None the wiser? Adding "None" Mitigates Superficial Cues in Multiple-Choice Benchmarks

Pride Kavumba[1,2] Ana Brassard[2,1] Benjamin Heinzerling[2,1]
Naoya Inoue[1,2,*] Kentaro Inui[1,2]
[1]Tohoku University [2]RIKEN Center for Advanced Intelligence Project (AIP)
{pkavumba, naoya-i, inui} @ecei.tohoku.ac.jp
{ana.brassard, benjamin.heinzerling} @riken.jp

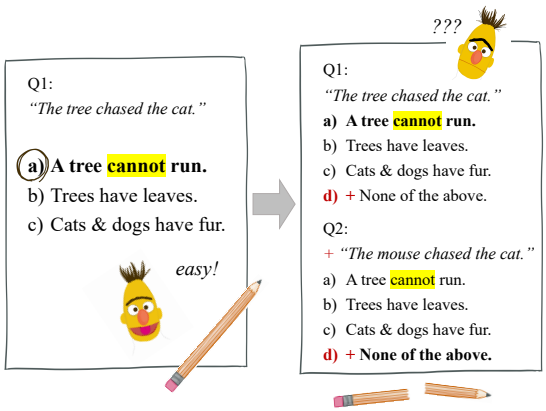
1 Introduction

It is now established that many benchmarks of natural language understanding contain superficial cues, which enable models to solve the task using shortcuts that do not generalize to datasets without these shortcuts [3, 13, 7, 8, 14, 4].

Previous works have employed two main approaches to mitigate superficial cues. The first approach is to discard instances that contain superficial cues using language models (LMs). For example, Zellers et al. [15] create SWAG using adversarial filtering which removes instances that can be easily solved by existing finetuned LMs. Zellers et al. [16] then created HellaSwag, a more challenging dataset than SWAG [15], using adversarial filtering with BERT [2]. However, this approach may reduce question diversity, as Gururangan et al. [3] notes on SNLI, and creates distributional bias [16] that can be exploited by other models not used in the filtering process. SWAG has been shown to contain superficial cues [14], and our analysis reveals superficial cues in HellaSwag: ALBERT [5] trained on only the answers achieved 76.2% accuracy.

The second approach is to efficiently reuse existing benchmarks and to change the training process of models. For example, Poliak et al. [9] use adversarial training to unlearn hypothesis-only bias on SNLI [1] and Stacey et al. [12] extends this approach by using an ensemble of adversaries. Schuster et al. [11] propose to weigh training instances based on the difficulty in the training objective. However, this approach requires solving additional optimization problems.

Here, we introduce AddNone—an automatic method that reuses the full existing dataset, does not rely on LMs, and does not change the training process—for mitigating superficial cues in multiple-choice benchmarks. Specifically, for each multiple-choice question, consisting of a context and multiple choices (Fig. 1, A), we introduce "*None of the above*" (Fig. 1, B, Q1). We then duplicate



A) Original question B) Add "none of the above"

Figure 1: A question containing a superficial cue (highlighted) that allows models to easily pick the correct answer (**bold**) (A). AddNone adds *None of the above* (B, Q1) and automatically generates a twin question with the same choices but a different context that makes *None of the above* the correct answer (B, Q2). The superficial cue is now less effective since it appears both as correct and incorrect.

the question and change the correct answer to "*None of the above*" by replacing the context (Fig. 1, B, Q2). This creates twin questions with the same choices but different contexts and correct answer, forcing models to consider the context while solving them. Our approach does not discard any instances, and modifying the data is computationally cheap and fully automated. Additionally, we do not have to modify the models and their training process except having the extra choice.

In summary, our contributions are as follows:

- We show that the answers in Commonsense-Explanation (Cs-Ex), HellaSwag and SWAG, commonly used commonsense benchmarks, contain superficial cues (§2), and that training on these datasets leads to poor generalization on datasets without superficial cues (§4);
- We present AddNone, a method for mitigating superficial cues in multiple-choice benchmarks, and apply it to these benchmarks (§3);
- We empirically show that training on AddNone-modified datasets encourages models to assess the association be-

* now Stony Brook University

tween a context and its choices, leading to better generalization on dataset without superficial cues (§4).

2 Cues in Cs-Ex, HellaSwag, and SWAG

In this section, we show that commonly used commonsense datasets, Cs-Ex, HellaSwag and SWAG, contain easy-to-exploit superficial cues.[1] All three benchmarks are multiple-choice tasks in which the model is required to choose the correct ending for a given incomplete sentence (SWAG, HellaSwag) or the reason why a given statement is false (Cs-Ex). The authors of Cs-Ex aimed to mitigate superficial cues by employing strict guidelines for crowd workers, while the authors of SWAG and HellaSwag made use of LM-based adversarial filtering. SWAG and HellaSwag should not be solvable by an LM in an answer-only setting, namely, by an LSTM-LM trained on the Book Corpus for SWAG, and by BERT for HellaSwag.

2.1 Input Ablation

We first ablate the input, i.e., we provide models with contextless questions (answers only). This setup follows similar ablations by Gururangan et al. [3], McCoy et al. [7] and is designed to reveal whether the answers contain superficial cues that allow models to solve the task by taking shortcuts, such as relying on different token distributions in correct and wrong answers.

We finetune BERT, RoBERTa [6], and ALBERT in this contextless setting. The high accuracy of BERT (87.8%), RoBERTa (85.3%) and ALBERT(85.7%) on Cs-Ex shows that the strict crowdsourcing protocol for creating this dataset was not effective, since even without the context, models are still able to identify the correct answer considerably above random chance (Table 1). On SWAG, model accuracies are similarly high, while on HellaSwag, the effect of using BERT for adversarial filtering is clearly visible in BERT's accuracy (37.0%). However, this filtering is not effective for RoBERTa (70.5%) and ALBERT (76.2%).[2] These results show that the answers of Cs-Ex, HellaSwag, and SWAG all contain exploitable superficial cues.

[1] Trichelair et al. [14] showed, using a different analysis method than the one presented here, that SWAG contains superficial cues. We include this dataset for completeness.

[2] When creating HellaSwag, the authors envisioned a co-evolution of models and datasets, in which increasingly stronger models are used as filters to create increasingly harder datasets, which, in turn, are used to train increasingly stronger models. However, given RoBERTa's worse accuracy on Cs-Ex, the question arises whether RoBERTa's strong results on answers-only HellaSwag are due to being *stronger* than the filter (i.e., BERT) or merely *different* than the filter.

Model	Cs-Ex	HellaSwag	SWAG
Random	33.3	25.0	25.0
ALBERT	$85.7_{\pm 0.7}$	$76.2_{\pm 0.1}$	$81.1_{\pm 0.1}$
RoBERTa	$85.3_{\pm 0.4}$	$70.5_{\pm 0.4}$	$79.1_{\pm 0.2}$
BERT	$87.8_{\pm 0.3}$	$36.8_{\pm 0.5}$	$73.1_{\pm 0.3}$

Table 1: Average accuracy with standard deviation (subscript) of models trained on the answers only.

2.2 Token-based Superficial Cues

To identify the actual superficial cues models may exploit, we collect unigrams that are predictive of the answer, using the *productivity* measure introduced by Niven and Kao [8, see definition in]. Intuitively, the productivity of a token expresses how precise a model is if it predicts only based on the presence of this token in a candidate answer. We found that *not* is highly predictive of the correct answer on Cs-Ex, followed by *to*. On SWAG and HellaSwag, unigram token productivity is below 20, suggesting that RoBERTa exploits different signals to achieve high accuracy in the answers-only setting. Bigram token productivity was much lower.

To further investigate the exploitability of unigram cues, we train a binary bag-of-word classifier to predict if a given choice is a correct or wrong answer. On Cs-Ex, this classifier achieved 89.5% accuracy, showing that correct and wrong answers are clearly distinguishable and confirming that the task is solvable with token-level cues. On SWAG and HellaSwag, the classifier fails to reach majority accuracy. This further confirms that superficial cues in SWAG and HellaSwag are not token-based.

3 AddNone to Mitigate Superficial Cues

To mitigate superficial cues in the answers, we ensure that each correct answer appears at least once as a wrong answer. This breaks the direct link between the cues and the correctness of the answer, since they now also point to the wrong answer at least once. To achieve this, one can duplicate each question and manually modify it so that an alternative choice becomes correct (henceforth, *twin question*) [8, 4], however, this does not scale to larger datasets.

As a scalable alternative, we propose creating twin questions by adding *none of the above* as an additional choice in all questions, then automatically replacing one of the contexts so that the added choice becomes the correct answer. This is similar to previous methods [8, 4], however, creating a context that does not fit *any answer* can be automated while curating one that specifically leads to an alternative answer requires creativity, i.e., a manual effort.

For example, consider the following question from Cs-Ex with the additional choice included.

Dataset	Easy	Hard	Total
Cs-Ex	1,896	125	2,021
HellaSwag	8,469	1,573	10,042
SWAG	18,217	1,789	20,006

Table 2: Number of Easy and Hard instances

Context: she danced to the piano
a) piano can not be danced to
b) piano can produce beautiful sounds
c) some people are born to dance
+ **d)** None of the above

This question stays unchanged, with only an added choice that is not correct in this case. Its twin, shown below, has its context replaced by a similar context that renders the correct answer (in bold above) incorrect.

Context: he types using a piano
a) piano can not be danced to
b) piano can produce beautiful sounds
c) some people are born to dance
+ **d) None of the above**

The replacement context is extracted from the set of all training contexts in the dataset. Candidate contexts should be similar enough to the original context to avoid trivial solving by (dis)similarity. In our experiments, we calculated the cosine similarity between TF-IDF vectors of all training contexts and the context to be replaced, and picked the most similar context, allowing the similarity to be 0.97 or below. Theoretically, it is possible that the replacement context is a paraphrase of the original context, however, in practice, we did not find this to be a problem. We used 90% of the original questions to create twin questions.

4 Experiments

4.1 Evaluation on Hard and Easy Subsets

To investigate if ALBERT, RoBERTa, and BERT learn to rely on superficial cues when trained on the original Cs-Ex, HellaSwag, and SWAG dataset, we split the evaluation set into (i) questions with superficial cues, *Easy subset*, and (ii) questions *without* superficial cues, *Hard subset*. The Easy subset consists of instances that are correctly solved by ALBERT or RoBERTa over three random seeds without being provided the context,[3] and the Hard subset consists of the remaining instances. The statistics of each subset are shown in Table 2.

The models trained on the original dataset perform badly on the Hard subset, while they perform much better on the Easy subset (Fig. 2). This indicates that the models strongly rely on superficial cues and the reported accuracy

[3]We do not use BERT because it was used for adversarial filtering of HellaSwag.

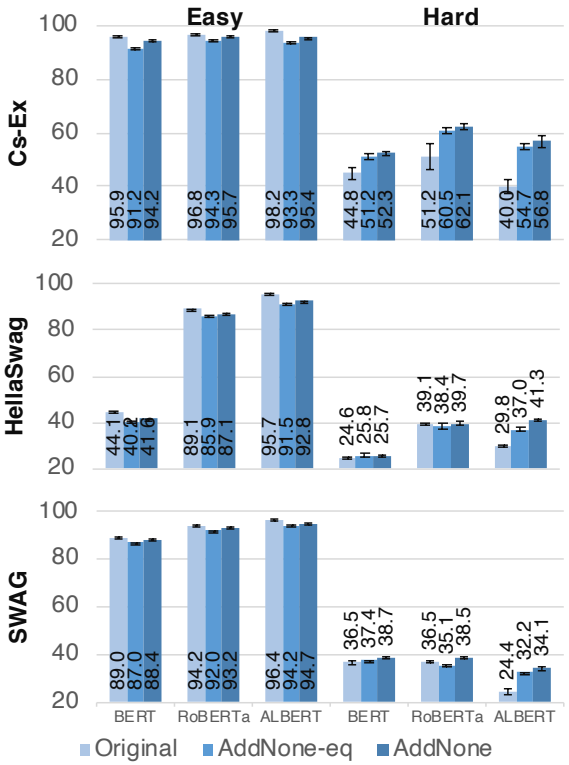


Figure 2: Average accuracy and standard deviation of models on Easy questions with superficial cues and Hard questions without superficial cues. On Hard questions, models trained on AddNone, and AddNone-eq (which is equal in size to the original dataset) perform better.

on these datasets may be inflated.

4.2 Evaluation of AddNone

How effective is AddNone in removing superficial cues, and how does it translate to model performance? We train ALBERT, RoBERTa, and BERT on the *AddNone-modified* Cs-Ex, HellaSwag, and SWAG (*AddNone models*), and evaluate them on the *original* Hard and Easy subsets.

As expected, training on the AddNone-modified datasets degraded the accuracy on the Easy subset containing superficial cues across all models and benchmarks (Fig. 2). This indicates that the models were discouraged from exploiting superficial cues.

On the other hand, AddNone-modified datasets improved the accuracy on the Hard subset lacking superficial cues. This suggests that the models were encouraged to learn more task-related cues, which lead to better generalization resulting in higher performance on the test set.

To control for data size, we repeated the experiments on AddNone-modified datasets that are equal in size to their originals (AddNone-eq) and found that the results still support the effectiveness of AddNone (Fig. 2). Rajpurkar et al. [10] proposed crowdsourcing unanswerable questions instead of generating them automatically, however, here we found that creating these automatically improves the model generalization.

Dataset	Model	Original		AddNone	
		C	NC	C	NC
Cs-Ex	BERT	92.7	75.4	91.6	50.5
	RoBERTa	93.9	76.0	93.7	61.3
	ALBERT	94.6	60.0	93.0	43.2
HellaSwag	BERT	41.1	33.1	39.1	30.6
	RoBERTa	81.3	65.6	79.7	61.5
	ALBERT	85.4	61.4	84.7	53.3
SWAG	BERT	84.3	67.6	83.9	65.7
	RoBERTa	89.0	75.5	88.3	74.0
	ALBERT	89.9	75.8	89.3	73.4

Table 3: Overall average accuracy of full-input models evaluated on context (C) and contextless (NC) setting. The larger drop of accuracy in AddNone-modified datasets indicates that models trained on original dataset rely more on superficial cues in the answers.

4.3 Model Sensitivity to Context

Are the models trained on AddNone-modified datasets more sensitive to given contexts? We compare AddNone models with the models trained on the original datasets (*original models*) on the test set with context (C) and no context (NC). Here, we use the same models whose results are shown in Fig. 2. AddNone models are more sensitive to the context than the original models across all datasets, indicating that AddNone models' predictions depended more on the connection between the context and the choices (Table 3).

5 Conclusions

We introduced AddNone, a simple and LM-independent method for mitigating superficial cues in multiple-choice benchmarks. AddNone can reuse existing benchmarks, allowing researchers to make more efficient use of the existing benchmarks instead of creating new ones. Our experiments demonstrated that training on AddNone-modified datasets encourages models to assess the association between a context and its choices, leading to better generalization on datasets without superficial cues.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19H04425.

References

[1] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. 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://www.aclweb.org/anthology/N19-1423.

[3] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. 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.

[4] P. Kavumba, N. Inoue, B. Heinzerling, K. Singh, P. Reisert, and K. Inui. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6004. URL https://www.aclweb.org/anthology/D19-6004.

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

[7] T. McCoy, E. Pavlick, and T. 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. URL https://www.aclweb.org/anthology/P19-1334.

[8] T. Niven and H.-Y. Kao. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1459.

[9] A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. 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://www.aclweb.org/anthology/S18-2023.

[10] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124.

[11] T. Schuster, D. Shah, Y. J. S. Yeo, D. Roberto Filizzola Ortiz, E. Santus, and R. 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 3410–3416, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1341. URL https://www.aclweb.org/anthology/D19-1341.

[12] J. Stacey, P. Minervini, H. Dubossarsky, S. Riedel, and T. Rocktäschel. There is strength in numbers: Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training, 2020.

[13] S. Sugawara, K. Inui, S. Sekine, and A. 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, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1453. URL https://www.aclweb.org/anthology/D18-1453.

[14] P. Trichelair, A. Emami, A. Trischler, K. Suleman, and J. C. K. Cheung. How reasonable are common-sense reasoning tasks: A case-study on the Winograd schema challenge and SWAG. 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 3373–3378, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1335. URL https://www.aclweb.org/anthology/D19-1335.

[15] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1009. URL https://www.aclweb.org/anthology/D18-1009.

[16] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. 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://www.aclweb.org/anthology/P19-1472.