Counterfactual QA: Eliminating Bias in Question Answering

NTU-CE7455: Project Final Report

Zhu Beier[1,*] **Wang Tan**[1,*] **Kong Lingdong**[1,*]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[*]equal contribution
`{BEIER002, TAN317, LINGDONG001}@e.ntu.edu.sg`

Abstract

Question Answering (QA) task in Natural Language Processing corresponds to predict the start and end positions of answers in the given paragraph. Existing methods mainly use the powerful Language Model (LM), *e.g.*, BERT or Transformer to encode the paragraph and question sentences together and predict the answer position. However, we find that the QA models may tend to rely on the position bias (*i.e.*, simply remember the answer position) as a shortcut and thus fail to sufficiently learn the true "knowledge", leading to the poor performance on the out-of-distribution QA data. In this project report, we investigate how to debias the QA task motivated by the *causality theory* and propose a novel counterfactual QA framework called CF-EQA. It enables us to capture the answer position bias as the direct causal effect of the paragraph on answers and reduce the bias by subtracting the direct effect from the total causal effect. Abundant experiments demonstrate the effectiveness of our method to generalize well to different distribution QA data[1].

1 Key Information

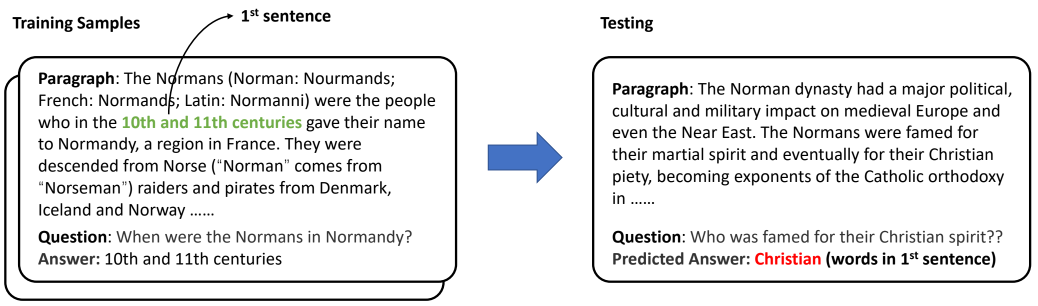
- Mentor: No
- External Collaborators: No
- Publication plan: EMNLP

2 Introduction

Question answering (QA) is a task that requires the machine to respond to questions based on the given passage context. Many researchers have been drawn to large-scale QA datasets in order to develop successful QA models with the introduction of deep learning. In this paper we focus on the extractive QA task: assumes that the answers are always lie in the passage and the deep learning model is trained to predict the start and end positions of the answers. Recently, extractive QA models have been known to outperform humans in some datasets [1].

However, we found that the generalization performance of existing models is still far from that of humans. As shown in Figure 1, we follow [2] to define the language bias of the QA model as the position bias — the answer position in the paragraph. For example, if QA models are trained on a biased dataset (answers are almost lie in the first sentence), then in testing the models tend to predict the word in the first sentence regardless of whether it is right. This phenomenon indicates that the models predicting the position can be severely biased when trained on datasets that have a very skewed answer position distribution. As a result, BERT [3] trained on a biased training set where every answer that appears in the first sentence only achieves a 37.48% F1 score in the SQuAD

[1]Code of this work is accessible at: `https://github.com/ldkong1205/cf-eqa`.



Training Samples
1st sentence
Paragraph: The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway ……
Question: When were the Normans in Normandy?
Answer: 10th and 11th centuries
Testing
Paragraph: The Norman dynasty had a major political, cultural and military impact on medieval Europe and even the Near East. The Normans were famed for their martial spirit and eventually for their Christian piety, becoming exponents of the Catholic orthodoxy in ……
Question: Who was famed for their Christian spirit??
Predicted Answer: Christian (words in 1st sentence)

Figure 1: The language position bias in extractive QA task.

development set whereas the same model trained on the same amount of randomly sampled examples can achieve 85.06% F1 score.

To tackle this problem, in this paper we propose a novel counterfactual framework CF-EQA (Counterfactual Extractive Question Answering framework) motivated by the counterfactual reasoning and causal effects.

Overall, we formulate the answer position bias as the direct causal effect of the paragraph on answers, and mitigate it by subtracting the direct position effect from the total effect. Specifically, we ask the **counterfactual** question: *What would the answer be, if the machine not sees the paragraph and question, but just achieves the position bias explicitly?* Compared to the **conventional** extractive QA: *What will the answer be, if the machine sees the paragraph, question and achieves the position bias explicitly?*, we manually blind the achievement of the model to the paragraph context and the question and the QA models can only rely on the dataset bias. Therefore, the language bias can be identified by estimating the direct causal effect of position bias on the answer.

In the training stage, we follow the previous methods [2, 4] to train ensemble methods with the prevailing QA BERT model and a single answer position bias branch. Then during testing, CF-EQA uses the debiased causal effect for inference which is obtained by subtracting the pure position bias effect from the total effect. With large amounts of experiments, our CF-EQA model shows great superiority compared to the baseline method and achieves new state-of-the-art results with significant improvements.

3 Related Work

3.1 Extractive Question Answering

As a basic task of NLP, there are various QA models proposed to predict the start and end positions. Following the structure of earlier deep learning-based QA models [5, 6, 7], currently, the methods can be mainly divided into three major categories: Pointer-based, LM-based, and semi-supervised-based. For the Pointer-based methods, one of the earliest works in extractive QA from [6] experimented with generative models based on index sequence generation via pointer networks [8] and now directly used traditional boundary models that focus on the prediction of the start/end of an answer span. Their work has shown substantial improvement of boundary models over the index sequence generative models.

Recently the success of Language Models [3, 9, 10] also greatly boost the performance of the extractive QA task. These methods try to learn language representations and are conceived to be used as the main architecture to the downstream extractive QA tasks by designing the proxy task of predicting the masked words and next sentence. These methods achieve the current state-of-the-art performance and we use the BERT as the base model in our project. Finally, the semi-supervised methods use other sources of data to promote the performance of extractive QA. For example, [11] exploited the structure of dataset passages to create cloze-style questions. They pretrain a powerful neural network on the cloze-style pseudo-questions and fine-tune the model on the labeled examples. [12] shows an improvement in open-domain QA when using distantly supervised ground truths with the same surface form as the original ground truth.

3.2 Debiasing Strategies

There are many works that try to tackle the bias in the language, which is usually called "language prior" [13, 14]. For example, the answer always appears in the first sentence; in visual question answering task simply answering "tennis" to the sport-related questions can achieve approximately 40% accuracy on the visual question answering dataset. Most recent solutions to reduce language bias can be grouped into two categories: implicit/explicit data augmentation and ensemble-based methods. First, recent works [15, 16] automatically generate additional language sources to balance the distribution of the training data. Second, the ensemble-based methods [17, 4] learn a biased model given source of bias as input, and debias through logits re-weighting or logits ensembling.

The most similar work to ours is [2] which apply the ensemble-based methods into extractive QA task to eliminate the position prior bias. Our proposed methods further refine this debiasing strategy from a causal view and boost the performance.

3.3 Causal Inference in Deep Learning

Counterfactual thinking and causal inference have inspired several studies in computer vision, including visual explanations [18, 19, 20, 21, 22], scene graph generation [23, 24], image recognition [24], video analysis [25, 26], zero-shot and few-shot learning [27, 28] incremental learning [29], representation learning [30, 31], semantic segmentation [32], and vision-language tasks [15, 33, 34, 35, 36, 37]. Especially, counterfactual learning has been exploited in recent visual question answering studies [15, 33, 16]. Different from these works that generate counterfactual samples for *debiased* training, our cause-effect look focuses on counterfactual inference with even *biased* training data.

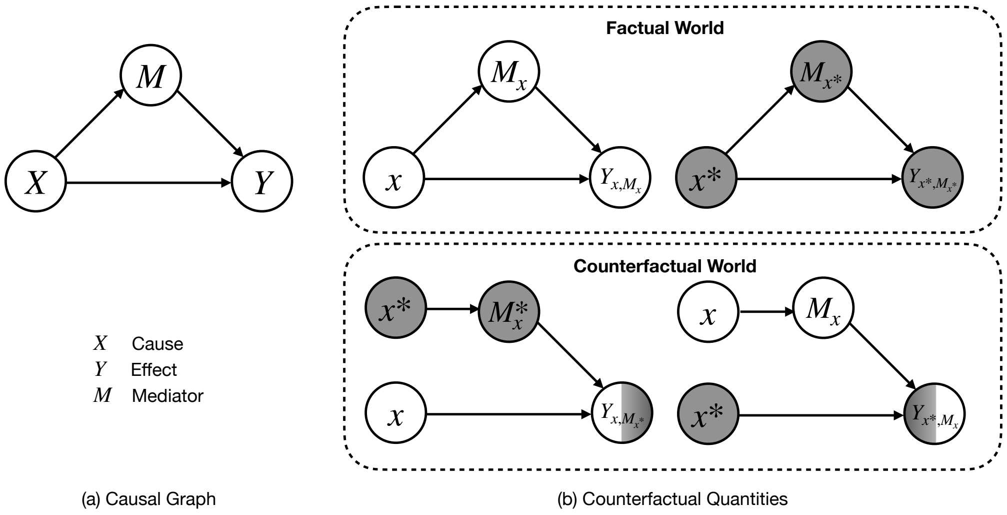


Figure 2: (a) Example of causal graph. (b) Counterfactual quantities. Variables in the treatment group are marked in white, while variables in the no-treatment group are marked in gray.

4 Approach

4.1 Preliminaries

This section introduces the basis of causal graph, counterfactual notations, and decomposition of a total effect.

Causal graph describes the causal relationships via a directed acyclic graph. We construct a causal graph with three variables $\{X, M, Y\}$ in Fig. 2(a). If X has a *direct* effect on Y, Y is the child of the exposure X, *i.e.*, $X \rightarrow Y$. If X causes Y through a variable M, we say that X has an *indirect* effect on Y and M is a mediator, *i.e.*, $X \rightarrow M \rightarrow Y$.

Counterfactual notation $Y_{x,m}$ states that Y would be $Y_{x,m}$, if X had been x and M had been m (the variables are denoted in capital letters, and the observed value is denoted as lowercase letter). We let M_x denote the value of the mediator had the exposure X been set to x. In the factual world, Y is set to Y_{x,M_x}. In the counterfactual world, the mediator is set to a different value. For example, $Y_{x,M_{x^*}}$ captures the effect of exposure $X = x$ on outcome Y, intervening to fix M to another value when X had been x^*. The visualization of the counterfactual quantities can be found in Fig. 2(b).

Effect decomposition [38, 39], allows one to decompose a *total* effect into a *direct* and an *indirect* effect. Let x^* denote the value of the exposure X under the no-treatment situation, and x denotes the value of the exposure X under the treatment situation. The total effect (TE) of the treatment $X = x$ comparing to the no-treatment $X = x^*$ is defined as:

$$\mathrm{TE} = Y_{x,M_x} - Y_{x^*,M_x^*} \tag{1}$$

The total effect can be decomposed as follows:

$$\mathrm{TE} = Y_{x,M_x} - Y_{x^*,M_x^*} = (Y_{x,M_x^*} - Y_{x^*,M_x^*}) + (Y_{x,M_x} - Y_{x,M_x^*}), \tag{2}$$

where $Y_{x,M_x^*} - Y_{x^*,M_x^*}$ is defined as natural direct effect (NDE) and $Y_{x,M_x} - Y_{x,M_x^*}$ is defined as the total indirect effect (TIE). Thus, the total effect can be decomposed into the natural direct effect and total indirect effect: $\mathrm{TE} = \mathrm{NDE} + \mathrm{TIE}$.

4.2 A Causal View on QA Positional Bias

To study the extractive QA task, we construct a causal graph with three variables $\{X, M, Y\}$ in Fig 2(a): X is embeddings from the word, position and token type of the passage and question that will be fed into QA models (*e.g.*, BERT in our experiment settings), M is the propensity to answer in some specific positions and Y is the logits of the predicted answer positions.

The causal link $X \to M \to Y$ is constructed when the distribution of the answer position is highly skewed, *i.e.*, answers usually appear in the k-th sentence of the passage[2]. The model trained on the biased training set makes the hidden feature more dependent on the first few words when answers are usually given in the first sentences in the training set. The direct causal link $X \to Y$ is our desiderata for extractive QA which delivers position information without position bias.

4.3 De-biased Prediction: Natural Direct Effect Inference

In the training stage, we follow [2] to use a BERT to learn the mapping from passage and questions to answers, *i.e.*, $Y(x|\theta_{\mathrm{BERT}})$, where θ_{BERT} is the parameters of BERT. In the inference stage, the unbiased prediction is defined as NDE:

$$Y^{\mathrm{NDE}} = Y_{x,M_x^*} - Y_{x^*,M_x^*} \tag{3}$$

where x^* can be viewed as the null input. In this paper, it is defined as the empirical mean embedding representation of the passages and the questions of the training set:

$$x^* = \frac{1}{|\mathcal{D}_{\mathrm{train}}|} \sum_{i=1}^{|\mathcal{D}_{\mathrm{train}}|} x_i. \tag{4}$$

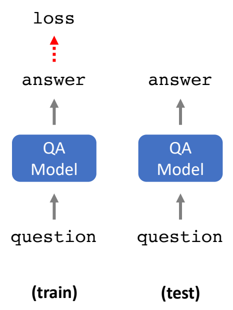
Similar to [40], we adopt the orthogonal decomposition for total effect as follows:

$$x = \lambda x^* + x^d, \tag{5}$$

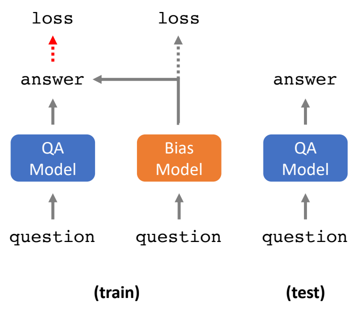
where $x^* \perp x^d$ and $\lambda = \cos(x, x^*) \cdot \|x\|_2$ is the projection length. The indirect effect is affected by x^* and the direct effect is affected by x^d, their combination composes the total effect. In this setting, NDE for Equ. (3) can be written as:

$$Y^{\mathrm{NDE}} = Y(x - \alpha\lambda x^* | \theta_{\mathrm{BERT}}) \tag{6}$$

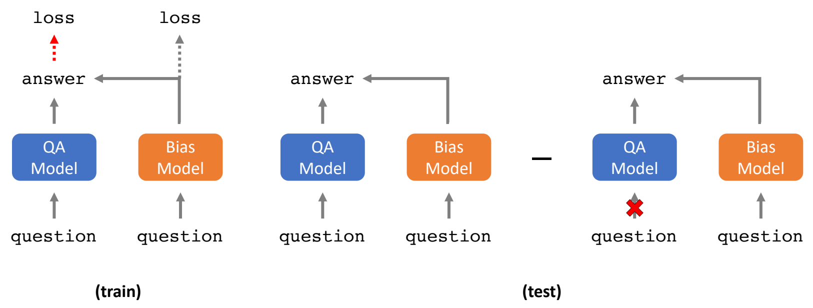
where α is a hyper-parameter that controls the trade-off between the indirect and direct effects. Using the calibrated embedding $x - \alpha\lambda x^*$ removes the spurious correlation between word position and answers. Fig. 3 gives an overview of the main difference among the traditional QA framework [3], ensemble-based framework [41, 17], and our proposed CF-EQA framework.



(a) Traditional QA



(b) Ensemble-based QA



(c) Counterfactual QA

Figure 3: Comparison among different QA models during the training and test stages. (a) Traditional QA framework (e.g., [3]); (2) Ensemble-based QA framework (e.g., [41] and [17]); (3) Our counterfactual QA framework.

5 Experiments

In this section, we conduct experiments to verify the effectiveness of our proposed counterfactual QA (CF-EQA) framework with several debiasing methods under both biased and full training set settings.

5.1 Data

We adopt the SQuAD dataset [1] and its subsets [2] through our experiments. Specifically, the five subsets $\text{SQuAD}_{\text{train}}^{k=i}$ are sampled based on the position k of answers appear in passages, where $i = 1, 2, 3, 4$, and ≥ 5 are sentence numbers. Recent work [2] found that more than 32% and 34% answers are laid in the first sentences in $\text{SQuAD}_{\text{train}}$ and $\text{SQuAD}_{\text{dev}}$, respectively, which introduces severe position biases in both training and evaluation.

5.2 Candidate Methods and Evaluation Metrics

Two types of methods are used in our experiments, i.e., relative position encoding [9] and ensemble-based debiasing [41, 17]. Specifically, we consider the following methods:

- **Random Position** perturbs the input positions and generates position embeddings by (1) randomly sampling indices within $[1, \text{Seq}_{\text{max}}]$, and (2) sorting indices accordingly to preserve input orders, where Seq_{max} denotes the maximum sequence length of BERT.
- **Entropy Regularization** aims at maximizing the entropy of normalized cosine similarity between word embeddings and their hidden representations [2]. The entropy regularization term is computed from the final network layer and added to the loss calculations.

Debiasing Method		**SQuAD$_{\text{dev}}^{k=1}$**		**SQuAD$_{\text{dev}}^{k=2,3,\dots}$**		**SQuAD$_{\text{dev}}$**	
		EM	**F1**	**EM**	**F1**	**EM**	**F1**
BERT trained on **SQuAD$_{\text{train}}^{k=1}$** (28,263 samples)							
Baseline	None (NAACL'19)	77.07	85.81	7.14	12.12	31.20	37.48
	RP (EMNLP'20)	69.95	80.73	27.32	34.63	41.99	50.49
	ER (EMNLP'20)	70.40	86.17	10.50	15.72	33.52	39.96
Ensemble	BP	78.05	86.73	13.33	18.73	36.27	42.72
	BP+CF-EQA (Ours)	74.32(-3.73)	83.08(-3.65)	25.16(+11.83)	31.60(+12.87)	42.07(+5.80)	49.31(+6.59)
	LM (EMNLP'19)	77.18	85.15	69.89	78.65	72.63	81.14
	LM+CF-EQA (Ours)	77.81(+0.63)	85.60(+0.45)	71.69(+1.80)	80.45(+1.80)	73.79(+1.16)	82.24(+1.10)
BERT trained on **SQuAD$_{\text{train}}$** (88,238 samples)							
Baseline	None (NAACL'19)	81.55	88.68	81.21	88.61	81.32	88.63
Ensemble	BP	81.80	88.94	80.61	88.44	81.02	88.61
	BP+CF-EQA (Ours)	81.81(+0.01)	88.98(+0.04)	80.61(+0.00)	88.44(+0.00)	81.03(+0.01)	88.63(+0.02)
	LM (EMNLP'19)	81.77	88.51	80.41	88.26	80.88	88.34
	LM+CF-EQA (Ours)	81.78(+0.01)	88.94(+0.43)	80.61(+0.20)	88.44(+0.18)	81.01(+0.13)	88.64(+0.30)

Table 1: Experimental results of different debiasing methods on SQuAD$_{\text{dev}}^{k=1}$, SQuAD$_{\text{dev}}^{k=2,3,\cdots}$, and SQuAD$_{\text{dev}}$ [1]. **RP**: Random Position [2]; **ER**: Entropy Regularization [2]; **BP**: Bias Product [41]; **LM**: Learned-Mixin [17]. **Top Block**: Models trained on SQuAD$_{\text{train}}^{k=1}$; **Bottom Block**: Models trained on SQuAD$_{\text{train}}$.

- **Bias Product** combines the probability distributions of the start/end position from the target model and the bias model. Here we follow [2] by defining the bias model as the prior distribution of answer positions in the dataset.
- **Learned-Mixin** improves the ensembling procedure one step further by adding in an affine transformation [17] on hidden representations before the softmax layer in the network.

We combine our CF-EQA framework with both Bias Product [41] and Learned-Mixin [17] to improve debiaing. For the evaluation metrics, we adopt both Exact Match (EM) and F1 scores for all experiments to compare the performance of different debiasing methods.

5.3 Experimental Details

To thoroughly analyze the debiasing ability of different methods, we first conduct experiments on SQuAD$^{k=1}$, i.e., answers for all passage-question pairs are laid in the first sentence. Then we adopt SQuAD$_{dev}^{k=1}$, SQuAD$_{dev}^{k=2,3,\cdots}$ (answers outside the first sentence), and SQuAD$_{dev}$ (the original SQuAD development set) for testing. We also train models on SQuAD$_{train}$ (the original SQuAD training set) for a more comprehensive comparison. Our next experiment further generalizes our findings to other bias subsets, i.e., SQuAD$^{k=2}$, SQuAD$^{k=3}$, SQuAD$^{k=4}$, and SQuAD$^{k=5,6,\cdots}$.

For Random Position [2], we follow the original setting by sampling t indices for BERT and sample $t = 384$ when Seq$_{\text{max}} = 512$. Indices are sorted in ascending order and then used to generate position embedding for each token position. For Bias Product [41] and Learned-Mixin [17], we follow the convention and adopt the Adam optimizer [42] with lr = 3e-5 and ϵ = 1e-8. The batch size is set as 12 and each model is trained for two epochs. We set the “trade-off" parameter α as 2.0 to weight between the direct and indirect effects. NVIDIA GeForce RTX 2080Ti graphics cards are used through our experiments.

5.4 Experimental Results

Table 1 shows the results of different debiasing methods trained on SQuAD$_{\text{train}}^{k=1}$ and the full set SQuAD$_{\text{train}}$, respectively, while Table 2 gives general results on other bais subsets.

Results on SQuAD$_{\text{train}}^{k=1}$. The top block of Table 1 gives the results for different methods trained on SQuAD$_{\text{train}}^{k=1}$. It can be easily observed that the position bias indeed hurts the evaluation performance

	SQuAD$_{\text{dev}}$							
SQuAD$_{\text{train}}^{k=i}$	$i=2$ (20,745 samples)		$i=3$ (15,669 samples)		$i=4$ (10,459 samples)		$i=5,6,...$ (13,102 samples)	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
None (NAACL'19)	36.16	43.14	44.76	52.89	49.13	58.01	57.95	66.69
BP	45.99	53.30	51.33	59.85	54.47	63.76	58.03	67.02
+CF-EQA (Ours)	51.72(+5.73)	59.46(+6.16)	51.77(+0.44)	60.32(+0.47)	57.82(+3.35)	67.35(+3.59)	58.24(+0.21)	67.45(+0.43)
LM (EMNLP'19)	71.28	80.13	69.04	77.91	64.31	73.65	62.29	71.56
+CF-EQA (Ours)	71.38(+0.10)	80.25(+0.12)	69.17(+0.13)	78.00(+0.09)	64.38(+0.07)	73.71(+0.06)	62.19(-0.10)	71.44(-0.12)

Table 2: Experimental results of debiasing methods trained on different bias subsets SQuAD$_{\text{train}}^{k=i}$ and tested on SQuAD$_{\text{dev}}$, where $i = 1, 2, 3, 4$, and ≥ 5. **BP**: Bias Product [41]; **LM**: Learned-Mixin [17].

for most methods except Learned-Mixin [17]. Taking the baseline None (BERT without debiasing) as an example. While it can get a 77.07 EM score on SQuAD$_{\text{dev}}^{k=1}$, only 7.14 and 31.20 EM scores are obtained on SQuAD$_{\text{dev}}^{k=2,3,\cdots}$ and SQuAD$_{\text{dev}}$, which verifies our early findings of the position bias problem in extractive QA. The baseline Random Position and Entropy Regularization [2] only improve the performance for a very limited margin. On the contrary, ensemble-based methods Bias Product [41] and Learned-Mixin [17] can alleviate the bias problem to a certain extent (18.73% and 42.72% F1 scores for Bias Product [41] and 78.65% and 81.14% F1 scores for Learned-Mixin [17] on SQuAD$_{\text{dev}}^{k=2,3,\cdots}$ and SQuAD$_{\text{dev}}$, respectively), especially when equipped with our CF-EQA framework. With the help of CF-EQA, 12.87% and 1.80% F1 improvements are achieved on SQuAD$_{\text{dev}}^{k=2,3,\cdots}$, as well as 6.59% and 1.10% on SQuAD$_{\text{dev}}$.

We also notice that CF-EQA is not very compatible with Bias Product [41] on SQuAD$_{\text{dev}}^{k=1}$. A possible reason for this is that in this case, the bias problem is not obvious anymore since both the training set and the test set are biased toward the first sentence. In other words, the causal effect of position bias becomes less effective and cannot be easily caught by CF-EQA.

Results on SQuAD$_{\text{train}}$. Results of cases without position bias are shown in the bottom block of Table 1. In this scenario, the baseline get competitive or slightly better performance compared with the ensemble methods Bias Product [41] and Learned-Mixin [17] on SQuAD$_{\text{dev}}^{k=1}$, SQuAD$_{\text{dev}}^{k=2,3,\ldots}$ and SQuAD$_{\text{dev}}$. This is reasonable since the distribution of answer positions is more smooth now. While ensemble-based methods maintain the performance under normal cases, our CF-EQA framework can further help them get even better performance in all settings.

Results on SQuAD$_{\text{train}}^{k=i}$. For the purpose of generalization, we also conduct experiments for biased training at different answer positions. Specifically, models are trained under SQuAD$_{\text{train}}^{k=2}$, SQuAD$_{\text{train}}^{k=3}$, SQuAD$_{\text{train}}^{k=4}$, and SQuAD$_{\text{train}}^{k=5,6,\ldots}$, with different amount of training samples and then tested on the full test set SQuAD$_{\text{dev}}$. Note that these bias subsets have much less samples than SQuAD$_{\text{train}}^{k=1}$. As we can see from Table 2, the baseline model cannot generalize well on unseen cases and thus is trapped in position bias under all four training settings. Compared with 81.32% on SQuAD$_{\text{train}}$, it only gets 43.14%, 52.89%, 58.01%, and 66.69% F1 scores with these four cases, respectively. Bias Product [41] and Learned-Mixin [17] ease the bias problem to a certain extent. While the former has boosted the performance by about 6% on F1, the latter has significantly improved for on average 18% on F1 under these four training settings. Evidence shows that our proposed CF-EQA framework can help both Bias Product [41] and Learned-Mixin [17] to achieve new state-of-the-art results.

6 Analysis

The "trade-off" parameter α plays an important role in weighting the direct and indirect effects in CF-EQA. To qualitatively analyze the impact of α, we experiment with the ensemble-based methods Bias Product [41] and Learned-Mixin [17] with training set SQuAD$_{\text{train}}^{k=1}$ and test set SQuAD$_{\text{dev}}$. Specifically, we evaluate α from 0.9 to 3.0, which are conventionally adopted values in other causal experiments. The results are shown in Figure 4.

For Bias Product [41], we notice that with the growth of α, both F1 and EM tend to have a continuous upward trend, which is rising rapidly before 2.0 and saturating around 3.0. Learned-Mixin [17] has a

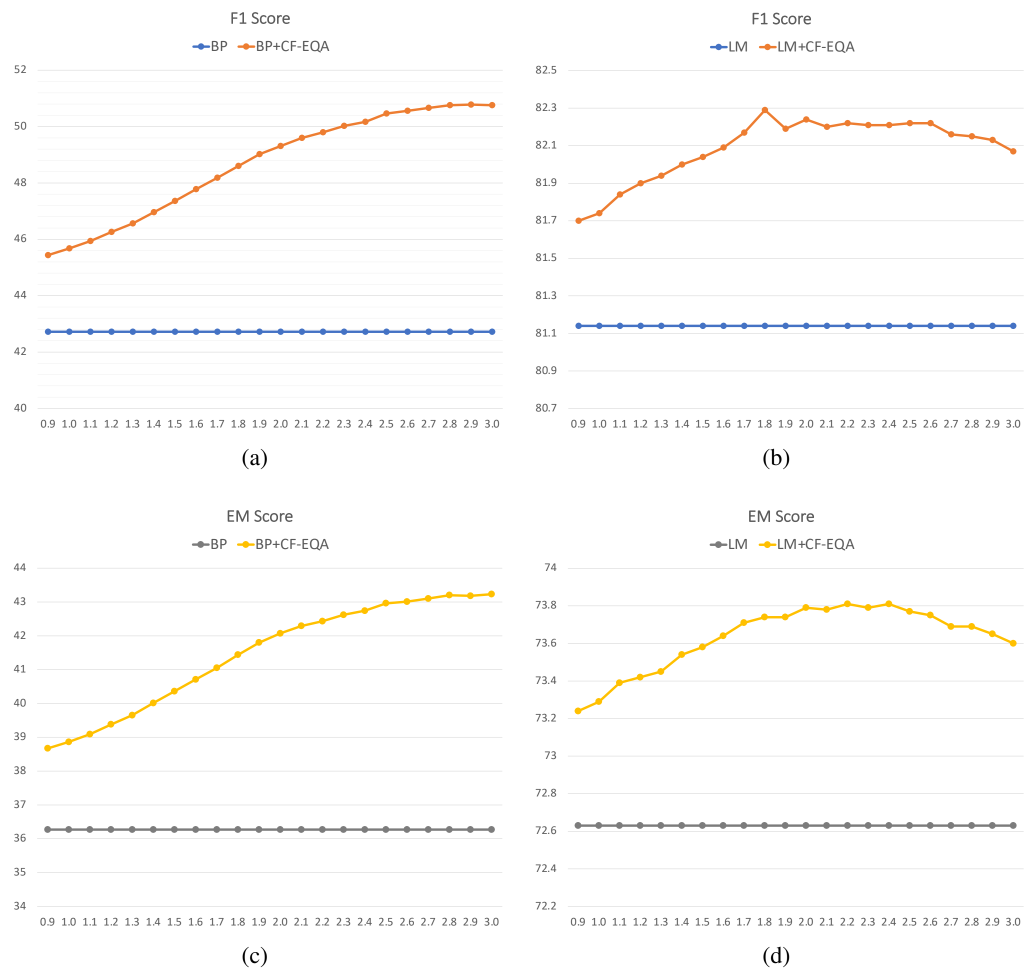


Figure 4: Qualitative results for the "trade-off" parameter α in CF-EQA. **Horizontal axis**: Different values of α (ranging from 0.9 to 3.0); **Vertical axis**: Evaluation scores (F1/EM). Models are trained on $\text{SQuAD}_{\text{train}}^{k=1}$ and tested on $\text{SQuAD}_{\text{dev}}$. (a) F1 score for Bias Product [41] and CF-EQA; (b) F1 score for Learned-Mixin [17] and CF-EQA; (c) EM score for Bias Product [41] and CF-EQA; (d) EM score for Learned-Mixin [17] and CF-EQA.

"crest" trend on F1 and EM, where the scores correspond to around 2.0 reach the top, and then the performance goes down. In general, for cases like position bias towards the first sentence, setting α to 2.0 will be a good choice.

7 Conclusion

In this project, we proposed a novel counterfactual framework called CF-EQA to reduce the position bias of the answers in the extractive QA task. The bias is formulated as the indirect causal effect of word positions on answers, and is further captured by a counterfactual scenario. The reduction of this bias is realized by subtracting the indirect effect from the total causal effect. Experimental results fully demonstrate the effectiveness and generalizability of our proposed unbiased inference strategy.

References

[1] Rajpurkar Pranav, Zhang Jian, Lopyrev Konstantin, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[2] Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. Look at the first sentence: Position bias in question answering. *arXiv preprint arXiv:2004.14602*, 2020.

[3] 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*, pages 4171–4186, 2019.

[4] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in Neural Information Processing Systems*, 32:841–852, 2019.

[5] Caiming Xiong, Victor Zhong, and Richard Socher. Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*, 2017.

[6] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.

[7] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[8] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *arXiv preprint arXiv:1506.03134*, 2015.

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[11] Bhuwan Dhingra, Danish Pruthi, and Dheeraj Rajagopal. Simple and effective semi-supervised question answering. *arXiv preprint arXiv:1804.00720*, 2018.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

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

[15] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10800–10809, 2020.

[16] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10044–10054, 2020.

[17] 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 4060–4073, 2019.

[18] 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.

[19] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[20] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279, 2018.

[21] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8990, 2020.

[22] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

[23] Long Chen, Hanwang Zhang, Jun Xiao, Xiangnan He, Shiliang Pu, and Shih-Fu Chang. Counterfactual critic multi-agent training for scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4613–4623, 2019.

[24] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725, 2020.

[25] Zhiyuan Fang, Shu Kong, Charless Fowlkes, and Yezhou Yang. Modularized textual grounding for counterfactual resilience. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6378–6388, 2019.

[26] Atsushi Kanehira, Kentaro Takemoto, Sho Inayoshi, and Tatsuya Harada. Multimodal explanations by predicting counterfactuality in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8594–8602, 2019.

[27] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. *arXiv preprint arXiv:2009.13000*, 2020.

[28] Zhongqi Yue, Tan Wang, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Counterfactual zero-shot and open-set visual recognition. *arXiv preprint arXiv:2103.00887*, 2021.

[29] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. *arXiv preprint arXiv:2103.01737*, 2021.

[30] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10760–10770, 2020.

[31] Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. Devlbert: Learning deconfounded visio-linguistic representations. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4373–4382, 2020.

[32] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2009.12547*, 2020.

[33] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. *arXiv preprint arXiv:2004.09034*, 2020.

[34] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10860–10869, 2020.

[35] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020.

[36] Tsu-Jui Fu, Xin Wang, Scott Grafton, Miguel Eckstein, and William Yang Wang. Iterative language-based image editing via self-supervised counterfactual reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4413–4422, 2020.

[37] Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. *arXiv preprint arXiv:2103.03493*, 2021.

[38] Tyler Vanderweele. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)*, 24, 01 2013.

[39] Judea Pearl. Direct and indirect effects, 2013.

[40] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *arXiv preprint arXiv:2009.12991*, 2020.

[41] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 2002.

[42] P. Kingma Diederik and Ba Jimmy. Adam: A method for stochastic optimization. *Proceedings of the 2015 International Conference on Learning Representations*, 2015.