

More Bang for Your Buck: Natural Perturbation for Robust Question Answering

Daniel Khashabi and Tushar Khot and Ashish Sabharwal

Allen Institute for AI, Seattle, WA, U.S.A.

{danielk,tushark,ashishs}@allenai.org

Abstract

Deep learning models for linguistic tasks require large training datasets, which are expensive to create. As an alternative to the traditional approach of creating new instances by repeating the process of creating one instance, we propose doing so by first collecting a set of seed examples and then applying human-driven *natural perturbations* (as opposed to rule-based machine perturbations), which often change the gold label as well. Such perturbations have the advantage of being relatively easier (and hence cheaper) to create than writing out completely new examples. Further, they help address the issue that even models achieving human-level scores on NLP datasets are known to be considerably sensitive to small changes in input. To evaluate the idea, we consider a recent question-answering dataset (BOOLQ) and study our approach as a function of the *perturbation cost ratio*, the relative cost of perturbing an existing question vs. creating a new one from scratch. We find that when natural perturbations are moderately cheaper to create (cost ratio under 60%), it is more effective to use them for training BOOLQ models: such models exhibit 9% higher robustness and 4.5% stronger generalization, while retaining performance on the original BOOLQ dataset.

1 Introduction

Creating large datasets to train NLP models has become increasingly expensive. While many datasets (Bowman et al., 2015; Rajpurkar et al., 2016) targeting different linguistic tasks have been proposed, nearly all are created by repeating a fixed process used for writing a single example. This approach results in many *independent* examples, each generated from scratch. We propose an alternative, often substantially cheaper training set construction method where, after collecting a few seed examples, the set is expanded by applying human-authored *minimal perturbations* to the seeds.

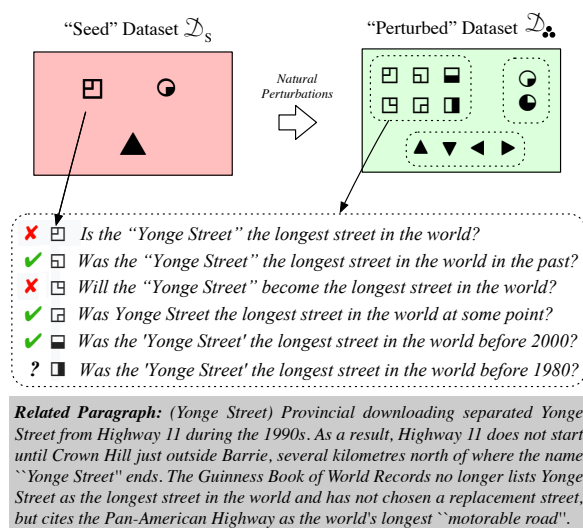


Figure 1: Training set creation via minimal-perturbation clusters. Left: Seed dataset \mathcal{D}_S with 3 instances (shown as different shapes). Right: Expanded dataset \mathcal{D}_\bullet with 10 instances, comprising 2-4 minimal-perturbations (illustrated as rotation, fills, etc.) of each seed instance. Human-authored perturbations aren’t required to always preserve the answer (yes/no in the example) and often add richness by altering the answer.

Fig. 1 illustrates our proposal of using *natural perturbations*. We use the traditional approach to first create a small scale *seed dataset* \mathcal{D}_S , shown as the red rectangle on the left with three instances (denoted by different shapes). However, rather than simply repeating this process to scale up \mathcal{D}_S to a larger dataset \mathcal{D} , we set up a different task: ask crowdworkers to create multiple *minimal perturbations* of each seed instance (shown as rotation, fills, etc.) with an encouragement to change the answer. The end result is a larger dataset \mathcal{D}_\bullet of a similar size as \mathcal{D} but with an inherent structure: *clusters* of *minimally-perturbed* instances with mixed labels, denoted by the green rectangle at the right in Fig. 1.

An inspiration for our approach is the lack of robustness of current state-of-the-art models to mi-

nor adversarial changes in the input (Jia and Liang, 2017). We observed a similar phenomenon even with model-agnostic, human-authored changes to yes/no questions (as shown in Fig.1), despite models achieving near-human performance on this task. Specifically, we found the accuracy of a ROBERTA model trained on BOOLQ (Clark et al., 2019) to drop by 15% when evaluated on locally perturbed questions. These new questions were, however, no harder for humans. This raises the question: *Can a different way of constructing training sets help alleviate this issue?* Minimal perturbations, as we show, provide an affirmative answer.

Perturbing a given example is generally a simpler task, costing only a fraction of the cost of creating a new example from scratch. We call this fraction the *perturbation cost ratio* (henceforth referred to as *cost ratio*), and assess the value of our perturbed training datasets as a function of it. As this ratio decreases (i.e., perturbations become cheaper), one, of course, obtains a larger dataset than the traditional method, at the same cost. More importantly, even when the ratio is only moderately low (at 0.6), models trained on our perturbed datasets exhibit desirable advantages: They are 9% more robust to minor changes and generalize 4.5% better across datasets than models trained on BOOLQ.

Specifically, our *generalization experiment* with the MULTIRC (Khashabi et al., 2018) dataset demonstrates that models trained on perturbed data outperform those trained on traditional data when evaluated on unseen, unperturbed questions from a different domain. Second, we *assess robustness* by evaluating on BOOLQ-*e* (Gardner et al., 2020), a test set of expert-generated perturbations that deviate from the patterns common in large-scale crowd-sourced perturbations. Our zero-shot results here indicate that models trained on perturbed questions go beyond simply learning to memorize particular patterns in the training data. Third, we find that training on the perturbed data, for the most part, continues to *retain performance* on the original task.

Even with the worst case cost ratio of 1.0 (when perturbing existing questions is no cheaper than writing new ones), models trained on perturbed examples remain competitive on all our evaluation sets. This is an important use case for situations that simply do not allow for sufficiently many distinct training examples (e.g., low resource settings, limited amounts of real user data, etc.). Our results

at ratio 1.0 suggest that simply applying minimal perturbations to the limited number of real examples available in these situations can be just as effective as (hypothetically) having access to large amounts of real data.

In summary, we propose a novel method to construct datasets that combines traditional independent example collection approach with minimal natural perturbations. We show that for many reasonable cases, using perturbation clusters for training can result in cost-efficiently trained high-quality, robust models that generalize across datasets.

2 Related Work

Data augmentation. There is a handful of work that studies semi-automatic contextual augmentation (Kobayashi, 2018; Cheng et al., 2018), often with the goal of creating better systems. We, however, study natural human-authored perturbations as an alternative dataset construction method. A related recent work is by Kaushik et al. (2020), who, unlike the goal here, study the value of natural-perturbations in reducing artifacts.

Adversarial perturbations. A closely relevant line of work is *adversarial perturbations* to expose the weaknesses of systems upon local changes and criticize their lack robustness (Ebrahimi et al., 2018; Glockner et al., 2018; Dinan et al., 2019). For instance, Khashabi et al. (2016) showed significant drops upon perturbing answer-options for multiple-choice question-answering. Such rule-based perturbations have simple definitions leading to them being easily reverse-engineered by models (Jia and Liang, 2017) and generally use label-preserving, shallow perturbations (Hu et al., 2019). In contrast, our natural human-authored perturbations are harder for models.¹ More broadly, adversarial perturbations research seeks examples that stumble existing models, while our focus is on expanding datasets in a cost-efficient way.

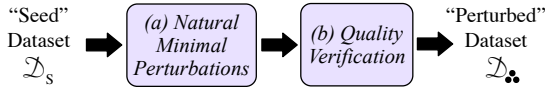
Minimal-pairs in NLP. Datasets with minimal-pair instances are relatively well-established in certain tasks, such as *Winograd schema* datasets (Levesque et al., 2011; Peng et al., 2015; Sakaguchi et al., 2020), or the recent *contrast sets* (Gardner et al., 2020). However, the importance of datasets with *pairs* (i.e., *clusters* of size two) is not well-understood. Our findings about

¹We tried the system by Hu et al. (2019) on our questions, but it resulted in very limited variations. See Appendix E.

perturbation clusters could potentially be useful for the future construction of datasets for such tasks.

3 Expansion via Perturbation Clusters

Our approach mainly differs from traditional approaches in how we expand the dataset given seed examples. Rather than repeating the process to generate more examples, we apply minimal alterations to the seed examples, in two high-level steps:



The first step generates the initial set of examples with natural perturbations. It should respect certain principles: (a) The construction should apply minimal changes (similar to the ones in Fig. 1), otherwise the resulting clusters might be too heterogeneous and less meaningful. (b) A substantial proportion of natural perturbations should *change* the answer to the questions. (c) It should incentivize creativity and diversity in local perturbations by, for instance, showing thought-provoking suggestions, using a diverse pool of annotators (Geva et al., 2019), etc. The second independent verification step ensures dataset quality by (a) getting the true gold label and (b) ensuring all generated questions are answerable given the relevant paragraph, in isolation from the original question.

BOOLQ_•: BOOLQ Expansion. We obtain D_S by sampling questions from BOOLQ (Clark et al., 2019), which is a QA dataset where each boolean (“yes”/“no” answer) question could be inferred from an associated passage. We then follow the above two-step process, resulting in BOOLQ_•, a naturally perturbed dataset with $17k$ questions derived from $4k$ seed questions:

a) minimal perturbations: Crowdworkers are given a question and its corresponding gold answer based on supporting paragraph. Then the workers are asked to change the question in order to *flip the answer to the question*. While making changes, the workers are guided to keep their changes minimal (adding or removing up to 4 terms) while resulting in proper English questions. Additionally, for each seed question, crowd-workers are asked to generate perturbations till the modified question is challenging for a machine solver (i.e., ROBERTA trained on BOOLQ, should have low confidence on the correct answer). Note that we do not require the model to answer the question incorrectly and not all ques-

tions are challenging for the model. Our main goal here is to encourage interesting questions by using the trained model as the guide.

b) question verification. Given the perturbed questions, we asked multiple annotators to answer these questions. These annotations served to eliminate ambiguous questions as well as those that cannot be answered from the provided paragraph. The annotation was done in two steps: (i) in the first step, we ask 3 workers to answer each question with one of the three options (“yes“, “no“ and “cannot be inferred from the paragraph”). We filtered out the subset of the questions that were not agreed upon (i.e., not a consistent majority label) or were marked as “cannot be inferred from the paragraph” by majority of the annotators. To speed up the annotations, the annotation were done on a cluster-level, i.e., annotators could see all the different modified questions corresponding to a paragraph. (ii) subsequently, each modified questions is also annotated *individually* to ensure that questions can be answered in isolation (as opposed to answering them while seeing all the questions in a cluster.) The annotations in this step only have two labels (“yes”/“no”) and again questions that were not agreed upon were filtered.

Sample questions generated by our process are shown in Fig. 1. We evaluate the impact of perturbations via this dataset.

Dataset subsampling. We sample questions from this expanded dataset to evaluate the value of perturbations as a function of different parameters. To simplify exposition, we will use the following notation. We assume a fixed budget b for constructing the dataset where each new question costs 1 unit, i.e., traditional methods would construct a dataset of size b in the given budget. The perturbation cost ratio $r \leq 1$ is the cost of creating a perturbed question. When $r \approx 1$, perturbations are equally costly as writing out new instances. If $r \ll 1$, perturbations are cheap. For instance, if $r = 0.5$, each hand-written question costs the same as two perturbed questions.

We denote the total number of instances and clusters with N, C , respectively. We use $\text{BOOLQ}_{\bullet}^{b,c,r}$ to denote the largest subset of BOOLQ_{\bullet} that can be generated with a total budget of b , with a *maximum* cluster size of c , and relative cost ratio of r . In the special case where all clusters are of the exact same size c , these parameters are related as follows:

$$b = (1 + (c - 1)r) \times C,$$

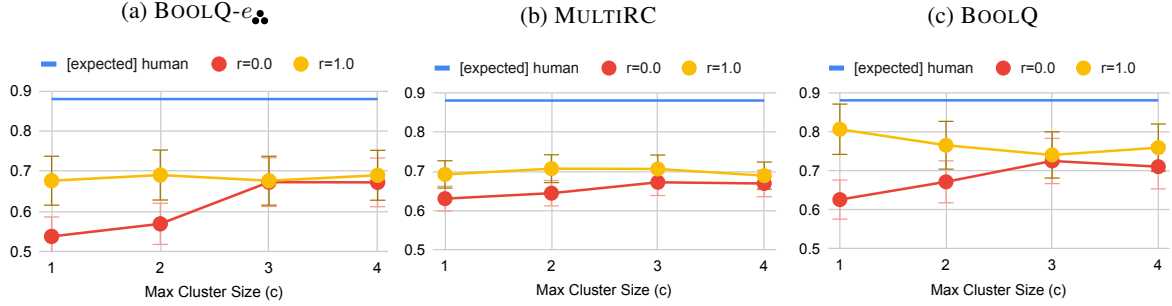


Figure 2: Model accuracy (y -axis) with a fixed budget b and varying cluster size c (x -axis), for two extreme cases: (i) $r = 1.0$ denoting a fixed total number of *questions*, (ii) $r = 0.0$ denoting a fixed total number of *clusters*. The plots indicate that including additional perturbations in each cluster (going left to right), particularly when they are cheap (closer to the $r = 0$ case), adds value to the dataset by increasing model accuracy.

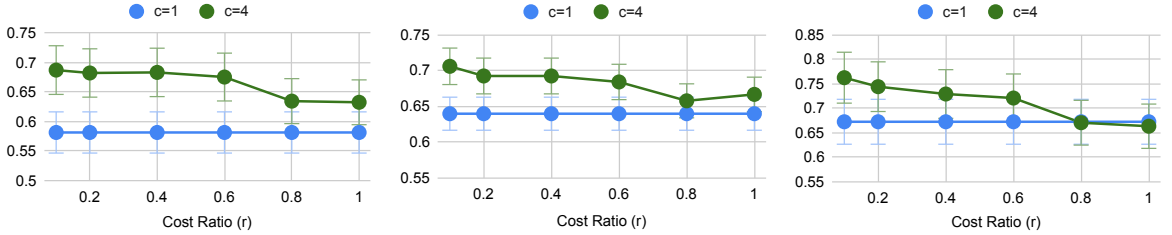


Figure 3: Model accuracy (y -axis) with a fixed total budget b and varying cost ratio r (x -axis), in two cases: (i) $c = 1$ denoting singleton clusters (standard approach), (ii) $c = 4$ denoting cluster size 4 (our approach). The smaller the cost, the higher the returns. For a moderately low cost ratio such as 0.6, the model trained on our perturbed datasets gains of 3-5% over a model trained on the traditionally constructed dataset.

where $1 + (c - 1)r$ is the cost of a single cluster calculated as the cost of one seed examples and its $c - 1$ perturbations.

To create $\text{BOOLQ}_{b,c,r}$ we subsample a maximum of c questions from each perturbation cluster, such that total number of clusters is no more than $\frac{b}{1+(c-1)r}$ and the ratio of “yes” to “no” questions is 0.55. Our subsampling starts with clusters of size at least c and also considers smaller clusters if necessary. $\text{BOOLQ}_{b,1,r}$ (singleton clusters) corresponds to a dataset constructed in a similar fashion to BOOLQ , whereas $\text{BOOLQ}_{b,4,r}$ (big clusters) roughly corresponds to the BOOLQ_{\bullet} dataset.

4 Experiments

To assess the impact of our perturbation approach, we evaluate standard RoBERTa-large model that has been shown to achieve state-of-the-art results on many tasks. Each experiment considers the effect of training on subsamples of BOOLQ_{\bullet} obtained under different conditions.

Each of the points in the figures are averaged over 5 random subsampling of the dataset (with error bars to indicate the standard deviation). The Appendix includes further details about the setup as well as additional experiments.

We evaluate the QA model trained on various question sets on three test sets. (i) For assessing **robustness**, we use an expert-generated set $\text{BOOLQ}_{e_{\bullet}}$ published in Gardner et al. (2020) with 339 high-quality perturbed questions based on BOOLQ . (ii) For assessing **generalization**, we use the subset of 260 training questions from MULTIRC (Khashabi et al., 2018) that have binary (yes/no) answers, from training section of the their data.² (c) The original BOOLQ test set, to ensure models trained on perturbed questions also **retain performance** on the original task.

4.1 Effect of Cluster Size (c)

We study the value of clusters sizes in the perturbations in two extreme cases: (i) when perturbations cost the same as new questions ($r = 1.0$) and the only limit is the our overall budget ($b = 3.7k$), and (ii) when the perturbations cost negligible ($r = 0.0$) but we are limited by the max cluster size c and $b = 1k$.³ For each case, we vary the max cluster size in the following range: $[1, 2, 3, 4]$. As a result, in (i), C vary from $3.7k$ to 951 ($N = 3.7k$), and in (ii), N vary from $1k$ to $4k$ ($C = 1k$).

²The yes/no subset of dev was too small.

³In practice, we expect r to lie somewhere in-between these two extremes, such as $r = 0.3$ as discussed in §4.2.

Fig. 2 shows the accuracy of models trained on these subsets across our three evaluation sets. In **scenario (i)** with a fixed number of instances ($r = 1$), it is evident that the size of the clusters (the number of perturbations) does not affect the model quality (on 2 out of 3 datasets). This shows that perturbation clusters are equally informative as (traditional) independent instances. However, in **scenario (ii)** with a fixed number of clusters ($r = 0$), the system performance consistently gets higher with larger clusters, even though the number of clusters is kept constant. This indicates that each additional perturbation adds value to the existing ones, especially in terms of model robustness and retaining performance on the original task.

4.2 Effect of Perturbations Cost Ratio (r)

We now study the value of perturbations as a function of their cost (r). We vary this parameter within the range $(0, 1]$ for $b = 1.5k$ and two max clusters sizes, $c = \{1, 4\}$. When $c = 1$ (no perturbations), N stays constant at $1.5k$. When $c = 4$, N varies from $4.6k$ to $1.5k$. Fig. 3 presents the accuracy of our model as a function of r .

While we don't know the exact crowdsourcing cost for BOOLQ, a typical question writing task might cost USD 0.60 per question. With our perturbed dataset costing USD 0.20 per question, we have $r = 0.33$. Given the same total budget $b = 1500$, we can thus infer from Fig. 3 that training on a dataset of perturbed questions would be about 10% and 5% more effective on BOOLQ-*e* and MULTIRC, respectively.

The result on all datasets indicates that there is value in using perturbations clusters when $r \leq 0.6$, i.e., larger clusters can be more cost-effective for build better training sets. Even when they are not much cheaper, they retain the same performance as independent examples, making them a good alternative for dataset expansion given few sources of examples (e.g., low resource languages).

5 Discussion

A key question with respect to the premise of this work is whether the idea would generalize to other tasks. Here, we chose yes/no questions since this is the least-explored sub-area of QA (compared to extractive QA, for example) and hence could benefit from more efficient dataset construction. We (the authors) are cautiously optimistic that it would, although that is subject to factors such as the rel-

ative cost of creating diverse and challenging perturbations. Concurrent works have also explored a similar construction for other tasks but with different purposes (Gardner et al., 2020; Kaushik et al., 2020).

We note that we assume a typical QA dataset construction process where workers write questions based on given fixed contexts (Rajpurkar et al., 2016). This assumption may not always hold for alternative dataset generation pipelines, such as using an already available set of questions (Kwiatkowski et al., 2019). Even in such cases, one can still use the lessons learned here to apply *natural perturbations* to a different stage in the annotation pipeline to make it more cost efficient.

6 Conclusion

We proposed an alternative approach for constructing training sets, by expanding seed examples via natural perturbations. Our results demonstrate that models trained on perturbations of BOOLQ questions are more robust to minor variations and generalize better, while preserving performance on the original BOOLQ test set as long as the natural perturbations are moderately cheap to create. Creating perturbed examples is often cheaper than creating new ones and we empirically observed notable gains even at a moderate cost ratio of 0.6.

While this is *not* a dataset paper (since our focus is on more on the value of natural perturbations for robust model design), we provide the natural perturbations resource for BOOLQ constructed during the course of this study.⁴

This work suggests a number of interesting lines of future investigation. For instance, how do the results change as a function of the total dataset budget b or large values of c ? Over-generation of perturbations can result in overly-similar (less-informative) variations of a seed example, making larger clusters valuable only up to a certain extent. While we leave a detailed study to future work, we expect general trends regarding the value of perturbations to hold broadly.

Acknowledgments

The authors would like to thank Chandra Bhagavatula, Sihao Chen, and Niket Tandon for discussions and suggestions on early versions of this work, and anonymous Amazon Mechanical Turk workers for their help with annotation.

⁴<https://github.com/allenai/natural-perturbations>

References

- S. Bowman, G. Angeli, C. Potts, and C. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.
- Y. Cheng, Z. Tu, F. Meng, J. Zhai, and Y. Liu. 2018. Towards robust neural machine translation. In *Proceedings of ACL*.
- C. Clark, K. Lee, M-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL*.
- E. Dinan, S. Humeau, B. Chintagunta, and J. Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of EMNLP-IJCNLP*, pages 4529–4538.
- J. Ebrahimi, D. Lowd, and D. Dou. 2018. On adversarial examples for character-level neural machine translation. In *Proceedings of COLING*.
- M. Gardner et al. 2020. Evaluating models local decision boundaries via contrast sets. In *Proceedings of EMNLP*.
- M. Geva, Y. Goldberg, and J. Berant. 2019. Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. In *Proceedings of EMNLP-IJCNLP*.
- M. Glockner, V. Shwartz, and Y. Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of ACL*.
- J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of NAACL-HLT*.
- R. Jia and P. Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*.
- D. Kaushik, E. Hovy, and Z. C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proceedings of ICLR*.
- D. Khashabi, S. Chaturvedi, M. Roth, S. Upadhyay, and D. Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of NAACL*.
- D. Khashabi, T. Khot, A. Sabharwal, P. Clark, O. Etzioni, and D. Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *Proceedings of IJCAI*.
- S. Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of NAACL*.
- T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. 2019. Natural questions: a benchmark for question answering research. *TACL*, 7:453–466.
- H. J. Levesque, E. Davis, and L. Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.
- H. Peng, D. Khashabi, and D. Roth. 2015. Solving hard coreference problems. In *Proceedings of NAACL*.
- P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*.
- K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. 2020. WINOGRANDE: An adversarial Winograd schema challenge at scale. In *Proceedings of AAAI*.
- M. Shah, X. Chen, M. Rohrbach, and D. Parikh. 2019. Cycle-consistency for robust visual question answering. In *Proceedings of CVPR*.
- A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. 2019. SuperGlue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of NourIPS*.

A Question Perturbations: Further Details

We provide further details about the annotation.

The task starts with a qualification step: we ask annotators to read a collection of meticulously designed instructions that describe the task. The annotators are allowed to participate, only after successfully passing the test included in the instructions.

In addition, we restrict the task to “Master” workers from English-speaking countries (USA, UK, Canada, and Australia), at least 500 finished HITs and at least a 95% acceptance rate.

Here is a screen cast of the relevant annotation interface interface: <https://youtu.be/MWbCRwanb0A>

During our earlier pilot experiments, we observed that the strategies used for perturbing “yes” questions tend to be different from those used for “no” questions. To make the task less demanding and help workers focus on a limited cognitive task, the annotation is done in two phases; one for “yes” questions, and another for “no” questions.

Table 1 provides a summary of BOOLQ₊ stats.

Measure	Full	Train	Dev	Test
# of questions	17,323	9727	4434	3162
# of “yes” questions	9,724	5733	2263	1728
# of “no” questions	7,599	3994	2171	1434
# of clusters	4064	2408	919	737
average cluster size	4.3	4.1	4.8	4.3
median cluster size	3.0	3.0	3.0	3.0

Table 1: Statistics of BOOLQ₊.

B Details of ROBERTA Training

We train the model on two-way questions using the input format: “[CLS] passage [SEP] question [SEP] answer”. The model scores each answer (“yes” or “no”) by applying a linear classifier over the [CLS] representation for each answer’s corresponding input. We train the linear classifier (and fine-tune ROBERTA weights) on the training sets and evaluate them on their corresponding dev/test sets. We fixed the learning rate to 1e-5 as it generally performed the best on our datasets. We only varied the number of training epochs: {7, 9, 11} and effective batch sizes: 16, 32. We chose this small hyper-parameter sweep to ensure that each model was fine-tuned using the same hyper-parameter sweep while not being prohibitively expensive. Each model was selected based on the

best validation set accuracy. We report the numbers corresponding to the selected models on the test set.

C Performances Across Datasets

We compare a collection of solvers across our target datasets: the complete BOOLQ₊ dataset (dataset constructed from \mathcal{D}_S via perturbation), the original BOOLQ dataset, expert perturbations on BOOLQ, and binary subset of MULTIRC.

The results are summarized in Table 2. Most of the rows are ROBERTA trained on a specified dataset. We have also included a row corresponding to a system trained on the union of BOOLQ and BOOLQ₊, referred to as BOOLQ++ for brevity. Most of the datasets are slightly skewed between the two classes, which is why the majority label baseline (*Always-Yes* or *Always-No*) achieves scores above 50%. Rows indicated with * are reported directly from prior work. The human prediction on BOOLQ₊ is the majority label of 5 independent AMT annotators. The human performance on BOOLQ and MULTIRC are directly reported from SuperGLUE (Wang et al., 2019) leaderboard.⁵

Here are the key observations in this table:

- While ROBERTA has almost human-level performance when trained and tested within BOOLQ, it suffers significant performance degradation when evaluated on other datasets (e.g., 68.7% on BOOLQ₊).
- The systems fine-tuned on BOOLQ++ consistently generalize better across datasets.

D Cluster-Level Evaluation

An additional benefit of our approach is that it produces datasets with an inherent cluster structure. This enables the use of metrics such as ConsensusScore (Shah et al., 2019) to evaluate the extent to which a model acts *consistently* within each cluster, which provides another measure of robustness.

While evaluation measures are often computed on *per-instance* level, the cluster structure of BOOLQ₊ enables us to provide *per-cluster* metrics of quality. In particular, we are interested in the following question: *to what extent do our models act consistently across questions in each cluster?*

To measure this, we use the *consensus score* $CS(k)$ introduced by Shah et al. (2019). For an

⁵<https://super.gluebenchmark.com/leaderboard/>

Model	Trained on	Evaluated on	Acc.
Human*	—	MULTIRC	~83
ROBERTA	BOOLQ++	MULTIRC	78.8
ROBERTA	BOOLQ _{••}	MULTIRC	70.3
ROBERTA	BOOLQ	MULTIRC	76.5
<i>Maj-Vote</i>	—	MULTIRC	63.4
Human*	—	BOOLQ	89.0
ROBERTA	BOOLQ++	BOOLQ	85.5
ROBERTA	BOOLQ	BOOLQ	86.1
ROBERTA	BOOLQ _{••}	BOOLQ	78.6
<i>Maj-Vote</i>	—	BOOLQ	62.2
Human	—	BOOLQ _{••}	89.4
ROBERTA	BOOLQ++	BOOLQ _{••}	81.1
ROBERTA	BOOLQ	BOOLQ _{••}	68.7
ROBERTA	BOOLQ _{••}	BOOLQ _{••}	78.4
<i>Maj-Vote</i>	—	BOOLQ _{••}	53.2
Human	—	BOOLQ- <i>e</i> _{••}	?
ROBERTA	BOOLQ++	BOOLQ- <i>e</i> _{••}	76.4
ROBERTA	BOOLQ	BOOLQ- <i>e</i> _{••}	71.1
ROBERTA	BOOLQ _{••}	BOOLQ- <i>e</i> _{••}	69.3
<i>Maj-Vote</i>	—	BOOLQ- <i>e</i> _{••}	50.7

Table 2: Various systems trained and evaluated on different datasets. Best non-human scores are in **bold**. Numbers in percentage.

integer parameter $k \geq 1$, the score $CS(k)$ for a single cluster C is defined as the fraction of size- k sub-clusters of C where the model answers *all* instances correctly. The $CS(k)$ score for a clustered dataset is the average of these scores across all clusters. Intuitively, $k = 1$ represents the traditional un-clustered accuracy (assuming all clusters with the same size). As k grows to reach the cluster size, models must answer the *entire* cluster correctly in order to score positively on that cluster.

We plot this score for $k \in \{1, 2, 3, 4\}$ for various QA models in Fig 4. While all the models (including human) have decreasing consensus score for larger values of k , machine solvers have a steeper slope compared to human. As a result, we have an even larger gap of 17% between human-ROBERTA (at $k = 4$), when evaluated on their consistency.

E Rule-Based Perturbations

An alternate way to get cheap perturbations would be to use rule-based paraphrase systems — which are arguably cheaper than human-annotated perturbations.

Our intuition is that rule-based perturbations generally have simplistic definitions and hence, rarely benefit general reasoning problems in language. Interesting and diverse rule-based perturbations can be difficult to develop, and existing approaches

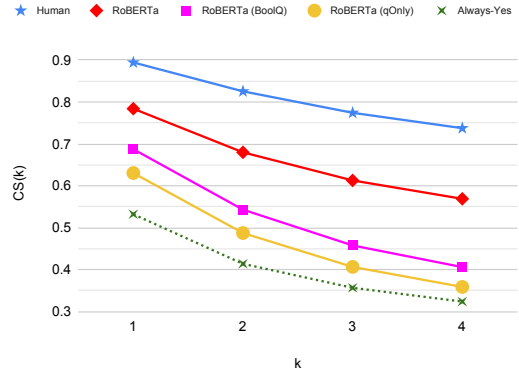


Figure 4: Consensus metric $CS(k)$ on the y-axis for various values of k on the x-axis.

are often reverse-engineered by QA models. Further, unlike our proposal, automatic perturbation approaches, such as question rephrasing, generally preserve the answer and do not use the provided context the question is referring to, limiting their richness.

That being said, we put some effort into developing rule-based/machine-generated baselines for comparison. However, since these efforts did not result in any reasonably sophisticated baselines, we decided to not include them in the main text.

Here we’re showing examples of perturbations generated via a recent machine paraphraser system.⁶

Original Question: Will there be a season 4 of da vincis demons? FALSE

The corresponding machine-perturbed questions are:

Will there be a season four of da vinci demons? FALSE
 Will there be season four of da vinci demons? FALSE
 Is there a season four of da vinci demons? FALSE
 Is there gonna be a season four of da vinci demons? FALSE

These automated perturbations stand in contrast with our human-perturbed questions, which also take the provided context into account:

Was there a season 3 of da vinci’s demons? TRUE
 There be a season 4 of da vinci’s demons? FALSE
 Will there be no season 4 of da vinci’s demons? TRUE

As evident by the example, the machine-generated perturbations are generally minor and, not surprisingly, did not provide a useful enough signal to the model to improve its accuracy. We are open to suggestions if the reviewers have any suggestion on creating more reasonable rule-based perturbation baselines.

⁶<https://github.com/decompositional-semantics-initiative/improved-ParaBank-rewriter>