Logic-Guided Data Augmentation and Regularization for Consistent Question Answering

Akari Asai[†] and **Hannaneh Hajishirzi**[†‡]
†University of Washington ‡Allen Institute for AI
{akari, hannaneh}@cs.washington.edu

Abstract

Many natural language questions require qualitative, quantitative or logical comparisons between two entities or events. This paper addresses the problem of improving the accuracy and consistency of responses to comparison questions by integrating logic rules and neural models. Our method leverages logical and linguistic knowledge to augment labeled training data and then uses a consistency-based regularizer to train the model. Improving the global consistency of predictions, our approach achieves large improvements over previous methods in a variety of question answering (QA) tasks including multiple-choice qualitative reasoning, cause-effect reasoning, and extractive machine reading comprehension. In particular, our method significantly improves the performance of RoBERTa-based models by 1-5% across datasets. We advance state of the art by around 5-8% on WIQA and QuaRel and reduce consistency violations by 58% on HotpotQA. We further demonstrate that our approach can learn effectively from limited data.[1]

1 Introduction

Comparison-type questions (Tandon et al., 2019; Tafjord et al., 2019; Yang et al., 2018) ask about relationships between properties of entities or events such as cause-effect, qualitative or quantitative reasoning. To create comparison questions that require inferential knowledge and reasoning ability, annotators need to understand context presented in multiple paragraphs or carefully ground a question to the given situation. This makes it challenging to annotate a large number of comparison questions. Most current datasets on comparison questions are much smaller than standard machine reading comprehension (MRC) datasets (Rajpurkar et al., 2016; Joshi et al., 2017). This poses new challenges to standard models, which are known to exploit statistical patterns or annotation artifacts in these datasets (Sugawara et al., 2018; Min et al., 2019a). Importantly, state-of-the-art models show inconsistent comparison predictions as shown in Figure 1. Improving the consistency of predictions has been previously studied in natural language inference (NLI) tasks (Minervini and Riedel, 2018; Li et al., 2019), but has not been addressed in QA.

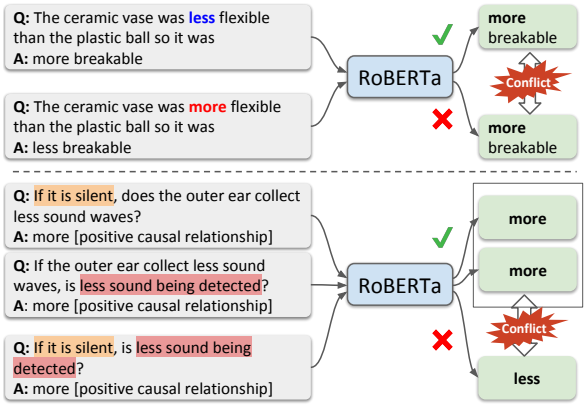

Figure 1: Inconsistent predictions by RoBERTa. Top row shows an example of symmetric inconsistency and the second row shows an example of transitive inconsistency. The examples are partially modified.

In this paper, we address the task of producing globally consistent and accurate predictions for comparison questions leveraging logical and symbolic knowledge for data augmentation and training regularization. Our data augmentation uses a set of logical and linguistic knowledge to develop additional consistent labeled training data. Subsequently, our method uses symbolic logic to incorporate consistency regularization for additional supervision signal beyond inductive bias given by data augmentation. Our method generalizes previous consistency-promoting methods for NLI tasks (Minervini and Riedel, 2018; Li et al., 2019)

[1]Our code and data is available at https://github.com/AkariAsai/logic_guided_qa.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5642–5650
July 5 - 10, 2020.

to adapt to substantially different question formats.

Our experiments show significant improvement over the state of the art on a variety of QA tasks: a classification-based causal reasoning QA, a multiple choice QA for qualitative reasoning and an extractive MRC task with comparisons between entities. Notably, our data augmentation and consistency constrained training regularization improves performance of RoBERTa-based models (Liu et al., 2019) by 1.0%, 5.0% and 2.5% on WIQA, QuaRel and HotpotQA. Our approach advances the state-of-the-art results on WIQA and QuaRel with 4.7 and 8.4% absolute accuracy improvement, respectively, reducing inconsistent predictions. We further demonstrate that our approach can learn effectively from limited labeled data: given only 20% of the original labeled data, our method achieves performance on par with a competitive baseline learned with the full labeled data.

2 Related Work

Data augmentation has been explored in a variety of tasks and domains (Krizhevsky et al., 2009; Cubuk et al., 2019; Park et al., 2019). In NLP, using back-translation (Yu et al., 2018) or dictionary based word replacement (Zhang et al., 2015) has been studied. Most relevant to our work, Kang et al. (2018) study NLI-specific logic and knowledge-based data augmentation. Concurrent to our work, Gokhale et al. (2020) study visual QA models' ability to answer logically composed questions, and show the effectiveness of logic-guided data augmentation. Our data augmentation does not rely on task-specific assumptions, and can be adapted to different formats of QA task. We further leverage consistency-promoting regularization, which gives improvements in accuracy and consistency.

Improving prediction consistency via training regularization has been studied in NLI tasks. Minervini and Riedel (2018) present model-dependent first-order logic guided adversarial example generation and regularization. Li et al. (2019) introduce consistency-based regularization incorporating the first-order logic rules. Previous approach is model-dependent or relies on NLI-specific rules, while our method is model-agnostic and is more generally applicable by combining it with data augmentation.

Regularizing loss to penalize violations of structural constraints in models' output has been also studied in previous work on constraint satisfaction in structured learning (Lee et al., 2019; Ganchev et al., 2010). Our work regularizes models to produce globally consistent predictions among augmented data following logical constraints, while those studies incorporates structured prediction models following linguistics rules.

3 Method

We present the components of our QA method: first-order logic guided data augmentation (Section 3.1 and Section 3.2), and consistency-based regularization (Section 3.3).

3.1 Consistent Question Answering

For globally consistent predictions in QA, we require responses to follow two important general logical rules: *symmetric consistency* and *transitive consistency*, which are illustrated in Figure 1 and are formally described below.

Let q, p, a be a question, a paragraph and an answer predicted by a model. $\mathbf{A}$ is a set of answer candidates. Each element of $\mathbf{A}$ can be a span in p, a class category, or an arbitrary answer choice. $X = \{q, p, a\}$ represents a logic atom.

Symmetric consistency In a comparison question, small surface variations such as replacing words with their antonyms can reverse the answer, while keeping the overall semantics of the question as before. We define symmetry of questions in the context of QA as follows: $(q, p, a^*) \leftrightarrow (q_{sym}, p, a^*_{sym})$, where q and q_{sym} are antonyms of each other, and a^*_{sym} is the opposite of the ground-truth answer a^* in $\mathbf{A}$. For example, the two questions in the first row of Figure 1 are symmetric pairs. We define the symmetric consistency of predictions in QA as the following logic rule:

$$(q, p, a) \rightarrow (q_{sym}, p, a_{sym}), \quad (1)$$

which indicates a system should predict a_{sym} given (q_{sym}, p), if it predicts a for (q, p).

Transitive consistency. Transitive inference between three predicates A, B, C is represented as: $A \rightarrow B \wedge B \rightarrow C$ then $A \rightarrow C$ (Gazes et al., 2012). In the context of QA, the transitive examples are mainly for causal reasoning questions that inquire about the effect e given the cause c. The second row of Figure 1 shows an example where transitive consistency is violated. For two questions q_1 and q_2 in which the effect of q_1 $(= e_1)$ is equal to the cause of q_2 $(= c_2)$, we define the transitive consistency of predictions as follows:

$$(q_1, p, a_1) \wedge (q_2, p, a_2) \rightarrow (q_{trans}, p, a_{trans}). \quad (2)$$

	WIQA (Tandon et al., 2019)	QuaRel (Tafjord et al., 2019)	HotpotQA (Yang et al., 2018)
reasoning format	Causal Reasoning classification	Qualitative Reasoning multiple choice	Qualitative Comparison of entities span extraction
p	The rain seeps into the wood surface. When rain evaporates it leaves the wood. It takes the finish of the wood with it. The wood begins to lose it's luster.	Supposed you were standing on the planet Earth and Mercury. When you look up in the sky and see the sun,	Golf Magazine is a monthly golf magazine owned by Time Inc. El Nuevo Cojo Ilustrado is an American Spanish language magazine.
q	q_1:If a tsunami happens, will **wood be more moist**?, q_2: If **wood is more moist**, is more weathering occurring?	Which planet would the sun appear **larger**?	El Nuevo Cojo and Golf Magazine, which one is owned by Time Inc?
A	{more, less, no effects}	{Mercury, Earth}	{Golf Magazine, El Nuevo Cojo}
a^*	a_1^* : more, a_2^* : more	Mercury	Golf Magazine
q_{aug}	If a tsunami happens, is more weathering occurring?	Which planet would the sun appear **smaller**?	Which one is **not** owned by Time Inc, Golf Magazine El Nuevo Cojo?
a_{aug}^*	more	Earth	El Nuevo Cojo

Table 1: An augmented transitive example for WIQA, and symmetric examples for QuaRel and HotpotQA. We partially modify paragraphs and questions. The bold characters denote a shared event connecting two questions. The parts written in red or blue denote antonyms, and highlighted text is negation added by our data augmentation.

3.2 Logic-guided Data Augmentation

Given a set of training examples X in the form of (q, p, a^*), we automatically generate additional examples $X_{aug} = \{q_{aug}, p, a_{aug}^*\}$ using symmetry and transitivity logical rules. The goal is to augment the training data so that symmetric and transitive examples are observed during training. We provide some augmented examples in Table 1.

Augmenting symmetric examples To create a symmetric question, we convert a question into an opposite one using the following operations: (a) replace words with their antonyms, (b) add, or (c) remove words. For (a), we select top frequent adjectives or verbs with polarity (e.g., *smaller, increases*) from training corpora, and expert annotators write antonyms for each of the frequent words (we denote this small dictionary as **D**). More details can be seen in Appendix A. For (b) and (c), we add negation words or remove negation words (e.g., *not*). For all of the questions in training data, if a question includes a word in **D** for the operation (a), or matches a template (e.g., `which * is` $\leftrightarrow$ `which * is not`) for operations (b) and (c), we apply the operation to generate q_{sym}.[2] We obtain a_{sym}^* by re-labeling the answer a^* to its opposite answer choice in **A** (see Appendix B).

Augmenting transitive examples We first find a pair of two cause-effect questions $X_1 = (q_1, p, a_1^*)$ and $X_2 = (q_2, p, a_2^*)$, whose q_1 and q_2 consist of (c_1, e_1) and (c_2, e_2), where $e_1 = c_2$ holds. When a_1^* is a *positive causal relationship*, we create a new example $X_{trans} = (q_3, p, a_2^*)$ for $q_3 = (c_1, e_2)$.

Sampling augmented data Adding all consistent examples may change the data distribution from the original one, which may lead to a deterioration in performance (Xie et al., 2019). One can select the data based on a model's prediction inconsistencies (Minervini and Riedel, 2018) or randomly sample at each epoch (Kang et al., 2018). In this work, we randomly sample augmented data at the beginning of training, and use the same examples for all epochs during training. Despite its simplicity, this yields competitive or even better performance than other sampling strategies.[3]

3.3 Logic-guided Consistency Regularization

We regularize the learning objective (task loss, $\mathcal{L}_{task}$) with a regularization term that promotes consistency of predictions (consistency loss, $\mathcal{L}_{cons}$).

$$\mathcal{L} = \mathcal{L}_{task}(X) + \mathcal{L}_{cons}(X, X_{aug}). \quad (3)$$

The first term $\mathcal{L}_{task}$ penalizes making incorrect predictions. The second term $\mathcal{L}_{cons}$[4] penalizes making predictions that violate symmetric and transitive logical rules as follows:

$$\mathcal{L}_{cons} = \lambda_{sym}\mathcal{L}_{sym} + \lambda_{trans}\mathcal{L}_{trans}, \quad (4)$$

where λ_{sym} and λ_{trans} are weighting scalars to balance the two consistency-promoting objectives.

[2] We observe that (b)(c) are less effective than (a) in WIQA or QuaRel, while especially (b) contributes to the performance improvements on HotpotQA as much as (a) does.

[3] We do not add X_{aug} if the same pair has already exist.

[4] We mask the $\mathcal{L}_{cons}$ for the examples without symmetric or transitive consistent examples.

Dataset	WIQA					QuaRel				HotpotQA		
	Dev			Test	v (%)	Dev		Test	v (%)	Dev		v (%)
x% data (# of X)	20% (6k)	40% (12k)	100 % (30k)	100% (30k)	100% (30k)	20% (0.4k)	100% (2k)	100% (2k)	100% (2k)	20% (18k)	100 % (90k)	100 % (90k)
SOTA	–	–	–	73.8	–	–	–	76.6	–	–	–	–
RoBERTa	61.1	74.1	74.9	77.5	12.0	56.4	81.1	80.0	19.2	71.0	75.5	65.2
DA	72.1	75.5	76.3	78.3	6.0	69.3	84.5	84.7	13.3	**73.1**	**78.0**	**6.3**
DA + Reg	**73.9**	**76.1**	**77.0**	**78.5**	**5.8**	**70.9**	**85.1**	**85.0**	**10.3**	71.9	76.9	7.2

Table 2: **WIQA, QuaRel and HotpotQA results**:we report test and development accuracy (%) for WIQA and QuaRel and development F1 for HotpotQA. DA and Reg denote data augmentation and consistency regularization. "SOTA" is Tandon et al. (2019) for WIQA and Mitra et al. (2019) for QuaRel. v presents violations of consistency.

Previous studies focusing on NLI consistency (Li et al., 2019) calculate the prediction inconsistency between a pair of examples by swapping the premise and the hypothesis, which cannot be directly applied to QA tasks. Instead, our method leverages consistency with data augmentation to create paired examples based on general logic rules. This enables the application of consistency regularization to a variety of QA tasks.

Inconsistency losses The loss computes the dissimilarity between the predicted probability for the original labeled answer and the one for the augmented data defined as follows:

$$\mathcal{L}_{sym} = |\log p(a|q,p) - \log p(a_{aug}|q_{aug},p)|. \quad (5)$$

Likewise, for transitive loss, we use absolute loss with the product T-norm which projects a logical conjunction operation $(q_1, p, a_1) \wedge (q_2, c, a_2)$ to a product of probabilities of two operations, $p(a_1|q_1,p)p(a_2|q_2,p)$, following Li et al. (2019). We calculate a transitive consistency loss as:

$$\mathcal{L}_{trans} = |\log p(a_1|q_1,p) + \log p(a_2|q_2,p) - \log p(a_{trans}|q_{trans},p)|.$$

Annealing The model's predictions may not be accurate enough at the beginning of training for consistency regularization to be effective. We perform annealing (Kirkpatrick et al., 1983; Li et al., 2019; Du et al., 2019). We first set $\lambda_{\{sym,trans\}} = 0$ in Eq. (4) and train a model for τ epochs, and then train it with the full objective.

4 Experiments

Datasets and experimental settings We experiment on three QA datasets: WIQA (Tandon et al., 2019), QuaRel (Tafjord et al., 2019) and HotpotQA (oracle, comparison questions[5]) (Yang et al., 2018).

[5]We train models on both bridge and comparison questions, and evaluate them on extractive comparison questions only.

	WIQA		QuaRel	
metric	acc	v (%)	acc	v (%)
DA (logic) + Reg	77.0	5.8	85.1	10.3
DA (logic)	76.3	6.0	84.5	13.5
DA (standard)	75.2	12.3	83.3	14.5
Reg	75.8	11.4	–	–
Baseline	74.9	12.0	81.1	19.2

Table 3: Ablation studies of data augmentation on WIQA and QuaRel development dataset.

As shown in Table 1, these three datasets are substantially different from each other in terms of required reasoning ability and task format. In WIQA, there are 3,238 symmetric examples and 4,287 transitive examples, while 50,732 symmetric pairs and 1,609 transitive triples are missed from the original training data. HotpotQA and QuaRel do not have any training pairs requiring consistency. Our method randomly samples 50, 80, 90% of the augmented data for WIQA, QuaRel and HotpotQA, resulting in 24,715/836/3,538 newly created training examples for those datasets, respectively.

We use standard F1 and EM scores for performance evaluation on HotpotQA and use accuracy for WIQA and QuaRel. We report a violation of consistency following Minervini and Riedel (2018) to evaluate the effectiveness of our approach for improving prediction consistencies. We compute the violation of consistency metric v as the percentage of examples that do not agree with symmetric and transitive logical rules. More model and experimental details are in Appendix.

Main Results Table 2 demonstrates that our methods (DA and DA + Reg) constantly give 1 to 5 points improvements over the state-of-the-art RoBERTa QA's performance on all three of the datasets, advancing the state-of-the-art scores on WIQA and QuaRel by 4.7% and 8.4%, respectively. On all three datasets, our method signifi-

	WIQA Input	RoBERTa	DA	DA+Reg
p	Sound enters the ears of a person. The sound hits a drum that is inside the ears.			
q	If the person has his ears more protected, will less sound be detected? [a^*: More]	More (0.79)	More (0.93)	More (0.93)
q_{sym}	If the person has his ears less protected, will less sound be detected? [$a^{sym}*$: Less]	More (0.87)	More (0.72)	Less (0.89)
p	Squirrels try to eat as much as possible. Squirrel gains weight.			
q_1	If the weather has a lot of snow, cannot squirrels eat as much as possible? [a_1^*: More]	Less (0.75)	More (0.48)	More (0.94)
q_2	If squirrels cannot eat as much as possible, will not the squirrels gain weight? [a_2^*: More]	More (0.86)	More (0.94)	More (0.93)
q_{trans}	If the weather has a lot of snow, will not the squirrels gain weight? [a_{trans}^*: More]	Less (0.75)	More (0.43)	More (0.87)
	HotpotQA (comparison) Input	RoBERTa	DA	
p	B. Reeves Eason is a film director, actor and screenwriter. Albert S. Rogell a film director.			
q	Who has more scope of profession, B. Reeves Eason or Albert S. Rogell? [a^*: B. Reeves Eason]	B. Reeves Eason	B. Reeves Eason	
q_{sym}	Who has less scope of profession, B. Reeves or Albert S. Rogell? [a_{sym}^*: Albert S. Rogell]	B. Reeves Eason	Albert S. Rogell	

Table 4: Qualitative comparison of RoBERTa, + DA, + DA + Reg. The examples are partially modified.

cantly reduces the inconsistencies in predictions, demonstrating the effects of both data augmentation and regularization components. Notably on WIQA, RoBERTa shows violation of consistency in 13.9% of the symmetric examples and 10.0% of the transitive examples. Our approach reduces the violations of symmetric and transitive consistencies to 8.3% and 2.5%, respectively.

Results with limited training data Table 2 also shows that our approach is especially effective under the scarce training data setting: when only 20% of labeled data is available, our DA and Reg together gives more than 12% and 14% absolute accuracy improvements over the RoBERTa baselines on WIQA and QuaRel, respectively.

Ablation study We analyze the effectiveness of each component on Table 3. DA and Reg each improves the baselines, and the combination performs the best on WIQA and QuaRel. DA (standard) follows a previous standard data augmentation technique that paraphrases words (verbs and adjectives) using linguistic knowledge, namely WordNet (Miller, 1995), and does not incorporate logical rules. Importantly, DA (standard) does not give notable improvement over the baseline model both in accuracy and consistency, which suggests that logic-guided augmentation gives additional inductive bias for consistent QA beyond amplifying the number of train data. As WIQA consists of some transitive or symmetric examples, we also report the performance with Reg only on WIQA. The performance improvements is smaller, demonstrating the importance of combining with DA.

Qualitative Analysis Table 4 shows qualitative examples, comparing our method with RoBERTa baseline. Our qualitative analysis shows that DA+Reg reduces the confusion between opposite choices, and assigns larger probabilities to the ground-truth labels for the questions where DA shows relatively small probability differences.

On HotpotQA, the baseline model shows large consistency violations as shown in Table 2. The HotpotQA example in Table 4 shows that RoBERTa selects the same answer to both q and q_{sym}, while DA answers correctly to both questions, demonstrating its robustness to surface variations. We hypothesize that the baseline model exploits statistical pattern, or dataset bias presented in questions and that our method reduces the model's tendency to exploit those spurious statistical patterns (He et al., 2019; Elkahky et al., 2018), which leads to large improvements in consistency.

5 Conclusion

We introduce a logic guided data augmentation and consistency-based regularization framework for accurate and globally consistent QA, especially under limited training data setting. Our approach significantly improves the state-of-the-art models across three substantially different QA datasets. Notably, our approach advances the state-of-the-art on QuaRel and WIQA, two standard benchmarks requiring rich logical and language understanding. We further show that our approach can effectively learn from extremely limited training data.

Acknowledgments

This research was supported by ONR N00014-18-1-2826, DARPA N66001-19-2-403, NSF (IIS1616112, IIS1252835), Allen Distinguished Investigator Award, Sloan Fellowship, and The Nakajima Foundation Fellowship. We thank Antoine Bosselut, Tim Dettmers, Rik Koncel-Kedziorski, Sewon Min, Keisuke Sakaguchi, David Wadden, Yizhong Wang, the members of UW NLP group and AI2, and the anonymous reviewers for their insightful feedback.

References

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. 2019. AutoAugment: Learning augmentation strategies from data. In *CVPR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Ming Ding, Chang Zhou, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *ACL*.

Xinya Du, Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. Be consistent! improving procedural text comprehension using label consistency. In *NAACL*.

Ali Elkahky, Kellie Webster, Daniel Andor, and Emily Pitler. 2018. A challenge set and methods for noun-verb ambiguity. In *EMNLP*.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Regina Paxton Gazes, Nicholas W Chee, and Robert R Hampton. 2012. Cognitive mechanisms for transitive inference performance in rhesus monkeys: Measuring the influence of associative strength and inferred order. *Journal of Experimental Psychology: Animal Behavior Processes*, 38(4):331.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. VQA-LOL: Visual question answering under the lens of logic. *arXiv:2002.08325*.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias for natural language inference by fitting the residual. In *EMNLP Workshop on DeepLo*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples. In *ACL*.

Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science*.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Jay Yoon Lee, Sanket Vaibhav Mehta, Michael Wick, Jean-Baptiste Tristan, and Jaime Carbonell. 2019. Gradient-based inference for networks with output constraints. In *AAAI*.

Tao Li, Vivek Cupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. In *EMNLP*.

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

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. Compositional questions do not necessitate multi-hop reasoning. In *ACL*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural NLI models to integrate logical background knowledge. In *ACL*.

Arindam Mitra, Chitta Baral, Aurgho Bhattacharjee, and Ishan Shrivastava. 2019. A generate-validate approach to answering questions about qualitative relationships. *arXiv:1908.03645*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv:1904.01038*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv:1904.08779*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *EMNLP*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019. QuaRel: A dataset and models for answering questions about qualitative relationships. In *AAAI*.

Niket Tandon, Bhavana Dalvi Mishra, Keisuke Sakaguchi, Antoine Bosselut, and Peter Clark. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *EMNLP*.

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

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V Le. 2019. Adversarial examples improve image recognition. *arXiv:1911.09665*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. QANet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NeurIPS*.

A Details of Human Annotations

In this section, we present the details of human annotations used for symmetric example creation (the (a) operation). We first sample the most frequent 500 verbs, 50 verb phrases and 500 adjectives from from the WIQA and QuaRel training data. Then, human annotators select words with some polarity (e.g., increase, earlier). Subsequently, they annotate the antonyms for each of the selected verbs and adjectives. Consequently, we create 64 antonym pairs mined from a comparison QA dataset. We reuse the same dictionary for all three datasets. Examples of annotated antonym pairs are shown in Table 5.

adjectives	verbs & verb phrases
more ↔ less	increase ↔ decrease
slowly ↔ quickly	heat up ↔ cool down
stronger ↔ weaker	lose weight ↔ gain weight
later ↔ earlier	raise ↔ drop
younger ↔ older	remove ↔ add

Table 5: Ten examples of annotated antonyms for comparison type questions.

B Details of answer re-labeling on WIQA and HotpotQA

We present the details of answer re-labeling operations in WIQA and HotpotQA, where the number of the answer candidates is more than two.

Answer re-labeling in WIQA (symmetric) In WIQA, each labeled answer a^* takes one of the following values: $\{$*more*, *less*, *no effects*$\}$. Although *more* and *less* are opposite, *no effects* is a neutral choice. In addition, in WIQA, a question q consists of a cause c and an effect e, and we can operate the three operations (a) replacement, (b) addition and (c) removal of words. When we add the operations to both of c and e, it would convert the question to opposite twice, and thus the original answer remains same. When we add one of the operation to either of c or e, it would convert the question once, and thus, the answer should be the opposite one. Given these two assumption, we re-label answer as: (i) if we apply only one operation to either e or c and a^* is *more* or *less*, the a^*_{sym} will be the opposite of a^*, (ii) if we apply only one operation to either e or c and a^* is *no effect*, the a^*_{sym} will remain *no effect*, and (iii) if we apply one operation to each of e and c, the a_{sym} remains the same.

Answer re-labeling in WIQA (transitive) For transitive examples, we re-label answers based on two assumptions on causal relationship. A transitive questions are created from two questions, $X_1 = (q_1, p, a_1^*)$ and $X_2 = (q_2, p, a_2^*)$, where q_1 and q_2 consist of (c_1, e_1) and (c_2, e_2) and $e_1 = c_2$ holds. If a_1 for X_1 is "more", it means that the c_1 causes e_1. e_1 is equivalent to the cause for the second question (c_2), and a_2^* represents the causal relationship between c_2 and e_2. Therefore, if a_1^* is a positive causal relationship, c_1 and e_2 have the relationship defined as a_2^*. We assume that if the a_1^* is "more", $a_3^*(= a_{trans}^*)$ will be same as a_2, and re-label answer following this assumption.

Answer re-labeling in HotpotQA In HotpotQA, answer candidates **A** are not given. Therefore, we extract possible answers from q. We extract two entities included in q by string matching with the titles of the paragraphs given by the dataset. If we find two entities to be compared and both of them are included in the gold paragraphs, we assume the two entities are possible answer candidates. The new answer a_{sym}^* will be determined as the one which is not the original answer a^*.

C Details of Baseline Models

We use RoBERTa (Li et al., 2019) as our baseline. Here, we present model details for each of the three different QA datasets.

Classification-based model for WIQA As the answer candidates for WIQA questions are set to {more, less, no effects}, we use a classification based models as studied for NLI tasks. The input for this model is `[CLS]` p `[SEP]` q `[SEP]`. We use the final hidden vector corresponding to the first input token (`[CLS]`) as the aggregate representation. We then predict the probabilities of an answer being a class C in the same manner as in (Devlin et al., 2019; Liu et al., 2019).

Multiple-choice QA model for QuaRel For QuaRel, two answer choices are given, and thus we formulate the task as multiple-choice QA. In the original dataset, all of the p, q and **A** are combined together (e.g., *The fastest land animal on earth, a cheetah was having a 100m race against a rabbit. Which one won the race? (A) the cheetah (B) the rabbit*), and thus we process the given combined questions into p, q and **A** (e.g., the question written above will be p =*The fastest land animal on earth, a cheetah was having a 100m race against a rabbit.* , q =Which one won the race? and **A** ={the cheetah, rabbit}). Then the input will be `[CLS]` p `[SEP]` *"Q: "* q *"A: "* a_i `[SEP]`, and we will use the final hidden vector corresponding to the first input token (`[CLS]`) as the aggregate representation. We then predict the probabilities of an answer being an answer choice a_i in the same manner as in (Liu et al., 2019).

Span QA model for HotpotQA We use the RoBERTa span QA model studied for SQuAD (Devlin et al., 2019; Liu et al., 2019) for HotpotQA. As we only consider the questions whose answers can be extracted from p, we do not add any modifications to the model unlike some previous studies in HotpotQA (Min et al., 2019b; Ding et al., 2019).

D Details of Implementations and Experiments

Implementations Our implementations are all based on PyTorch. In particular, to implement our classification based and span-based model, we use `pytorch-transformers` (Wolf et al., 2019)[6]. To implement our multiple choice model, we use `fairseq` (Ott et al., 2019)[7].

Hyper-parameters For HotpotQA, we train a model for six epochs in total. For the model without data augmentation or regularization, we train on the original dataset for six epochs. For the models with data augmentation, we first train them on the original HotpotQA train data (including both bridge and comparison questions) for three epochs, and then train our model with augmented data and regularization for three epochs. For HotpotQA, we train our model with both bridge and comparison questions, and evaluate on comparison questions whose answers can be extracted from the context.

Due to the high variance of the performance in the early stages of the training for small datasets such as QuaRel or WIQA, for these two datasets, we set the maximum number of training epochs to 150 and 15, respectively. We terminate the training when we do not observe any performance improvements on the development set for 5 epochs for WIQA and 10 epochs for QuaRel, respectively. We use Adam as an optimizer ($\epsilon = 1\mathrm{E} - 8$) for

[6] https://github.com/huggingface/transformers

[7] https://github.com/pytorch/fairseq

all of the datasets. Other hyper-parameters can be seen from Table 6

hyper-parameters	WIQA	QuaRel	HotpotQA
train batch size	4	16	12
gradient accumulation	16	1	1
max token length	256	512	384
doc stride	–	–	128
learning rate	2E-5	1E-5	5E-5
weight decay	0.01	0.01	0.0
dropout	0.1	0.1	0.1
warm up steps	0	150	0
τ for annealing	3	25	3
λ_{sym}	0.5	0.1	0.25
λ_{trans}	0.05	–	–

Table 6: Ten examples of annotated antonyms for comparison type questions.

E Qualitative Examples on HotpotQA

As shown in Table 2, the state-of-the-art RoBERTa model produces a lot of consistency violations. Here, we present several examples where our competitive baseline model cannot answer correctly, while our RoBERTa+DA model answers correctly.

A question requiring world knowledge One comparison question asks "Who has **more** scope of profession, B. Reeves Eason or Albert S. Rogell", given context that B. Reeves is an American film director, actor and screenwriter and Albert S. Rogell is an American film director. The model correctly predicts "B. Reeves Eason" but fails to answer correctly to "Who has **less** scope of profession, B. Reeves Eason or Albert S. Rogell", although the two questions are semantically equivalent.

A question with negation We found that due to this reasoning pattern our model struggles on questions involving negation. Here we show one example. We create a question by adding a negation word, q_{sym},"Which species is not native to asia, corokia or rhodotypos?", where we add negation word not and the paragraph corresponding to the question is p ="**Corokia** is a genus in the Argophyllaceae family comprising about ten species native to New Zealand and one native to Australia. **Rhodotypos scandens** is a deciduous shrub in the family Rosaceae and is native to China, possibly also Japan.". The model predicts **Rhodotypos scandens**, while the model predicts the same answer to the original question q, 'which species is native to asia, corokia or rhodotypos?". This example shows that the model strongly relies on surface matching (i.e., "native to") to answer the question, without understanding the rich linguistic phenomena or having world knowledge.