

Look to the Right: Mitigating Relative Position Bias in Extractive Question Answering

Kazutoshi Shinoda^{1,2} Saku Sugawara² Akiko Aizawa^{1,2}

¹The University of Tokyo

²National Institute of Informatics

shinoda@is.s.u-tokyo.ac.jp

{saku,aizawa}@nii.ac.jp

Abstract

Extractive question answering (QA) models tend to exploit spurious correlations to make predictions when a training set has unintended biases. This tendency results in models not being generalizable to examples where the correlations do not hold. Determining the spurious correlations QA models can exploit is crucial in building generalizable QA models in real-world applications; moreover, a method needs to be developed that prevents these models from learning the spurious correlations even when a training set is biased. In this study, we discovered that the relative position of an answer, which is defined as the relative distance from an answer span to the closest question-context overlap word, can be exploited by QA models as superficial cues for making predictions. Specifically, we find that when the relative positions in a training set are biased, the performance on examples with relative positions unseen during training is significantly degraded. To mitigate the performance degradation for unseen relative positions, we propose an ensemble-based debiasing method that does not require prior knowledge about the distribution of relative positions. We demonstrate that the proposed method mitigates the models' reliance on relative positions using the biased and full SQuAD dataset. We hope that this study can help enhance the generalization ability of QA models in real-world applications.¹

1 Introduction

Deep learning-based natural language understanding (NLU) models are prone to use spurious correlations in the training set. This tendency results in models' 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. Question answering (QA) models trained on intentionally biased training sets are more likely

¹Our codes are available at <https://github.com/KazutoshiShinoda/RelativePositionBias>.

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

Table 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 §2.1 for the definition of the relative position.

to learn solutions based on spurious correlations rather than on causal relationships between inputs and labels. For example, QA models can learn question-answer type matching heuristics (Lewis and Fan, 2019), and absolute-positional correlations (Ko et al., 2020), particularly when a training set is biased toward examples with corresponding spurious correlations. Collecting a fully unbiased dataset is challenging. Therefore, it is vital to discover possible dataset biases that can degrade the generalization and develop debiasing methods to learn generalizable solutions even when training on unintentionally biased datasets.

In extractive QA (e.g., Rajpurkar et al., 2016), in which answers to questions are spans in textual contexts, we find that the relative position of an answer, which is defined 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 1 for the

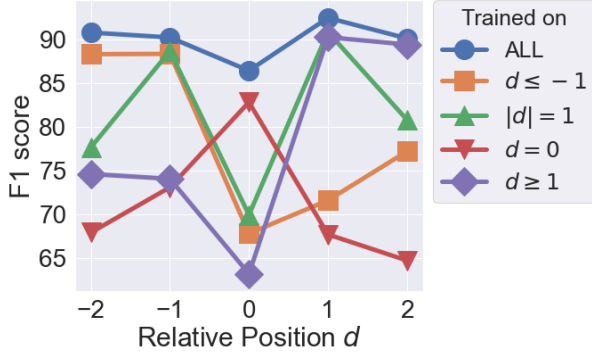


Figure 1: F1 score for each relative position d in the SQuAD development set. “ALL” in the legend refers to a QA model trained on all the examples in the SQuAD training set. The other terms refer to models trained only on examples for which the respective conditionals are satisfied. BERT-base was used for the QA models. The accuracy is comparable to ALL for examples with seen relative positions, but worse for others. Please refer to §2.1 for the definition of d .

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, as shown in Figure 1. For example, when a QA model is trained on examples with negative relative positions, as shown in Table 1, the QA performance on examples with non-negative relative positions is degraded by 10 ~ 20 points, as indicated by the square markers (■) in Figure 1. Similar phenomena were observed when the distribution of the relative positions in the training set was biased differently, as shown in Figure 1. This observation implies that the model may preferentially learn to find answers from seen relative positions.

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 this end, we propose debiasing methods based on an ensemble (Hinton, 2002) of intentionally biased and main models. The biased model makes predictions relying on relative positions, which promotes the main model not depending 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 improved the scores for unseen relative positions by 0~10 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. Furthermore, when applied to the full training set, our method improves the generalization to examples where questions and contexts have no lexical overlap.

2 Relative Position Bias

2.1 Definition

In this study, we call a word that is contained in both the question and the context as an overlapping word. Let d be the relative position of the nearest overlapping word from the answer span in extractive QA. If w is 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:

$$f(j, s, e) = \begin{cases} j - s, & \text{for } j < s \\ 0, & \text{for } s \leq j \leq e \\ j - e, & \text{for } j > e \end{cases} \quad (1)$$

$$D = \{f(j, s, e) | w_j^c \in q\} \quad (2)$$

$$d = \operatorname{argmin}_{d' \in D} |d'| \quad (3)$$

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.² 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 considerable (Sugawara et al., 2018), the one with the lowest absolute value in Equation 3 is used as the relative position.³

2.2 Distribution of Relative Position d

Figure 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 Appendix B for more details. This

²Because function words as well as content words are important clues for reading comprehension, D in Equation 2 can contain function and content words.

³There are a few cases where d in Equation 3 is not fixed to one value. However, such examples are excluded from the training and evaluation sets for brevity.

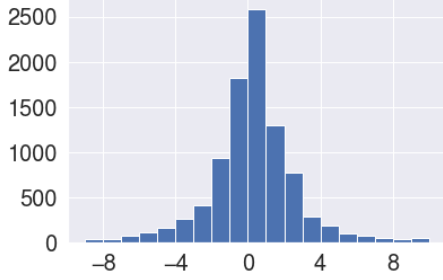


Figure 2: Histogram of relative position d in the SQuAD training set.

difference may be caused by how the datasets were collected or to the contexts’ domains. Therefore, building a QA model that does not overfit a specific distribution of relative positions is necessary.

3 Method

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), following Ko et al. (2020). In both methods, after an intentionally biased model is prepared, the cross-entropy loss is calculated using a product of the biased and main models. The biased model is fixed when minimizing the loss to train the main model. 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 as \hat{p} .

3.1.1 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 follows.

$$\hat{p} = \text{softmax}(\log p + \log b) \quad (4)$$

This encourages the the main model to learn examples where the biased model make incorrect predictions, rather than examples that contain biases allowing the biased model to make correct predictions.

3.1.2 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.

$$\hat{p} = \text{softmax}(\log p + g(c, q) \log b) \quad (5)$$

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

3.2 Biased Model

We describe how to construct the biased model described in §3.1. The first model, Answer Prior, is a fully rule-based model with a prior probability of answer span. The second, the Position-only model, is a QA model trained with only binarized contexts from which models can know only which token is an overlapping word.

3.2.1 Answer Prior (AnsPrior)

We first use a simple heuristic called AnsPrior as a biased model. 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 1.

$$b_i = \begin{cases} \mathbb{1}[w_{i+1}^c \in q] / Z, & \text{for } d \leq -1 \\ \mathbb{1}[(w_{i+1}^c \in q) \vee (w_{i-1}^c \in q)] / Z, & \text{for } |d| = 1 \\ \mathbb{1}[w_i^c \in q] / Z, & \text{for } d = 0 \\ \mathbb{1}[w_{i-1}^c \in q] / Z, & \text{for } d \geq 1 \end{cases} \quad (6)$$

where $\mathbb{1}$ denotes an indicator function (e.g., $\mathbb{1}[w_{i+1}^c \in q]$ returns 1 if w_{i+1}^c is contained in q , otherwise it returns 0) and 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 as they are prior probabilities specific to the distribution of relative positions of a training set.

3.2.2 Position-only model (PosOnly)

We propose PosOnly as a biased model that can be used without prior knowledge about the distributions of relative positions in training sets. PosOnly accepts as input the sequence of binary variables indicating if a word in a context is overlapped

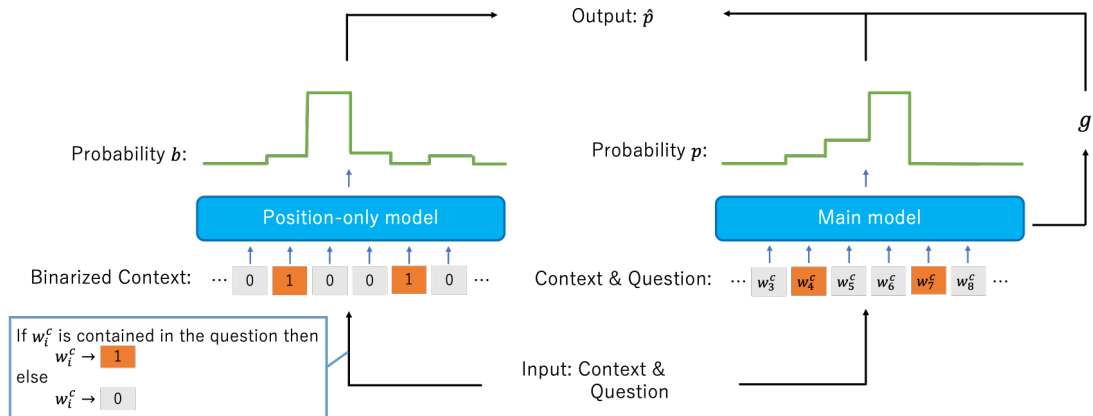


Figure 3: Illustration of LearnedMixIn with Position-only model (PosOnly) as a biased model. The output \hat{p} is used for computing the loss, and the probability p is used for inference.

with a question. The only information available to PosOnly to predict answer spans is the relative distances from the overlapping words. Hence, it 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 3.

4 Experiments

4.1 Generalization to Unseen Relative Positions

Dataset SQuAD 1.1 (Rajpurkar et al., 2016) was used as the dataset. The training set is filtered to be biased in four different ways to assess the applicability of our methods. The four subsets were constructed by extracting only examples whose relative positions d satisfied the conditions $d \leq -1$, $|d| = 1$, $d = 0$, and $d \geq 1$. The sizes of the subsets are 33,256, 30,003, 21,266, and 25,191, respectively. The scores of BERT-base trained on the full 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.

Method We compare 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. The training details are given in Appendix A.

Results Table 2 shows the results. First, when the full training set is used (ALL), the performance

of the BERT-base baseline exceeds 90 points when $|d| = 1, 2$, whereas it drops by about eight points when $|d| \geq 3$. As shown in §2.2, there is a correlation between the accuracy for a specific range of relative positions and the frequency of examples whose relative positions are in the corresponding range in the training set.

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 relative positions $|d| = 1$, 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 exploited spurious correlations regarding relative positions 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 2. 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 $d \geq 3$.

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

In contrast, regarding the scores on examples with relative positions seen during training (i.e., the scores in the gray cells in Table 2), LearnedMixin-AnsPrior was superior to LearnedMixin-PosOnly. As pointed out in Utama et al. (2020), 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.

4.2 Effect of Mitigating Relative Position Bias in Normal Settings

Although we verified the effectiveness of our methods on intentionally biased datasets in §4.1, 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 train-

ing 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 3. 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.

5 Related Work

Recent studies have found that NLU models tend to learn shortcut solutions specific to the distribution of the training sets. In natural language inference,

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

Table 3: 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. The scores for each relative position are given in Appendix C.

models can use spurious correlations regarding lexical items (Gururangan et al., 2018) and lexical overlap (McCoy et al., 2019). In extractive QA, existing studies have shown that models can learn several kinds of shortcut solutions. Weissenborn et al. (2017) showed that a substantial number of questions could be answered only by matching the types of questions and answers. Sugawara et al. (2018) demonstrated that only using partial inputs is sufficient for finding correct answers in most cases. Ko et al. (2020) indicated that the absolute position bias of answers could severely degrade the generalization. Shinoda et al. (2021) showed that question generation models can amplify lexical overlap bias when used for data augmentation in QA.

Sampling training and test sets from different distributions (Jia and Liang, 2017; Fisch et al., 2019; Lewis and Fan, 2019; Ko et al., 2020) is one of the most effective frameworks to assess whether models can learn generalizable solutions. In this study, we also employed the framework to analyze the generalization capability from a new perspective of relative position bias, and developed effective debiasing methods for this bias. Specifically, we used the same loss functions as Ko et al. (2020), and proposed new biased models in this study.

6 Conclusion

We showed that an extractive QA model tends to exploit relative position bias in training sets, causing the performance degradation for relative positions unseen during training. To mitigate this problem, we proposed new biased models that perform well with existing debiasing algorithms on intentionally filtered training sets. Furthermore, when applied to the full training set, our method improved the generalization to examples where questions and contexts have no common words. Future work includes refining the definition of relative position

bias that is more learnable for QA models, mitigating the trade-off between the accuracies for seen and unseen relative positions, and increasing the size of the test set with no lexical overlap between question and context.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by JST SPRING Grant Number JPMJSP2108 and JSPS KAKENHI Grant Numbers 22J13751 and 22K17954. This work was also supported by the SIP-2 “Big-data and AI-enabled Cyberspace Technologies” by the New Energy and Industrial Technology Development Organization (NEDO).

References

- Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. [Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. [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. Association for Computational Linguistics.

- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nature Machine Intelligence*, 2(11):665–673.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- He He, Sheng Zha, and Haohan Wang. 2019. [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. Association for Computational Linguistics.
- Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.
- Robin Jia and Percy Liang. 2017. [Adversarial examples for evaluating reading comprehension systems](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. [Look at the first sentence: Position bias in question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online. Association for Computational Linguistics.
- Mike Lewis and Angela Fan. 2019. [Generative question answering: Learning to answer the whole question](#). In *International Conference on Learning Representations*.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.
- Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. [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. Association for Computational Linguistics.
- Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. [What makes reading comprehension questions easier?](#) In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.
- Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. [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. Association for Computational Linguistics.
- Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. [Making neural QA as simple as possible but not simpler](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

A Training 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.

B Distribution of Relative Position d

The frequency distribution of relative positions in the SQuAD, NewsQA, TriviaQA, and NaturalQuestions development sets is shown in Figure 4. Although there are differences among the datasets, they all show that the relative positions are biased around 0.

C Results on the Full Training Set

The detailed scores of our methods when trained on the full training set are shown in Table 4. Compared to the BERT-base baseline, LearnedMixin-PosOnly can maintain the score in each column.

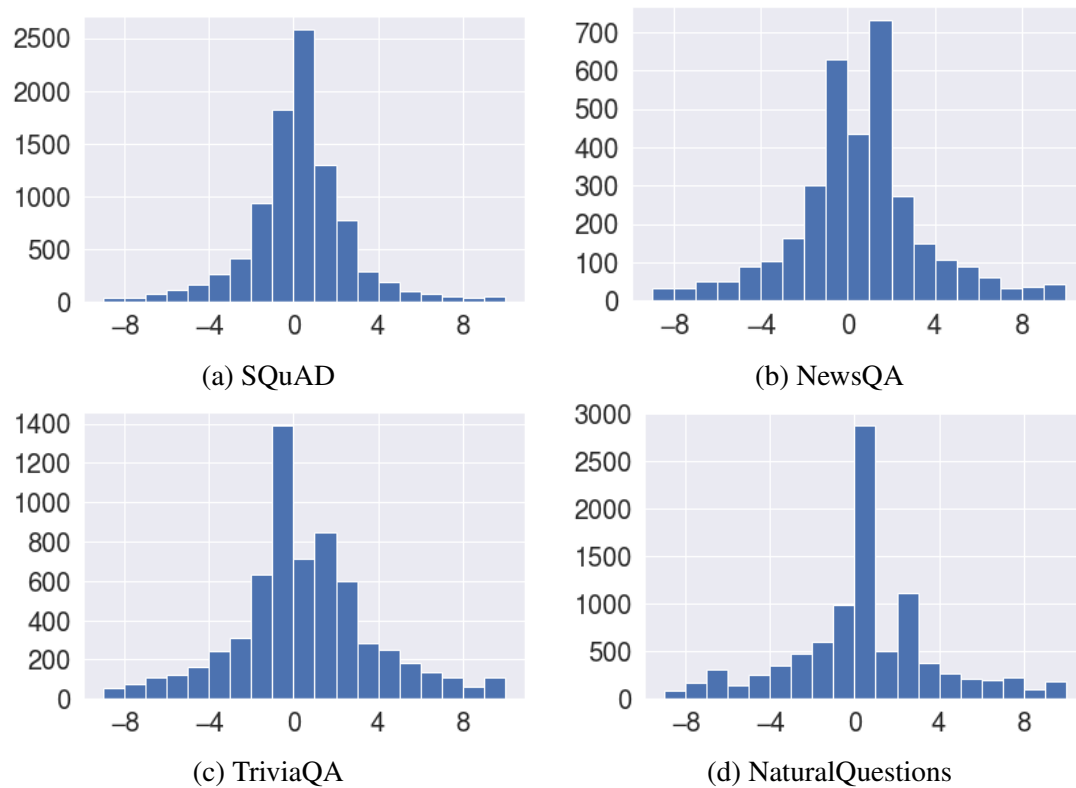


Figure 4: Histograms of relative position d in the SQuAD, NewsQA, TriviaQA, and NaturalQuestions development sets.

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
ALL	BiasProduct-PosOnly	81.04	85.72	86.89	83.53	88.47	87.16	80.61
ALL	LearnedMixin-PosOnly	82.48	90.63	90.15	85.69	91.33	89.14	81.21

Table 4: F1 scores for each subset of the SQuAD development set.