{Position}} \cap \mathcal{D}_{\text{Word}} \cap \mathcal{D}_{\text{Type}}$ and $\mathcal{D}_{\text{Top-1}} \cap \mathcal{D}_{\text{Overlap}}$ learned to use the answer-position and word-label correlation shortcuts, respectively.

5.3.4 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.

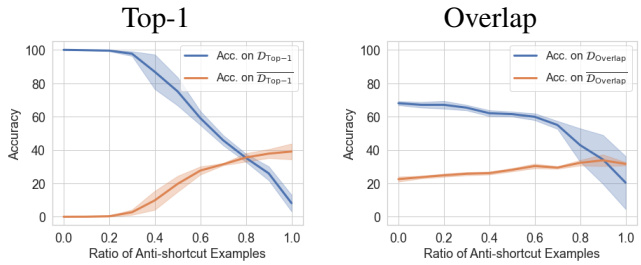


Figure 5.6: Accuracies on shortcut and anti-shortcut examples from RACE with different proportions of anti-shortcut examples in the training set, with the size set to 4k. The mean±standard deviations over 5 random seeds are displayed.

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 $\{t_1, t_2, ..., t_S\}$[3], the MDL is estimated with the online code as follows:

$$L = \sum_{i=0}^{S-1} \sum_{n=t_i+1}^{t_{i+1}} -\log_2 p_{\theta_i}(y_n|x_n), \tag{5.2}$$

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 et al. (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 5.2. Note that the MDLs cannot be compared across datasets because the MDLs are dependent on the dataset size t_S as shown in Eq. 5.2. 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 5.2. 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 5.2 and visualization in Figures 5.3 and 5.4.

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 5.2, the MDLs may also reflect the generalization capability of models as well as the characteristics of shortcuts.

5.3.5 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?

[3]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).

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., 2017a; Shinoda et al., 2021b) 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 5.5 and 5.6 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 et al. (2019) to further mitigate the gap.

5.4 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 answered 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 et al., 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 et al. (2021a) found that automatic 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. (2020b) used virtual adversarial training to improve the robustness to adversarial attacks. Wang et al. (2021a) 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.

5.5 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 tend 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 the shortcut.

Chapter 6

Model Architecture Modification for Debiasing Vision-and-Language Models

6.1 Introduction

To operate robots in human spaces, instruction following tasks in 3D environments have attracted substantial attention (Anderson et al., 2018; Chen et al., 2019; Puig et al., 2018). In these tasks, robots are required to translate natural language instructions and egocentric vision into sequences of actions. To enable robots to perform further complex tasks that require interaction with objects in 3D environments, the "interactive instruction following" task has been proposed (Shridhar et al., 2020). Here, interaction with objects refers to the movement or the state change of objects caused by actions such as picking up or cutting.

In interactive instruction following, agents need to be robust to variations of objects and language instructions that are not seen during training. For example, as shown in Figure 6.1, objects are of the same class but vary in attributes such as color, shape, and texture. Also, as shown in Figure 6.2, language instructions vary in predicates, referring expressions pointing to objects, and the presence or absence of modifiers, even though their intents are the same.

However, our analysis revealed that the end-to-end neural baseline proposed by Shridhar et al. (2020) for the task is not robust to variations of objects and language instructions, i.e., it often fails to interact with objects of unseen attributes or to take the correct actions consistently when language instructions are replaced by their paraphrases. Similar phenomena have been observed in the existing studies. For example, end-to-end neural models that compute outputs from vision or language inputs with only continuous representations in the process are shown to be sensitive to small perturbations in inputs in image classification (Szegedy et al., 2013) and natural language understanding (Jia and Liang, 2017).

Given these observations, we hypothesize that reasoning over the high-level symbolic representations of objects and language instructions are robust to small changes

Figure 6.1: An example of four different apples that an agent is required to pick up, taken from the ALFRED benchmark (Shridhar et al., 2020). An agent is required to interact with objects of various shapes, colors, and textures.

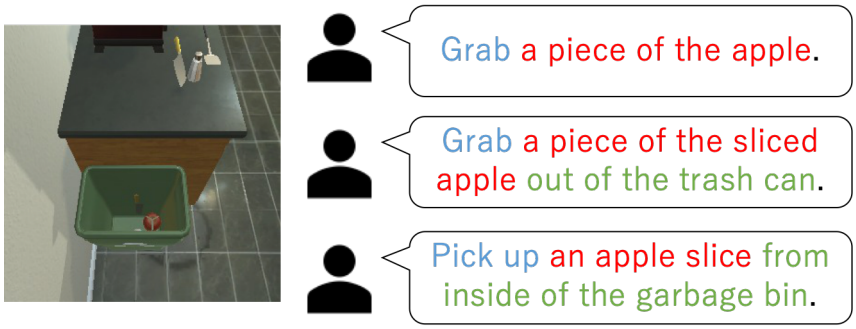


Figure 6.2: An example where different language instructions are given by different annotators to the same action, taken from the ALFRED benchmark (Shridhar et al., 2020). Predicates (blue), referring expressions (red), and modifiers (green) have the same meaning but can be expressed in various ways. Modifiers can be omitted. Agents should take the correct action consistently no matter how the given instruction is expressed.

in inputs. In this study, we aim to mitigate this problem by utilizing high-level symbolic representations that can be extracted from raw inputs and reasoning over them. Specifically, high-level symbolic representations in this study refer to classes of objects, high-level actions, and their arguments of language instructions. These symbolic representations are expected to be robust to small changes in the input because of their discrete nature.

Our contributions in this chapter are as follows.

- We propose Neuro-Symbolic Instruction Follower (NS-IF), which introduces high-level symbolic feature extraction and reasoning modules to improve the robustness to variations of objects and language instructions for the interactive instruction following task (§6.2).
- In subtasks requiring interaction with objects, our NS-IF significantly outperforms an existing end-to-end neural model, S2S+PM, in the success rate while improving the robustness to the variations of vision and language inputs (§6.3 and §6.4).

6.2 Method: Neuro-Symbolic Instruction Follower

We propose Neuro-Symbolic Instruction Follower (NS-IF) to improve the robustness to variations of objects and language instructions as illustrated in Figures 6.1 and 6.2. The whole picture of the proposed method is shown in Figure 6.3. Specifically, different from the S2S+PM baseline (Shridhar et al., 2020), we introduce semantic understanding module (§6.2.4) and MaskRCNN (§6.2.5) to extract high-level symbolic features from raw inputs, subtask updater (§6.2.6) to make the model recognize which subtask is being solved, and object selector (§6.2.8) to make robust reasoning over the extracted symbolic features. Other components are adopted following S2S+PM. Each component of NS-IF is explained below in detail.

6.2.1 Notation

The length of the sequence of actions required to accomplish a task is T. The action at time t is a_t. The observed image at time t is v_t. The total number of subtasks is N. The step-by-step language instruction for the n-th subtask is l_n, and the language instruction indicating the goal of the overall task is g. Let b_n be the high-level action for the language instruction l_n for each subtask, and r_n be its argument. The total number

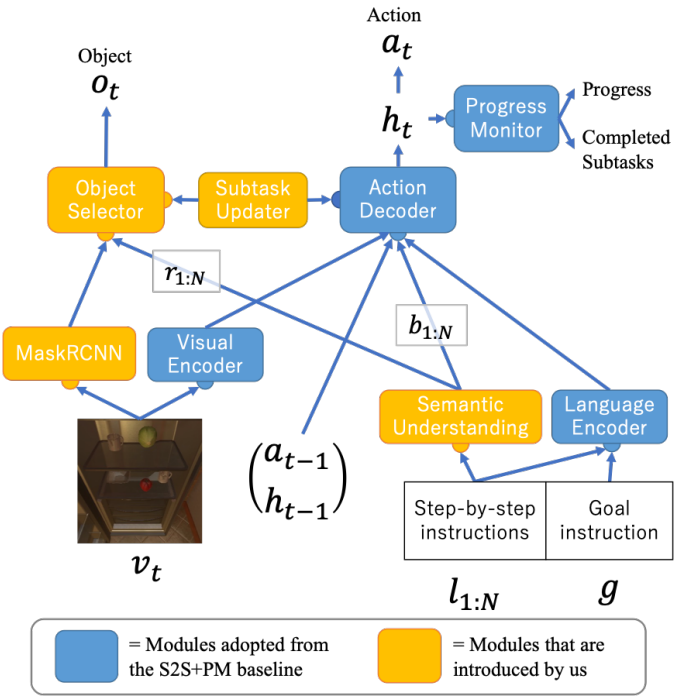


Figure 6.3: Overview of the proposed NS-IF. The modules are colored to clarify the difference between the S2S+PM baseline (Shridhar et al., 2020) and our NS-IF.

of observable objects in v_t is M. The mask of the m-th object is u_m, and the class of the m-th object is c_m. An example is displayed in Figure 6.4.

6.2.2 Language Encoder

The high-level symbolic representations of step-by-step language instructions consist of only the high-level actions $b_{1:N}$ and the arguments $r_{1:N}$, and information about mod-

Goal Instruction g: Put the chilled slice apple in the trash bin

n	**Step-by-step instructions** l_n	**High-level action** b_n	**Argument** r_n
0	Turn right then head to the counter beside the microwave	GotoLocation	countertop
1	Pick up the knife on the counter	PickupObject	knife
2	Turn left then head to the sink	GotoLocation	apple
3	Slice the apple in the sink	SliceObject	apple
…	…	…	…
N	Put the slice apple in the trash bin	PutObject	garbagecan

(a) Instructions and their high-level actions and arguments

n	0	0	…	0	1	2	…	2	3	4	…
t	0	1	…	10	11	12	…	16	17	18	…
Action a_t	Look Down	Rotate Right	…	Move Ahead	Pickup Object	Rotate Left	…	Look Down	Slice Object	Look Down	…
Object o_t	-	-	…	-	Knife	-	…	-	Apple	…	…
Vision v_t			…				…				…

(b) Visual inputs and ground-truth actions and objects for each time step

Figure 6.4: An example of the interactive instruction following task taken from ALFRED.

ifiers is lost. To avoid the failure caused by the lack of information, we input all the words in the language instructions to the language encoder to obtain continuous representations. The word embeddings of the language instruction g representing the goal and the step-by-step language instruction $l_{1:N}$ for all subtasks are concatenated and inputted into bidirectional LSTM (Hochreiter and Schmidhuber, 1997b) (BiLSTM) to obtain a continuous representation H of the language instruction.[1]

6.2.3 Visual Encoder

Similarly, for the image v_t, a continuous representation V_t is obtained with ResNet-18 (He et al., 2016), whose parameters are fixed during training.

6.2.4 Semantic Understanding

Here, we convert the language instructions l_n for each subtask into high-level actions b_n and their arguments r_n. To this end, we trained RoBERTa-base (Liu et al., 2019b) on the ALFRED training set. We adopted RoBERTa-base here because it excels BERT-base (Devlin et al., 2019) in natural language understanding tasks (Liu et al., 2019b). For predicting b_n and r_n from l_n, two classification heads are added in parallel on top of the last layer of RoBERTa-base.

We used the ground truth b_n and r_n provided by ALFRED during training. At test time, we used b_n and r_n predicted by the RoBERTa-base. To see the impact of the prediction error of semantic understanding, we also report the results when using the ground truth b_n and r_n at test time.

6.2.5 MaskRCNN

MaskRCNN (He et al., 2017) is used to obtain the masks $u_{1:M}$ and classes $c_{1:M}$ of each object from the image v_t. Here, we use a MaskRCNN pre-trained on ALFRED.[2]

6.2.6 Subtask Updater

We find that the distribution of the output action sequences varies greatly depending on which subtask is being performed. In this section, to make it easier to learn the distribution of the action sequences, the subtask s_t being performed is predicted at each time. Since our aim is to evaluate the approach on each subtask, we conducted experiments under the condition that the ground truth s_t is given during both training and testing.

6.2.7 Action Decoder

The action decoder predicts the action a_t at each time using LSTM. Different from S2S+PM, the action decoder takes high-level actions $b_{1:N}$ as inputs. Namely, the inputs are the hidden state vector h_{t-1} at time $t-1$, the embedding vector of the previous action a_{t-1}, the embedding representation of the high-level action $E(b_{1:N})^T p(s_t)$ and V_t at time t obtained using the embedding layer E and s_t, and the output x_{t-1} from h_{t-1} to H. V_t, and w_t, which is the concatenation of the output x_t of attention from h_{t-1} to H. Then, after concatenating w_t to the output h_t of LSTM, we obtain the distribution of behavior a_t via linear layer and Softmax function.

[1] When using only high-level symbolic expressions as input to the BiLSTM, the accuracy decreased. Therefore, we use continuous representation as input here.

[2] https://github.com/alfworld/alfworld

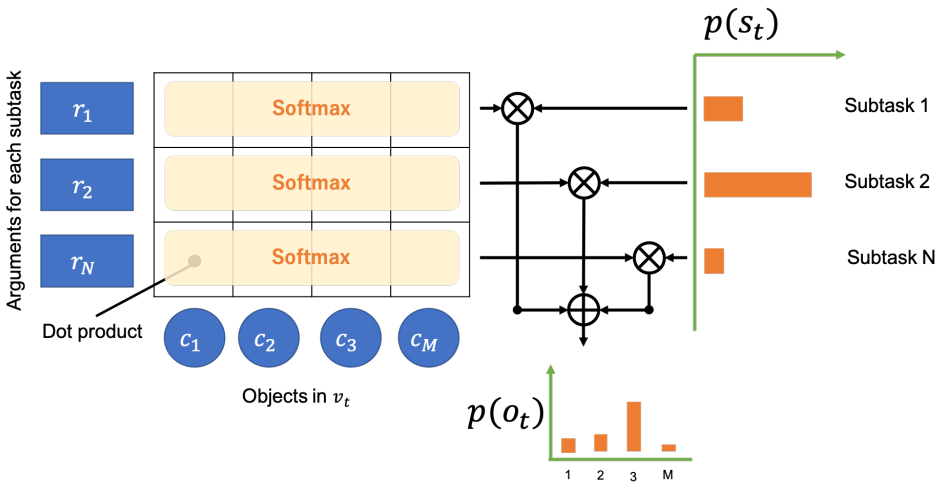


Figure 6.5: Detailed illustration of the object selector (§6.2.8).

6.2.8 Object Selector

When the action a_t is an interaction action such as Pickup or Slice, models need to select the object with a mask. The object selector module outputs the mask of an selected object detected by MaskRCNN as follows:

$$p(o_t) = \sum_n p(s_t = n)\text{Softmax}(E(c_{1:M})E(r_n)^T) \tag{6.1}$$

$$m^* = \text{argmax}_{o_t} p(o_t). \tag{6.2}$$

Then, the model outputs the mask u_{m^*}. The overview of the object selector is shown in Figure 6.5.

6.2.9 Progress Monitor

Following Shridhar et al. (2020), our model learns the auxiliary task with the Progress Monitor, which monitors the progress of the task. Specifically, from h_t and w_t, we obtain normalized progress (t/T) and completed subtasks (number of accomplished subtasks divided by N) through independent linear layers.

6.3 Experiments

6.3.1 Dataset

We used the ALFRED dataset, in which roughly three annotators provided different language instructions for the final objective and each subtask for each demonstration played by skilled users of AI2-Thor (Kolve et al., 2017). ALFRED also provides the Planning Domain Definition Language (McDermott et al., 1998), which contains the high-level actions and their arguments. They are used to define the subtasks when creating the dataset. In this study, we defined high-level actions and their arguments as the output of semantic understanding. The number of training sets is 21,023. Since the test sets are not publicly available, we use the 820 validation sets for rooms that are seen during training, and the 821 validation sets for rooms that are not seen during training. Note that the object to be selected in the validation set is an object that has never been seen during training, regardless of rooms. Therefore, models need to be robust to unseen objects in both the validation sets.

6.3.2 Training Details

For NS-IF, we followed the hyperparameters proposed by Shridhar et al. (2020). For RoBERTa-base, we used the implementation and default hyperparameters provided by Huggingface (Wolf et al., 2019). The hyperparameters for training NS-IF and RoBERTa-base are summarized in Table 6.1.

Hyperparamete	**NS-IF**	**RoBERTa-base**
Dropout	0.3	0.1
Hidden size (encoder)	100	768
Hidden size (decoder)	512	-
Warmup ratio	0.0	0.1
Optimizer	Adam (Kingma and Ba, 2014)	AdamW (Loshchilov and Hutter, 2019)
Learning rate	1e-4	5e-5
Epoch	20	5
Batch Size	8	32
Adam ϵ	1e-8	1e-8
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Gradient Clipping	0.0	1.0

Table 6.1: Hyperparameters for training NS-IF and RoBERTa-base.

6.3.3 Main Results

In this study, we evaluated the performance on each subtask, which is appropriate to assess the robustness to variations of objects and instructions in detail. The baseline models are SEQ2SEQ+PM (Shridhar et al., 2020), which uses only continuous representations in the computation process at each time, and MOCA (Pratap Singh et al., 2021), which factorizes the task into perception and policy. Note that both the baselines are end-to-end neural models unlike our NS-IF.

	Model	Goto	Pickup	Slice	Toggle
Seen	S2S+PM (Shridhar et al., 2020)	- (51)	- (32)	- (25)	- (100)
	S2S+PM (Reproduced)	55 (46)	37 (32)	20 (15)	**100** (100)
	MOCA (Pratap Singh et al., 2021)	**67** (54)	**64** (54)	67 (50)	95 (93)
	NS-IF	43 (37)	**64** (58)	**71** (57)	83 (83)
	NS-IF (Oracle)	43 (37)	69 (63)	73 (59)	100 (100)
Unseen	S2S+PM (Shridhar et al., 2020)	- (22)	- (21)	- (12)	- (32)
	S2S+PM (Reproduced)	26 (15)	14 (11)	3 (3)	34 (28)
	MOCA (Pratap Singh et al., 2021)	**50** (32)	**60** (44)	68 (44)	11 (10)
	NS-IF	32 (19)	**60** (49)	**77** (66)	**43** (43)
	NS-IF (Oracle)	32 (19)	65 (53)	78 (53)	49 (49)

Table 6.2: Success rate (%) for each subtask in seen and unseen environments. The scores that take into account the number of actions required for success are given in parentheses. Higher is better. The best success rates among the models without oracle are **boldfaced**. The best success rates among all the models are underlined.

	High-level Action				Argument			
	Goto	Pickup	Slice	Toggle	Goto	Pickup	Slice	Toggle
Seen	99.40	99.01	91.39	97.87	71.58	89.68	92.72	76.60
Unseen	99.22	98.84	97.14	99.42	73.73	89.78	96.19	64.74

Table 6.3: Accuracy (%) of semantic understanding (i.e., high-level action and argument prediction) for each subtask in seen and unseen environments.

We report the results in Table 6.2. The proposed NS-IF model improves the success rate especially in the tasks requiring object selection, such as Pickup, Slice and Toggle. Notably, NS-IF improved the score on Slice in the Unseen environments from 3% to 77% compared to S2S+PM, and surpass MOCA. The fact that only objects of unseen attributes need to be selected to accomplish the tasks in the test sets indicates that the proposed method is more robust to variations of objects on these subtasks than the baselines.

	Model	Goto	Pickup	Slice	Toggle
Seen	S2S+PM (Reproduced)	315 / 240 / 239	105 / 52 / 202	7 / 5 / 29	**29** / **0** / **0**
	MOCA (Pratap Singh et al., 2021)	**386** / 281 / **131**	184 / 90 / **86**	24 / 8 / **9**	26 / 2 / 1
	NS-IF	243 / **204** / 349	**215** / **37** / 107	**29** / **3** / **9**	17 / 12 / **0**
	NS-IF (Oracle)	250 / 178 / 368	253 / 9 / 97	32 / 0 / 9	29 / 0 / 0
Unseen	S2S+PM (Reproduced)	147 / **99** / 513	42 / **21** / 281	1 / **0** / 31	13 / 10 / 30
	MOCA (Pratap Singh et al., 2021)	**216** / 307 / **233**	155 / 84 / **103**	18 / 6 / 7	4 / **6** / 44
	NS-IF	168 / 145 / 441	**182** / 36 / 122	**24** / 1 / **6**	**19** / 9 / **25**
	NS-IF (Oracle)	165 / 89 / 502	218 / 12 / 113	25 / 0 / 7	28 / 0 / 25
					(I) ↑ / (II) ↓ / (III) ↓

Table 6.4: Three kinds of scores, (I), (II), and (III), that reflect the robustness to variations of language instructions in the subtask evaluation. These scores indicate the number of unique demonstrations where a model (I) succeeds with all the language instructions, (II) succeeds with at least one language instruction but fails with other paraphrased language instructions, or (III) fails with all the language instructions. Higher is better for (I), and lower is better for (II) and (III). The best scores among the upper three models are **boldfaced**.

On the other hand, the S2S+PM model fails in many cases and does not generalize to unknown objects. Moreover, the accuracy of S2S+PM is much lower in Unseen rooms than in Seen ones, which indicates that S2S+PM is less robust not only to unknown objects but also to the surrounding room environment. By contrast, the difference in accuracy of NS-IF between Seen and Unseen is small, indicating that the proposed model is relatively robust to unknown rooms. This may be related to the fact that the output of ResNet is sensitive to the scenery of the room, while the output of MaskRCNN is not. The failed cases of NS-IF in Pickup and Slice are caused by the failure to predict the action a_t, or failure to find the object in drawers or refrigerators after opening them.

There are still some shortcomings in the proposed model. There was little improvement in the Goto subtask. It may be necessary to predict the bird's eye view from the first person perspective, or the destination based on the objects that are visible at each time step. In addition, the accuracy of other subtasks (PutObject, etc.) that require specifying the location of the object has not yet been improved. This is because the pre-trained MaskRCNN used in this study has not been trained to detect the location of the object.

6.3.4 Semantic Understanding Performance

To investigate the cause of the performance gap between NS-IF and its oracle, we evaluated the performance of the semantic understanding module for each subtask. The results are given by Table 6.3.

The accuracies of high-level action prediction are $91 \sim 99$ %. Whereas, the accuracies of argument prediction are $64 \sim 96\%$. This may be because the number of classes of arguments are 81, while that of high-level actions is eight.

For the Toggle subtask, the accuracy of argument prediction is lower than 80%. This error might primarily cause the drop of the success rate in Toggle in Table 6.2 from NS-IF to NS-IF (Oracle). Thus, improving the accuracy of argument prediction would close the gap.

In contrast, despite of the error of argument prediction in Goto as seen in Table 6.3, the success rates of NS-IF and its oracle in Table 6.2 were almost the same. This observation implies that our NS-IF failed to fully utilize the arguments to perform the Goto subtasks. Mitigating this failure is future work.

6.4 Analysis: Evaluating the Robustness to Variations of Language Instructions

The robustness of models to variations of language instructions can be evaluated by seeing whether the predictions remains correct even if the given language instructions are replaced by paraphrases (e.g., Figure 6.2) under the same conditions of the other variables such as the room environment and the action sequence to accomplish the task.

The results are shown in Table 6.3.3. The reported scores show that the proposed model increased the overall accuracy while improving the robustness to variations of language instructions compared to S2S+PM. The numbers of demonstrations corresponding to (I), "succeeds with all the language instructions", for NS-IF were superior to the baselines for Pickup, Slice, and Toggle in unseen environments, which indicates that NS-IF is the most robust to paraphrased language instructions. Using oracle information further increased the robustness.

The cases that fall into the category (III), "fails with all the language instructions", are considered to result from causes unrelated to the lack of the robustness to variations of language instructions. These failures are, for example, caused by the failure to select an object in a drawer or a refrigerator after opening them.

6.5 Related Work

6.5.1 Neuro-Symbolic Method

In the visual question answering (VQA) task, Yi et al. (2018) proposed neural-symbolic VQA, where the answer is obtained by executing a set of programs obtained by semantic parsing from the question against a structural symbolic representation obtained from the image using MaskRCNN (He et al., 2017). Reasoning on a symbolic space has several advantages such as (1) allowing more complex reasoning, (2) better data and memory efficiency, and (3) more transparency, making the machine's decisions easier for humans to interpret. In the VQA task, several similar methods have been proposed. Neuro-Symbolic Concept Learner (Mao et al., 2019) uses unsupervised learning to extract the representation of each object from the image and analyze the semantics of the questions. Neural State Machine (Hudson and Manning, 2019) predicts a scene graph including not only the attributes of each object but also the relationships between objects to enable more complex reasoning on the image. However, they are different from our study in that they all deal with static images and the final output is only the answer. Neuro-symbolic methods were also applied to the video question answering task, where a video, rather than a static image, is used as input to answer the question (Yi* et al., 2020). However, here too, the final output is only the answer to the question.

6.5.2 Embodied Vision-and-Language Task

Tasks that require an agent to move or perform other actions in an environment using visual and language information as input have attracted much attention in recent years. In the room-to-room dataset (Anderson et al., 2018), a Vision-and-Language Navigation task was proposed to follow language instructions to reach a destination. In both the embodied question answering (Das et al., 2018) and interactive question answering (Gordon et al., 2018) tasks, agents need to obtain information and answer questions through movement in the environment, and the success or failure is determined by only the final output answer. In contrast to these tasks, ALFRED (Shridhar et al., 2020) aims to accomplish a task that involves moving, manipulating objects, and changing states of objects in 3D environments.

6.6 Conclusion

In this study, we proposed a neuro-symbolic method to improve the robustness to variations of objects and language instructions for interactive instruction following. In addition, we introduced the subtask updater that allows the model to recognize which subtask it is solving at each time step. Our experiments showed that the proposed method significantly improved the success rate in the subtask requiring object selection, while the error propagated from the semantic understanding module degraded the performance. The experimental results suggest that the proposed model is robust to a wide variety of objects. Furthermore, the analysis showed that the robustness to variations of language instructions was improved by our model.

ALFRED contains the ground truth output of semantic understanding and the prior knowledge of which subtask was being solved at each step, so it was possible to use them in training. It should be noted that the cost of annotations of them can not be ignored for other datasets or tasks. If the cost is impractical, it may be possible to solve the problem by unsupervised learning, as in NS-CL (Mao et al., 2019). Whereas, for training MaskRCNN, annotation is not necessary because the mask and class information of the object can be easily obtained from AI2-Thor. Therefore, whether annotation of mask

and class is necessary or not depends on how well an object detection model trained on artificial data obtained from simulated environments generalizes to real world data. Future work includes learning subtack updater to enable evaluation on the whole task.

Chapter 7

Discussion

In this section, we provide possible directions of future work based on the implications of our thesis. Firstly, we discuss the diversity of training sets and how to improve it in §7.1. Secondly, we discuss the possibility that prior knowledge about NLU improves the generalization of NLU models in §7.2. Lastly, we discuss combining different types of debiasing methods to further improve the generalization in §7.3.

7.1 Improving the Diversity of Training Examples

As shown in Chapter 3, improving the diversity of training examples is effective to improve the generalization to OOD test sets. Moreover, the relative position bias described in Chapter 4 suggests that relative positions in training sets should be diverse enough; otherwise models' generalization is degraded. As shown in Chapter 5, when absolute positions of answers lack diversity or some words are highly correlated with labels, data augmentation methods specialized for these features may be useful for debiasing QA models.

However, data augmentation based on generative models have a drawback. We proposed a data augmentation method based on variational auto-encoder to improve the diversity of questions and answers. As we showed in Chapter 3, generative models can be biased towards generating examples of some features such as high lexical overlap, degrading the diversity unintentionally. When augmenting training sets with trained generative models, biases of generative models should be carefully investigated. In addition, as shown in Chapter 5, data-centric methods alone may not be sufficient to unlearn less learnable shortcut solutions such as lexical overlap.

On the other hand, what kind of features in inputs and outputs should be diverse is still remained unclear. As discussed in Schwartz and Stanovsky (2022), biased distributions of features may be effective to teach models world knowledge or common sense that are not clearly stated in text. (e.g., when humans are given presents from others, they usually become usually glad.) From this perspective, we suggest that improving diversity by removing shortcut features such as word-label correlation and lexical overlap is at least important for OOD generalization.

7.2 Injecting Prior Knowledge about Tasks into Models

Besides increasing the size of training sets, we claim that prior knowledge about NLU tasks is useful to make NLU models learn more human-like solutions. We aimed to verify this claim in Chapter 4, but OOD generalization was not improved contrary to our expectation. However, the causal interpretation of perturbations in Chapter 4 suggests that carefully designed perturbations based on the requisite skills for NLU could make models recognize such small but important changes in inputs. As discussed in Chapter

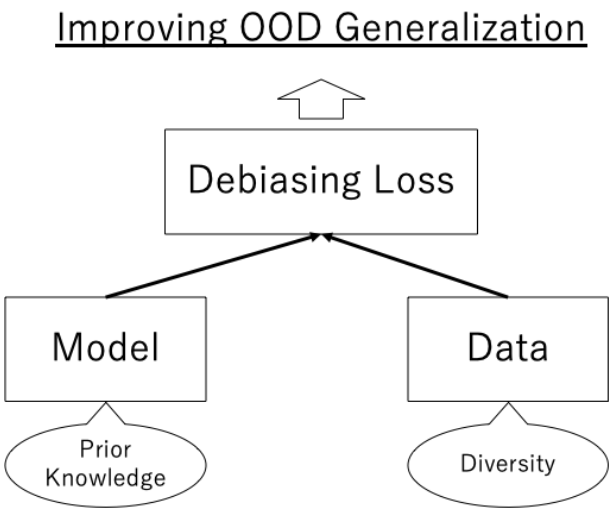


Figure 7.1: Illustration of possible directions of future work for improving OOD generalization by mitigating shortcut learning.

4, designing perturbations that transform text while maintaining naturalness of text is future work.

In Chapter 6, we exploited prior knowledge about the vison-and-language task via symbolic rule-based functions and high-level features of inputs. The proposed method was generally successful, while limiting the applicability to other tasks. At least, we can say that models may exploit unintended features in datasets as shortcuts without those knowledge. This can be avoided by explicitly giving models such rules and features that are indispensable for human-like solutions.

Injecting prior knowledge through data has been recently studied in computer vision (Wen et al., 2022) in NLU (Chen et al., 2022). They specified where "key features" (e.g., regions in images, or textual spans), which are necessary for humans to solve tasks, are in whole inputs. Then models are made use those key features, thereby improving the generalization.

7.3 Combining Different Types of Debiasing Methods

In most existing studies, only one of the three categories, data-, loss- and model-centric approaches, has been applied to shortcut mitigation independently. Moreover, the mitigation selection did not depend on the characteristics of shortcuts to be mitigated. As shown in Chapter 5, different shortcut solutions in QA have different learnabilities, and the learnabilities are roughly correlated to the requirements of data balancing for shortcut mitigation. In addition, we found that less learnable shortcuts can not be unlearned with only data balancing. This finding suggests that debiasing methods should be chosen depending on the learnability of shortcut to be mitigated. For example, less learnable shortcut solutions may be mitigated by combining data augmentation, loss function modification, and model architecture modification. Therefore, combining data-, loss-, or model-centric debiasing methods based on the characteristics of shortcut solutions may mitigate shortcut learning effectively.

Wu et al. (2022) proposed to combine data augmentation and debiasing loss funcitions. They found that both the two methods contributed to improve the accuracy. In addition, Shrestha et al. (2022) found that modifying a model architecture and a loss function can be combined to further improve the accuracy. This suggests that different types of debiasing methods have orthogonal effects on OOD generalization.

To better choose and combine the types of mitigation methods, we claim that it is also important to understand the pros and cons of each type of methods. In this thesis, we comprehensively studied the different types of debiasing methods to understand the difference. For example, we showed that bias-aware data augmentation and loss function modification methods can induce trade-offs between in- and out-of-distribution accuracies in Chapter 3 and 4. For data augmentation methods, we found that the order in which synthetic and original data is given to models can affect the performance of models in Chapter 3. However the understandings about the difference are still limited. We expect that more studies in this direction will be done in future work.

Chapter 8

Conclusions

In this thesis, we tackle the shortcut learning problem of NLU models from four perspectives; data augmentation, modifying loss functions, mitigation method selection, and model architectures.

As for data augmentation for QA, question-answer pair generation (QAG) methods based on generative models have been studied to mitigate the scarcity of QA training sets. Previous QAG methods can enhance the IID generalization of QA models by mainly improving the quality of generated question-answer pairs. However, the diversity of generated question-answer pairs has not been received much attention by the community. In addition, QAG methods have not been applied to OOD settings. Thus, we aim to answer the following question; does improving the diversity of generated question-answer pairs lead to enhanced OOD generalization? To answer the question, we propose a generative model based on a conditional variational auto-encoder that prioritizes diversity of generated question-answer pairs. We show that our models is less likely to degrade OOD accuracy compared to existing methods on 12 OOD test sets, while improving IID generalization. However, our analysis reveal that QAG methods based on generative models, which are trained on the same training set as a QA model, amplify dataset bias in terms of question-context lexical overlap; that is, they tend to generate questions with high lexical overlap. Moreover, we find that they often degrade the QA performance on questions with low lexical overlap when used for data augmentation. To alleviate this issue, we use a simple data augmentation approach based on synonym replacement to reduce the lexical overlap. Our method is effective to improve the robustness of QA models to questions with low lexical overlap, but degrades the QA performance on questions with high lexical overlap.

As described above, data augmentation methods can improve the accuracy on examples similar to generated examples, but we find that data augmentation has the side effect of unintentionally amplifying dataset bias. To avoid such problems, we also study modifying loss functions to control the training process, which has the potential to compensate for the drawbacks of data augmentation. However, previous methods for modifying loss functions rely on prior information about dataset bias, limiting their applicability to real-world applications. To overcome this issue, we focus on the two directions: uncovering unknown bias, and devising a method that does not rely on bias information. Specifically, we first discover that a QA model can learn relative answer positions as shortcut cues, resulting in the lack of robustness to unseen relative positions. The relative positions in this study is defined as the relative distance of the most closest overlapping words from answers. We propose a product-of-expert based algorithm with a newly proposed technique for learning the relative position bias. We show that the proposed method can significantly mitigate the performance drop on examples with unseen relative positions. However, as the existing methods, our method also suffer from the trade-off between the accuracies for seen and unseen relative positions. Second, we aim

to answer the research question; if QA models are made sensitive to largely perturbed questions and contexts like humans, do they acquire the human-like OOD generalization ability? We study four types of perturbations that remove indispensable features that are necessary for understanding word- and sentence-level meanings and order. This method is potentially promising in that it does not require prior information about dataset biases but just knowledge about human reading comprehension. Contrary to our expectation, the proposed method can not improve the OOD generalization, which may be due to the unnaturalness of those perturbed examples.

From the perspective of mitigation method selection, we focus on the limitation that previous shortcut mitigation methods do not fully utilize the characteristics of shortcut solutions. We hypothesize that the learnability of shortcut solutions, i.e., how easy it is to learn a shortcut solution, is useful to constructing a new dataset.

Empirically, we show that when the bias is known, models can be made debiased successfully. However, prior knowledge about bias in a training set is not always easily accessible. Therefore, designing methods that is applicable even when the bias is unknown, such as our debiasing methods based on generative models and perturbations, has to be studied in parallel.

In summary, we find that 1) data augmentation based on generative models can unintentionally amplify dataset biases, 2) data augmentation based on synonym replacement for mitigating question-context lexical overlap bias may introduce a trade-off between ID and OOD accuracy, 3) regularizing model behavior on perturbed inputs does not improve OOD accuracy but has a potential impact, and 4) regularizing model training based on product-of-experts can mitigate the relative position bias. Mitigating the trade-off and explaining the difference between data augmentation and regularization methods is future work.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1203. URL https://aclanthology.org/D16-1203.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4971–4980, 2018.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1620. URL https://www.aclweb.org/anthology/P19-1620.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, June 2018.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv e-prints*, art. arXiv:1907.02893, July 2019.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL https://www.aclweb.org/anthology/2020.acl-main.421.

Mana Ashida and Saku Sugawara. Possible stories: Evaluating situated commonsense reasoning under multiple possible scenarios. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3606–3630, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.319.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. URL https://arxiv.org/abs/1409.0473.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 528–539. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bahng20a.html.

Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1672–1682. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1142.

Yujia Bao, Shiyu Chang, and Regina Barzilay. Predict then interpolate: A simple algorithm to learn stable classifiers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 640–650. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/bao21a.html.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. Beat the AI: Investigating Adversarial Human Annotation for Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678, 11 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00338. URL https://doi.org/10.1162/tacl_a_00338.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.696. URL https://aclanthology.org/2021.emnlp-main.696.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://www.aclweb.org/anthology/2020.acl-main.463.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.703. URL https://aclanthology.org/2020.emnlp-main.703.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://www.aclweb.org/anthology/K16-1002.

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.113. URL https://aclanthology.org/2021.emnlp-main.113.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/bras20a.html.

Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2683–2688, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1625275.1625707.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in $beta$-VAE. *arXiv preprint arXiv:1804.03599*, 2018. URL https://arxiv.org/abs/1804.03599.

Remi Cadene, Corentin Dancette, Hedi Ben-Younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing Unimodal Biases for Visual Question Answering. In *Advances in Neural Information Processing Systems 32*, 2019. URL https://arxiv.org/abs/1906.10169.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.278. URL https://aclanthology.org/2022.naacl-main.278.

Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference*

on Empirical Methods in Natural Language Processing, pages 215–226, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1020.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 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)*, pages 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1418. URL https://www.aclweb.org/anthology/D19-1418.

Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1554–1563, 2021. doi: 10.1109/ICCV48922.2021.00160.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR*, June 2018.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1606. URL https://www.aclweb.org/anthology/D19-1606.

Sunipa Dev, Tao Li, Jeff M. Phillips, and Vivek Srikumar. On Measuring and Mitigating Biased Inferences of Word Embeddings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7659–7666, April 2020. doi: 10.1609/aaai.v34i05.6267. URL https://ojs.aaai.org/index.php/AAAI/article/view/6267.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2092. URL https://www.aclweb.org/anthology/N18-2092.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, pages 13042–13054. Curran Associates, Inc., 2019.

Yana Dranker, He He, and Yonatan Belinkov. IRM—when it works and when it doesn't: A test case of natural language inference. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=KtvHbjCF4v.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.71. URL https://aclanthology.org/2021.naacl-main.71.

Xinya Du and Claire Cardie. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1177. URL https://aclanthology.org/P18-1177.

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1123. URL https://aclanthology.org/P17-1123.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.253. URL https://aclanthology.org/2020.acl-main.253.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1407. URL https://aclanthology.org/D18-1407.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*,

pages 968–988, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.84. URL https://aclanthology.org/2021.findings-acl.84.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL https://aclanthology.org/D19-5801.

Wee Chung Gan and Hwee Tou Ng. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1610. URL https://www.aclweb.org/anthology/P19-1610.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.117. URL https://www.aclweb.org/anthology/2020.findings-emnlp.117.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.135. URL https://aclanthology.org/2021.emnlp-main.135.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1107. URL https://www.aclweb.org/anthology/D19-1107.

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online, August 2021. Association for Computational Linguistics. doi: 10.

18653/v1/2021.findings-acl.168. URL https://aclanthology.org/2021.findings-acl.168.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *CVPR*, 2018.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL https://www.aclweb.org/anthology/P16-1154.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1014. URL http://aclweb.org/anthology/P16-1014.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. Bert & family eat word salad: Experiments with text understanding. *arXiv preprint arXiv:2101.03453*, 2021.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://www.aclweb.org/anthology/N18-2017.

Peter Hase, Harry Xie, and Mohit Bansal. Search methods for sufficient, socially-aligned feature importance explanations with in-distribution counterfactuals. *arXiv preprint arXiv:2106.00786*, 2021.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6115. URL https://www.aclweb.org/anthology/D19-6115.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. doi: 10.1109/CVPR.2016.90.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, Oct 2017.

Michael Heilman and Noah A. Smith. Good question! Statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics, 2010. URL http://aclweb.org/anthology/N10-1086.

Jack Hessel and Alexandra Schofield. How effective is BERT without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.27. URL https://aclanthology.org/2021.acl-short.27.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Sepp Hochreiter and Jürgen Schmidhuber. Flat Minima. *Neural Computation*, 9(1): 1–42, 01 1997a. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997b. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

Wei Hu, Lechao Xiao, Ben Adlam, and Jeffrey Pennington. 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, 2020. Curran Associates Inc. ISBN 9781713829546.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015. URL http://arxiv.org/abs/1508.01991.

Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In *NeurIPS*, 2019. URL https://proceedings.neurips.cc/paper/2019/file/c20a7ce2a627ba838cfbff082db35197-Paper.pdf.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 336–351. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/idrissi22a.html.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 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, pages 624–635, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445923. URL https://doi.org/10.1145/3442188.3445923.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://www.aclweb.org/anthology/D17-1215.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.197. URL https://aclanthology.org/2020.acl-main.197.

Yichen Jiang and Mohit Bansal. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1262. URL https://aclanthology.org/P19-1262.

Richard Johansson and Alessandro Moschitti. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-2910.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147.

Masahiro Kaneko and Danushka Bollegala. Debiasing pre-trained contextualised embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1256–1266, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.107. URL https://aclanthology.org/2021.eacl-main.107.

Junmo Kang, Haritz Puerto San Roman, and sung-hyon myaeng. Let me know what to ask: Interrogative-word-aware question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 163–171, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5822. URL https://www.aclweb.org/anthology/D19-5822.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. Improving neural question generation using answer separation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6602–6609, Jul. 2019. doi: 10.1609/aaai.v33i01.33016602. URL https://ojs.aaai.org/index.php/AAAI/article/view/4629.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1181. URL http://aclweb.org/anthology/D14-1181.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. URL https://arxiv.org/abs/1312.6114.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.84. URL https://www.aclweb.org/anthology/2020.emnlp-main.84.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. Putting the horse before the cart: A generator-evaluator framework for question generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 812–821, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1076. URL https://www.aclweb.org/anthology/K19-1076.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy, August

2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3823. URL https://aclanthology.org/W19-3823.

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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019. doi: 10.1162/tacl_a_00276. URL https://www.aclweb.org/anthology/Q19-1026.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1086. URL https://www.aclweb.org/anthology/P15-1086.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.85. URL https://aclanthology.org/2021.findings-acl.85.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. Mind the gap: Assessing temporal generalization in neural language models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29348–29363. Curran Associates, Inc., 2021.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.20.

Hector J. Levesque. On our best behaviour. *Artif. Intell.*, 212(1):27–35, July 2014. ISSN 0004-3702. doi: 10.1016/j.artint.2014.03.007. URL https://doi.org/10.1016/j.artint.2014.03.007.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Mike Lewis and Angela Fan. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkx0RjA9tX.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018a. URL https://proceedings.neurips.cc/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf.

Jingjing Li, Yifan Gao, Lidong Bing, Irwin King, and Michael R. Lyu. Improving question generation with to the point context. 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 3214–3224, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1317. URL https://www.aclweb.org/anthology/D19-1317.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://www.aclweb.org/anthology/N16-1014.

Jiwei Li, Will Monroe, and Dan Jurafsky. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*, 2016b. URL https://arxiv.org/abs/1611.08562.

Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016c.

Juntao Li, Yan Song, Haisong Zhang, Dongmin Chen, Shuming Shi, Dongyan Zhao, and Rui Yan. Generating classical chinese poems via conditional variational autoencoder and adversarial training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3890–3900. Association for Computational Linguistics, 2018b. URL http://aclweb.org/anthology/D18-1423.

Jieyu Lin, Jiajie Zou, and Nai Ding. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 333–342, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.43. URL https://aclanthology.org/2021.acl-short.43.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-2114.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. Learning to generate questions by learningwhat not to generate. In *The World Wide Web Conference*, WWW '19, pages 1106–1118, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313737. URL https://doi.org/10.1145/3308558.3313737.

Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, WWW '20, pages 2032–2043, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380270. URL https://doi.org/10.1145/3366423.3380270.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*, 2020b.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Shayne Longpre, Yi Lu, and Chris DuBois. On the transferability of minimal prediction preserving inputs in question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1288–1300, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.101. URL https://aclanthology.org/2021.naacl-main.101.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mNtmhaDkAr.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*, 2019.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.532. URL https://aclanthology.org/2020.acl-main.532.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://www.aclweb.org/anthology/P19-1334.

Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl the planning domain definition language, 1998.

Niall McLaughlin, Jesus M. Del Rincon, and Paul Miller. Data-augmentation for reducing dataset bias in person re-identification. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, Karlsruhe, Germany, 2015. IEEE.

Danielle S McNamara and Joe Magliano. Toward a comprehensive model of comprehension. *Psychology of learning and motivation*, 51:297–384, 2009.

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1416. URL https://aclanthology.org/P19-1416.

Jack Mostow and Wei Chen. Generating instruction automatically for the reading strategy of self-questioning. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 465–472, NLD, 2009. IOS Press. ISBN 9781607500285.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 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, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1176. URL https://aclanthology.org/P18-1176.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems*, volume 33, pages 20673–20684. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/eddc3427c5d77843c2253f1e799fe933-Paper.pdf.

Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Let's ask again: Refine network for automatic question generation. 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 3312–3321, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1326. URL https://www.aclweb.org/anthology/D19-1326.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://www.aclweb.org/anthology/2020.acl-main.441.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

Judea Pearl. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversity-Press*, 19, 2000.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1162. URL http://aclweb.org/anthology/D14-1162.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. Rissanen data analysis: Examining dataset characteristics via description length. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8500–8513. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/perez21a.html.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=aExAsh1UHZo.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL https://aclanthology.org/S18-2023.

Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. Factorizing perception and policy for interactive instruction following. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1868–1877, 2021. doi: 10.1109/ICCV48922.2021.00190.

X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018. doi: 10.1109/CVPR.2018.00886.

Jiazuo Qiu and Deyi Xiong. Generating highly relevant questions. 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 5982–5986, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1614. URL https://www.aclweb.org/anthology/D19-1614.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. NoiseQA: Challenge set evaluation for user-centric question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2976–2992, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.259.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. Are red roses red? Evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1621. URL https://www.aclweb.org/anthology/P19-1621.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5): 465–471, 1978. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(78)90005-5. URL https://www.sciencedirect.com/science/article/pii/0005109878900055.

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

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, April 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i05.6399. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Hf3qXoiNkR.

Viktor Schlegel, Goran Nenadic, and Riza Batista-Navarro. Semantics Altering Modifications for Evaluating Comprehension in Machine Reading. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13762–13770, May 2021. doi: 10.1609/aaai.v35i15.17622. URL https://ojs.aaai.org/index.php/AAAI/article/view/17622.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1341. URL https://www.aclweb.org/anthology/D19-1341.

Roy Schwartz and Gabriel Stanovsky. On the limitations of dataset balancing: The lost battle against spurious correlations. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2182–2194, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.168. URL https://aclanthology.org/2022.findings-naacl.168.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 15–25, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1004. URL https://aclanthology.org/K17-1004.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will DNNs choose? a study from the parameter-space perspective. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=qRDQi3ocgR3.

Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1066. URL https://www.aclweb.org/anthology/D17-1066.

Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.190.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HJ0UKP9ge.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3295–3301. AAAI Press, 2017. URL http://dl.acm.org/citation.cfm?id=3298023.3298047.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc., 2020.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2119. URL https://www.aclweb.org/anthology/P18-2119.

Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. A conditional variational framework for dialog generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 504–509, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2080. URL https://www.aclweb.org/anthology/P17-2080.

Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. Improving the robustness of QA models to challenge sets with variational question-answer pair generation. In *Proceedings of the ACL-IJCNLP 2021 Student Research Workshop*, pages 197–214, Online, August 2021a. Association for Computational Linguistics. URL https://aclanthology.org/2021.acl-srw.21.

Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 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*, pages 63–72, Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrqa-1.6. URL https://aclanthology.org/2021.mrqa-1.6.

Robik Shrestha, Kushal Kafle, and Christopher Kanan. Occamnets: Mitigating dataset bias by favoring simpler hypotheses. 2022.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *CVPR*, 2020. URL https://arxiv.org/abs/1912.01734.

Qingyi Si, Fandong Meng, Mingyu Zheng, Zheng Lin, Yuanxin Liu, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. Language prior is not the only shortcut: A benchmark for shortcut learning in vqa. *arXiv preprint arXiv:2210.04692*, 2022.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.569. URL https://aclanthology.org/2021.acl-long.569.

Linfeng Song, Zhiguo Wang, Wael Hamza, Yue Zhang, and Daniel Gildea. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 569–574. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-2090. URL http://aclweb.org/anthology/N18-2090.

Neha Srikanth and Rachel Rudinger. Partial-input baselines show that NLI models can ignore context, but they don't. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4753–4763, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.350. URL https://aclanthology.org/2022.naacl-main.350.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Joe Stacey, Yonatan Belinkov, and Marek Rei. Supervising Model Attention with Human Explanations for Robust Natural Language Inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11349–11357, June 2022. doi: 10.1609/aaai.v36i10.21386. URL https://ojs.aaai.org/index.php/AAAI/article/view/21386.

Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. Neural models for key phrase extraction and question generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 78–88. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/W18-2609.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1453. URL https://www.aclweb.org/anthology/D18-1453.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. Assessing the Benchmarking Capacity of Machine Reading Comprehension Datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8918–8927, April 2020. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v34i05.6422. URL https://aaai.org/ojs/index.php/AAAI/article/view/6422.

Saku Sugawara, Pontus Stenetorp, and Akiko Aizawa. Benchmarking machine reading comprehension: A psychological perspective. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1592–1612, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.137. URL https://aclanthology.org/2021.eacl-main.137.

Md Arafat Sultan, Shubham Chandel, Ramón Fernandez Astudillo, and Vittorio Castelli. On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online, July 2020. Association for Computational Linguis-

tics. doi: 10.18653/v1/2020.acl-main.500. URL https://www.aclweb.org/anthology/2020.acl-main.500.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/D18-1427.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL https://papers.nips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Alon Talmor and Jonathan Berant. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1485. URL https://www.aclweb.org/anthology/P19-1485.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16761–16772, June 2022.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJzSgnRcKX.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. Debiasing NLU Models via Causal Intervention and Counterfactual Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11376–11384, June 2022. doi: 10.1609/aaai.v36i10.21389. URL https://ojs.aaai.org/index.php/AAAI/article/view/21389.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL https://aclanthology.org/W17-2623.

Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1239.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020. doi: 10.1162/tacl_a_00335. URL https://aclanthology.org/2020.tacl-1.40.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

Alan M. Turing. Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423. doi: 10.1093/mind/LIX.236.433. URL https://doi.org/10.1093/mind/LIX.236.433.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.770. URL https://www.aclweb.org/anthology/2020.acl-main.770.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.613. URL https://www.aclweb.org/anthology/2020.emnlp-main.613.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5866-pointer-networks.pdf.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. Info{bert}: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=hpH98mK5Puk.

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng. CLINE: Contrastive learning with semantic negative examples for natural language understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2332–2342, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.181. URL https://aclanthology.org/2021.acl-long.181.

Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1170. URL https://www.aclweb.org/anthology/N16-1170.

Shuohang Wang and Jing Jiang. Machine comprehension using match-LSTM and answer pointer. In *International Conference on Learning Representations*, 4 2017. URL https://openreview.net/forum?id=B1-q5Pqxl.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Yang Liu, and Xuanjing Huang. A multi-agent communication framework for question-worthy phrase extraction and question generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 7168–7175, Jul. 2019b. doi: 10.1609/aaai.v33i01.33017168. URL https://ojs.aaai.org/index.php/AAAI/article/view/4700.

Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1670. URL https://www.aclweb.org/anthology/D19-1670.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 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, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1028. URL https://www.aclweb.org/anthology/K17-1028.

Chuan Wen, Jianing Qian, Jierui Lin, Jiaye Teng, Dinesh Jayaraman, and Yang Gao. Fighting fire with fire: Avoiding DNN shortcuts through priming. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23723–23750. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wen22d.html.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: When to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, Gold Coast, QLD, Australia, 2016. IEEE.

Mingzhu Wu, Nafise Sadat Moosavi, Andreas Rücklé, and Iryna Gurevych. Improving QA generalization by concurrent modeling of multiple biases. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 839–853, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.74. URL https://www.aclweb.org/anthology/2020.findings-emnlp.74.

Winston Wu, Dustin Arendt, and Svitlana Volkova. Evaluating neural model robustness for machine comprehension. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2470–2481, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.210.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 2660–2676, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.190. URL https://aclanthology.org/2022.acl-long.190.

Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. Diversity-promoting GAN: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1428. URL https://www.aclweb.org/anthology/D18-1428.

Zhicheng Yang, Jinghui Qin, Jiaqi Chen, and Xiaodan Liang. Unbiased math word problems benchmark for mitigating solving bias. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1401–1408, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.104. URL https://aclanthology.org/2022.findings-naacl.104.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 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)*, pages 1040–1050, Vancouver, Canada, July 2017a. Association for Computational Linguistics. doi: 10.18653/v1/P17-1096. URL https://aclanthology.org/P17-1096.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 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, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3881–3890. JMLR.org, 2017b.

Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu. Teaching machines to ask questions. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4546–4552. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/632. URL https://doi.org/10.24963/ijcai.2018/632.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, 2018.

Kexin Yi*, Chuang Gan*, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *ICLR*, 2020.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019. URL https://arxiv.org/abs/1901.11373.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgJtT4tvB.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25. Association for Computational Linguistics, 2017. doi: 10.18653/v1/W17-2603. URL http://aclweb.org/anthology/W17-2603.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1050. URL https://www.aclweb.org/anthology/D16-1050.

Shiyue Zhang and Mohit Bansal. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1253. URL https://aclanthology.org/D19-1253.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31*, pages 1810–1820. Curran Associates, Inc., 2018.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL https://www.aclweb.org/anthology/N19-1131.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. URL https://www.aclweb.org/anthology/N18-2003.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017a. URL https://arxiv.org/abs/1706.02262.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664. Association for Computational Linguistics, 2017b. doi: 10.18653/v1/P17-1061. URL http://aclweb.org/anthology/P17-1061.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910. Association for Computational Linguistics, 2018b. URL http://aclweb.org/anthology/D18-1424.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study. In *Natural Language Processing and Chinese Computing*, pages 662–671, Cham, 2018. Springer International Publishing. ISBN 978-3-319-73618-1.

Wangchunshu Zhou, Qifei Li, and Chenle Li. Learning from perturbations: Diverse and informative dialogue generation with inverse adversarial training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 694–703, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.57. URL https://aclanthology.org/2021.acl-long.57.

Wenjie Zhou, Minghua Zhang, and Yunfang Wu. Question-type driven question generation. 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 6031–6036, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1622. URL https://www.aclweb.org/anthology/D19-1622.

Xiang Zhou and Mohit Bansal. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.773. URL https://www.aclweb.org/anthology/2020.acl-main.773.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygzbyHFvB.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1097–1100, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356572. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.