Shortcutted Commonsense: Data Spuriousness in Deep Learning of Commonsense Reasoning

Ruben Branco and **António Branco** and **João Silva** and **João Rodrigues**
University of Lisbon
NLX – Natural Language and Speech Group, Department of Informatics
Faculdade de Ciências, Campo Grande, 1749-016 Lisboa, Portugal
{rmbranco,ambranco,jrsilva,jarodrigues}@fc.ul.pt

Abstract

Commonsense is a quintessential human capacity that has been a core challenge to Artificial Intelligence since its inception. Impressive results in Natural Language Processing tasks, including in commonsense reasoning, have consistently been achieved with Transformer neural language models, even matching or surpassing human performance in some benchmarks. Recently, some of these advances have been called into question: so called data artifacts in the training data have been made evident as spurious correlations and shallow shortcuts that in some cases are leveraging these outstanding results.

In this paper we seek to further pursue this analysis into the realm of commonsense related language processing tasks. We undertake a study on different prominent benchmarks that involve commonsense reasoning, along a number of key stress experiments, thus seeking to gain insight on whether the models are learning transferable generalizations intrinsic to the problem at stake or just taking advantage of incidental shortcuts in the data items.

The results obtained indicate that most datasets experimented with are problematic, with models resorting to non-robust features and appearing not to be learning and generalizing towards the overall tasks intended to be conveyed or exemplified by the datasets.

1 Introduction

Reasoning helps humans to cope with their experience, whether in a complex situation such as devising a plan to solve a pandemic or in a simple, intuitive inference like what will happen to a coffee mug when dropped to the ground. It helps to foresee new events as well as to form new beliefs and justify and defend them before others (Kintsch and Van Dijk, 1978; Mercier and Sperber, 2017).

Reasoning with knowledge widely shared by humans, usually termed as commonsense, makes up a significant portion of our higher level cognitive skills: commonsense knowledge encompasses human values and needs, and by reasoning with it we can organize sensible arguments and decide on effective actions. Endowing computing devices with these knowledge and reasoning capabilities should allow them to better get access to world view of humans, their needs, capabilities, beliefs, and thus allow them to act more appropriately.

To a large extent, acting on the basis of these capabilities can be verbalized, knowledge can be written down, and the inference chain of a reasoning process can be expressed in some human idiom. In Natural Language Processing (NLP), commonsense processing tasks usually involve answering questions that, when addressed by humans, require different types of commonsense knowledge and reasoning to be answered. Accordingly, for a model coping with this kind of tasks to achieve good performance, it should have acquired such knowledge and reasoning skills.

Whether current state-of-the-art NLP deep learning models genuinely grasp the underlying tasks they are handling is a debated topic. Recently, an increasing number of published experiments indicate that models may be latching at spurious cues present in the data (Zech et al., 2018; Geirhos et al., 2020; Niven and Kao, 2019), implying that they will severely lack generalization capacity when presented with out-of-distribution data.

As they become more refined, pre-trained language models are continuously closing the gap to humans in commonsense reasoning tasks (Zhou et al., 2020; Tamborrino et al., 2020; Lourie et al., 2021). In light of the skepticism about the true capabilities of deep learning models, this question cannot be avoided: to what extent are they actually learning commonsense reasoning?

In the present paper, we seek to contribute to a better understanding about if and how the models are learning commonsense reasoning skills. We se-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1504–1521
November 7–11, 2021.

lect four prominent commonsense reasoning tasks, namely Argument Reasoning Comprehension Task (ARCT) (Habernal et al., 2018), AI2 Reasoning Challenge (ARC) (Clark et al., 2018), Physical IQA (PIQA) (Bisk et al., 2020), and CommonsenseQA (CSQA) (Talmor et al., 2019)).

We adopt the successful Transformer architecture and resort to prominent pre-trained language models: the encoder-only RoBERTa (Liu et al., 2019b), the decoder-only GPT-2 (Radford et al., 2019), the encoder-decoder T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), and the neuro-symbolic COMET(BART) (Hwang et al., 2021)).

And we set on to gain insight with stress experiments that seek to find evidence that hopefully permits to advance in answering questions like the following:

Are the models taking into account the actual input as a whole? Or could the models be only looking at certain parts of the input, therefore not performing the underlying task but some derivative.

How robust are the models? Can they withstand adversarial attacks on the basis of powerful generalization they gained during training, or are they brittle and crumble under attack?

How well can models perform zero-shot evaluation on each other? To what extent can they transfer the knowledge between each other and how much could that be hindered by their being stuck in learning non-transferring cues based on spurious data artifacts?

The answers obtained in this paper indicate that most datasets experimented with are problematic, with models resorting to non-robust features and appearing not to be generalizing towards the overall tasks intended to be conveyed by the datasets.

In the next Section 2, we cover related work. The tasks experimented with are presented in Section 3, and the experiments undertaken are described in Section 4. Section 5 presents and discusses the results obtained. The paper closes with concluding remarks in Section 6.[1]

2 Related Work

NLP has found an overarching yet flexible approach to a wide range of its processing tasks in the Transformer model (Vaswani et al., 2017). An extensive research path has since then been pursued in the literature, seeking to refine its architecture and training methodology. It has become commonplace to first pre-train a language model on large corpora and then refine it with respect to a specific language processing task (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2019). Subsequent works would refine this methodology by introducing other pre-training tasks that benefit downstream performance (Liu et al., 2019b; Raffel et al., 2020; Lewis et al., 2020), and with that approach steadily increasing the state of the art on benchmarks such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019).

Shortcuts Deep neural networks, such as Transformers, as well as loss functions (cross-entropy) and optimizers, tend to favor simple functions (De Palma et al., 2018; Jacobsen et al., 2018; Sun and Nielsen, 2019; Valle-Pérez et al., 2018; Wu et al., 2017), which can be susceptible to wrong generalizations as the eventual models "hook" onto spurious artifacts in the data. This has been shown to happen in NLP, as illustrated with these examples, among others:

- Machine reading comprehension (MRC) models appear not to do much "reading" (Kaushik and Lipton, 2018), as models can perform reasonably well when given only a passage or a passage with a randomly assigned question, instead of a whole input with which this task was conceived.

- Large-scale natural language inference (NLI) datasets exhibit linguistic phenomena that correlate well with certain classes. Simple classifier models can perform well by looking only at the hypothesis, instead of the rest of the argument (Gururangan et al., 2018; Poliak et al., 2018).

- A study (Geva et al., 2019) on the annotator bias on natural language understanding datasets has found that annotator bias is evident and models did not generalize well across annotators. Moreover, a model can exploit those biases to inflate its performance if data from the same annotator is present in the training and testsets.

Even when using novel pre-trained language models, such as BERT, which have more knowledgeable priors due to their pre-training regime, such phenomena persists. (Niven and Kao, 2019)

[1] Our code is publicly available at: https://github.com/nlx-group/Shortcutted-Commonsense-Reasoning.

studied the results from the SemEval 2018 Task 12 – Argument Reasoning Comprehension Task (ARCT) (Habernal et al., 2018). BERT shined, obtaining 77% test accuracy, slightly below untrained humans, who reach 80%. These authors found that the presence of "not" and other high-frequency words such as "is", "do" and "are" was highly correlated with the output, obtaining above random performance with just "not". Adding adversarial examples that counteracted this correlation, the strong spurious signal disappeared, and the performance dropped to random chance.

Data contamination A different type of "cheating" has also attracted some attention recently. Pre-trained language models appear to be able to use (factual) knowledge encoded in its parameters during fine-tuning.In one study, the authors were able to show that in open-book Q&A challenges, large pre-trained language models can be competitive with systems that access external knowledge sources by just accessing their internal "memory" (Roberts et al., 2020).

This memorization power is a problem if the task dataset and the pre-training corpus have been constructed from the same sources, and thus have some text in common.

The authors of GPT-2 (Radford et al., 2019) creates bloom filters from WebText, its pre-training corpus, and calculatse an upper bound for text collisions between downstream tasks and WebText, labelled as data contamination. It was found that due to text overlap, the model gains small but consistent benefits, arguably due to memorization. In the next iteration, GPT-3 (Brown et al., 2020) gave rise to larger-scale data contamination experiments with exact n-gram matching (instead of bloom filters). The results varied, with some datasets being completely clean and others worryingly contaminated.

Popular machine reading datasets such as QuAC (Choi et al., 2018), SQuAD 2 (Rajpurkar et al., 2018) or DROP (Dua et al., 2019) are flagged for >90% contamination. PIQA (Bisk et al., 2020) was flagged with 29% contamination, however, removing the contaminated text only decreases the performance by 3%, regardless of model size. The authors see this as a sign that memorization may not be at play but rather statistical cues in the data, though they did not offer empirically based support to that conjecture.

3 Tasks

We adopt four commonsense related tasks, covering different domains and demands in terms of reasoning, setting a challenging environment to probe the capacity of models addressing them. Figure 1 provides a dataset example for each of the tasks described in this section. Appendix B describes the dataset size for each task.

3.1 Argument Reasoning Comprehension

The Argument Reasoning Comprehension Task (ARCT) (Habernal et al., 2018) aims to test argument reasoning ability, requiring not only language and logic skills but also commonsense knowledge.

An argument is a set of premises/reasons that support a given claim/conclusion and a warrant that establishes the connection between the two (such that the claim follows from the premises) (Toulmin, 1958). Warrants may be implicit, under the assumption that they are shared knowledge (Freeman, 2011). This makes identification of warrants an exercise that requires commonsense.

This task is as follows: given a reason and a claim, from two possible warrants, choose the appropriate one. One of the warrants is a distraction, not supporting the sequitur from reason to claim.

The original dataset has been flagged with problems of spurious correlation, and we will be using its cleaned version (Niven and Kao, 2019).

3.2 AI2 Reasoning Challenge

The AI2 Reasoning Challenge (ARC) (Clark et al., 2018) is a multi-choice natural sciences question answering task, whose dataset is a collection of questions from 3rd to 9th-grade exams, comprising the easy and the challenge sets. The latter contains questions that cannot be trivially solved with token co-occurrence, and our experiments will use it.

ARC requires models to cover knowledge in different formats: definitions, facts & properties, structure, processes & causal, teleology/purpose, algebraic, and many more. It also requires different reasoning types: question logic, linguistic matching, multi-hop, comparison, algebraic, etc. This diversity makes ARC a highly demanding task.

3.3 Physical Interaction Question Answering

The Physical Interaction Question Answering task (PIQA) (Bisk et al., 2020) tests the capabilities of models to answer commonsense questions regarding the physical world. Models will need to

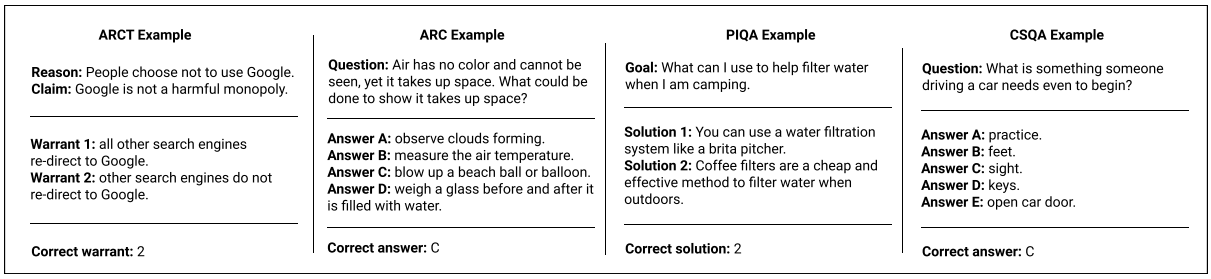

Figure 1: Dataset example for each task.

learn, from raw text only, physical commonsense knowledge.

Humans find this task easy, as they interact with the physical world constantly, manipulating objects and learning about their properties and how they may be used to solve problems. Large scale language models though struggle with this task, with the state-of-the-art achieving 77% accuracy, compared to the human 95% score.

3.4 CommonsenseQA

CommonsenseQA (CSQA) (Talmor et al., 2019) is a multi-choice question answering dataset that targets commonsense knowledge in different formats, much like ARC. It covers a large number of knowledge types: spatial, cause & effect, has parts, is member of, purpose, social, activity, definition and preconditions.

It was built resorting to ConceptNet (Liu and Singh, 2004) for triplets and then using Amazon Mechanical Turk to crowdsource questions.

4 Methodology

In the present paper, the tasks described above are assessed under the experiments described below.

4.1 Experiments

Tasks baselines To have a baseline against which to compare the performance in the stress experiments below, we fine-tune the models on each one of the four tasks.

Partial inputs We perform a stress test consisting in removing certain parts of the input and retrain the task, in line with what was done for instance in MRC (Kaushik and Lipton, 2018), NLI (Gururangan et al., 2018; Poliak et al., 2018) and ARCT (Niven and Kao, 2019). In case there is no substantial degradation in performance, the task can be resolved with partial inputs and this is a strong indicator that the model may be using spurious shortcuts to solve it.

Adversarial attacks Attacks are performed with adversarial test examples that can be obtained from "regular" test examples by means of minimal superficial changes that seek to preserve their semantic value. In case the model performance drops substantially in the face of such attacking examples, this is a symptom of its brittleness and that the expected generalizations were likely not learned, with the model possibly resorting to spurious shortcuts (Ilyas et al., 2019).

To obtain the adversarial examples for this experiment, We resorted to TextFooler (Jin et al., 2020), using the implementation in TextAttack (Morris et al., 2020b).

A study on algorithms to generate adversarial examples found that they may not fully preserve semantics and may introduce up to 38% of grammatical errors (Morris et al., 2020a). To mitigate this, through comparison with human performance, its authors suggest a number of TextFooler's hyper-parameters. One such hyper-parameter is the minimum word cosine similarity in order to consider a given word as a candidate to replace another one and generate an adversarial example — following that study, we set it at 0.9. Another hyper-parameter is the sentence similarity threshold to accept a candidate adversarial example — though in that study, a value of 0.98 is deemed as suitable, here we experimented with a slightly more lenient 0.9.

Data contamination Following the methodology established in the GPT-3 data contamination study (Brown et al., 2020), we search for n-gram collisions between the testsets of the tasks at stake and pre-training datasets of the models, namely BookCorpus (Zhu et al., 2015), English Wikipedia[2], CC-News (Nagel, 2016)[3], OpenWebText (Gokaslan and Cohen, 2019) and STORIES (Trinh and Le, 2018). To cover the neuro-symbolic model, ATOMIC2020 (Hwang et al.,

[2]https://dumps.wikimedia.org/

[3]Extracted with news-please (Hamborg et al., 2017).

2021) was also included in the search for collisions.

Large language models can be prone to memorize previously seen text, which can inflate evaluation scores should a portion of the testset be included in the pre-training regime.

Cross tasks If these models learn commonsense knowledge and reasoning then this should be transferable to other similar tasks in a zero-shot manner. In this experiment, to test its generalization ability, a model trained in one task is tested on every other task in a zero-shot manner.

The more a model has built its strength from spurious cues present in its specific training dataset, the more it will fail on the datasets of the other tasks, given these cues are absent in the latter.

4.2 Models

To widen the net, the baselines are sought with five pre-trained language models. We adopted RoBERTa (Liu et al., 2019b) as an encoder-only exemplar. We used though a different fine-tuning technique, akin to a siamese network, changing the problem into a sequence ranking problem (Liu et al., 2019a) by passing the elements of input pairs separately and producing a value for each, the maximum value being the chosen answer.

GPT-2 (Radford et al., 2019) is selected to feature as a decoder-only examplar. Similar to RoBERTa, a sequence ranking fine-tune approach is followed.

As for an encoder-decoder architecture, we resorted to T5 (Raffel et al., 2020), and also to BART (Lewis et al., 2020), the latter being included as a baseline for the Neuro-Symbolic model.

Concerning a Neuro-Symbolic approach, to inject some finer priors into the language model, we followed the COMET (Bosselut et al., 2019) method, which enriches the model with a commonsense knowledge base CSKG through a generative task. We use COMET(BART), a pre-trained BART-Large model trained on ATOMIC2020 CSKG (Hwang et al., 2021). We adopt the sequence ranking fine-tuning procedure for COMET(BART).

Experiments are based on Huggingface (Wolf et al., 2020), used for pre-trained model weights.

Hyper-parameters were kept to their defaults, except that a sequential hyper-parameter search is performed for the learning rate and batch size. Learning rate is optimized by selecting the model yielding the best dev accuracy score, after fine-tuning for 10 epochs, from the set {1e-3, 1e-4, 1e-5, 2e-3, 2e-4, 2e-5, 3e-3, 3e-4, 3e-5}. Using the learning rate determined in the previous step, an appropriate batch size is determined the same way from the set {4, 8, 16, 32}. Hyper-parameters found are described in Appendix A.

Each model is trained up to 30 epochs. The checkpoint with the best dev accuracy is selected to test with. Due to instability, the reported results are the mean of five runs, each with the different seeds 42, 1128, 1143, 1385 and 1415.

Two of the proposed tasks, PIQA and CSQA, are active competitions and their testsets are private. We report results on the devset for those tasks. To preserve the original distribution of classes, from their training data, we produce two splits: 90% of the data is kept as training data, and 10% is set aside as dev data using stratified splitting.

All experiments were done on a single NVIDIA Titan RTX 24Gb VRAM.

5 Results and Discussion

5.1 Baselines

The baseline performance for the commonsense related tasks is displayed in Table 1. While there is a considerable gap with respect to human performance, some tasks are notably more challenging than others, with mixed results when run with different models. For ARCT, the best scoring model is RoBERTa-Large, having an encouraging gap of only 0.094 accuracy to the human upperbound. CSQA has a gap of 0.156, pretty similar to PIQA.

RoBERTa is the best in two of four tasks, namely ARCT and PIQA, and is a close second best in CSQA, emerging as the most capable reasoner.

It is also interesting to note that the Neuro-symbolic COMET(BART) outperforms BART-Large on all but the CSQA task.

Another interesting contrast concerns ARC and CSQA, which are both multiple-choice problems with up to five possible answers, as the performance in CSQA almost doubles the ARC score. While CSQA covers a wide array of commonsense domains, ARC dataset was obtained from science exams and permits thus to probe more focused and profound knowledge about the physical world, including physics and chemistry laws. It is reasonable then to assume that ARC is a more hard task to solve.

[4] https://www.tau-nlp.org/csqa-leaderboard

	ARCT	ARC	PIQA	CSQA	Params
Random	0.5	0.25	0.5	0.2	-
HUMAN	0.909	N/A	0.949	0.889	-
RoBERTa-Large	**0.815 ± 0.011**	0.411 ± 0.022	**0.789 ± 0.006**	0.733 ± 0.006	355M
GPT2-Medium	0.540 ± 0.071	0.318 ± 0.009	0.706 ± 0.005	0.551 ± 0.012	345M
T5-Large	0.743 ± 0.006	**0.440 ± 0.008**	0.772 ± 0.005	0.713 ± 0.007	770M
BART-Large	0.655 ± 0.154	0.382 ± 0.027	0.777 ± 0.005	**0.738 ± 0.005**	406M
COMET(BART)	0.790 ± 0.005	0.412 ± 0.011	0.783 ± 0.008	0.718 ± 0.008	406M

Table 1: Baselines (accuracy with standard deviation), with best result for each task in bold. Human benchmarks for CSQA obtained from their public leaderboard;[4] for ARCT from (Habernal et al., 2018).

As RoBERTa emerges as the most capable reasoner, we select it for the stress experiments described in the next subsections. Additionally, We select COMET(BART), as its promising and novel neuro-symbolic nature with refined priors offer a better promise to escape the eventual greedy pursuit of data spuriousness to minimize loss.

5.2 Partial inputs

The results from the partial inputs experiment are in Table 2. For both ARCT and PIQA, the scores obtained with partial inputs are pretty close to the scores obtained when using full inputs, which can be taken as a strong indicator that some spurious shortcutting affects these two tasks.

Concerning ARCT, it is interesting to note that in previous work that flagged severe problems of data artifacts in this dataset (Niven and Kao, 2019), the authors noticed that providing just one or two segments of the input was enough for the model to perform above the random baseline. After having cleaned the dataset from these cues, this was not possible any longer when using BERT. We can observe now that this is, however, still possible provided one uses models other than BERT, namely RoBERTa or COMET(BART).

PIQA, in turn, shows the same problem as ARCT. By providing the respective models with only one of the candidate "solutions" as input, this leads to a performance close to the performance observed with the full input for the task. The models are providing solutions for an unknown "goal", yet they are able to perform way above the random baseline. Without being provided with a "goal" in the input in this partial setting, both "solutions" should be equally likely, unless one of the solutions across the different pairs of candidates was always so blatantly nonsensical—which does not happen to be the case—that it is ruled out just off of plain commonsense, which would mean that the model would be performing another type of commonsense reasoning task.

Interestingly, both tasks, ARCT and PIQA, are reported to have gone through pre-processing steps to eliminate statistical lexical cues. In the face of these results, it is not unreasonable, however, to assume that some lexical cues remain, possible of other types, not cleaned yet.

As to the other two tasks, with COMET(BART), ARC is reasonably solved by just looking at the "answers", though this is not happening with RoBERTa-Large, which is in line with the overall superiority of the latter over the former observed in the previous experiment to obtain their baseline scores. Be that as it may, in the other experiments below, further evidence emerges indicating that also ARC may be affected by spurious cues.

Despite CSQA also has a score above random baseline when only using "answers", this happens only by a slim margin. CSQA seems thus to be more resistant dataset in this stress test. Providing just the "question" or just the"answer" leaves the models confused, as it should, resulting in scores in the vicinity of the random score.

5.3 Adversarial attacks

Results from the experiment with adversarial attacks are in Table 3. An example of a successful adversarial example is provided in Figure 2. Both RoBERTa-Large and COMET(BART) show brittleness, with the neuro-symbolic model, despite having been exposed to fine-grained commonsense facts, showing that is not any less susceptible to the attacks than the purely neural RoBERTa-Large.

As to the tasks being experimented with, in contrast with CSQA, with a drop of less than 31%,

	Random	Full inputs	Score	Partial Input	Score
ARCT	0.5	Claim (C) + Reason (R) + Warrant 0 & 1 (W)	0.831$^{\diamondsuit}$ / 0.795$^{\square}$	C+R	0.500$^{\diamondsuit}$ / 0.500$^{\square}$
				R+W	0.500$^{\diamondsuit}$ / 0.500$^{\square}$
				C+W	**0.785**$^{\diamondsuit}$ / **0.782**$^{\square}$
ARC	0.25	Question (Q) + Candidate Answers (A)	0.435$^{\diamondsuit}$ / 0.422$^{\square}$	Q	0.227$^{\diamondsuit}$ / 0.227$^{\square}$
				A	0.245$^{\diamondsuit}$ / **0.344**$^{\square}$
PIQA	0.5	Goal (G) + Solution 1 & 2 (Sol)	0.795$^{\diamondsuit}$ / 0.794$^{\square}$	G	0.495$^{\diamondsuit}$ / 0.495$^{\square}$
				Sol	**0.735**$^{\diamondsuit}$ / **0.724**$^{\square}$
CSQA	0.2	Question (Q) + Candidate Answers (A)	0.738$^{\diamondsuit}$ / 0.727$^{\square}$	Q	0.196$^{\diamondsuit}$ / 0.196$^{\square}$
				A	**0.218**$^{\diamondsuit}$ / 0.184$^{\square}$

Table 2: Results with partial input. Scores above random are in bold. $\diamondsuit$: RoBERTa-Large; $\square$: COMET(BART).

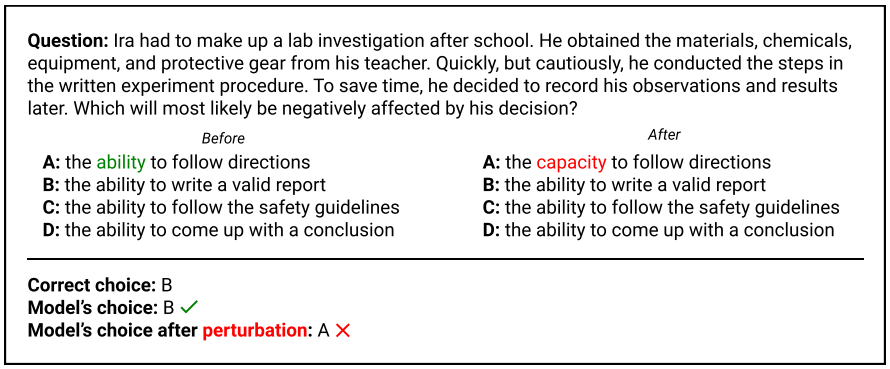

Figure 2: An adversarial example produced with TextFooler on the ARC dataset, targeting a fine-tuned RoBERTa on the task. Simply changing *ability* to *capacity* in option A is enough for the prediction to change.

a drop over 35% in performance is observed in ARCT, ARC and PIQA, the same tasks that were flagged in the previous experiment with partial inputs. This is consistent with the results in the previous section and the assumption that adversarial attacks target non-robust features, present when models learn the task through shortcuts.

5.4 Data contamination

Table 4 displays the statistics concerning data contamination, where a test example is considered dirty if it has n-gram collisions with any of the datasets used for pre-training. The only fully clean dataset is ARCT, which supports one of tasks most affected by data spuriousness in previous experiments. This effectively eliminates memorization as a possible justification for that vulnerability, and since previous work eliminated trivial lexical spurious cues (Niven and Kao, 2019), the plausible explanation about the brittleness of this task is the eventual presence of highly non-linear shortcuts in the ARCT data.

The remaining tasks, in turn, have different levels of contamination. ARC was flagged for 1.19%, followed by CSQA with 5.08%. The most contaminated task is PIQA, with 13.22%.

To further study the eventual impact of contamination on the performance of these three tasks, two sets were created from the testsets/devsets: the "Dirty Set", containing only dirty examples, and the "Clean Set", containing only clean examples. Tables 5 and 6 show their respective performance scores for RoBERTa and COMET(BART), and the deltas for the original testset/devset.

The deltas that were found indicate an almost negligible impact of data contamination as an important factor leveraging model performance, except possibly for ARC that, with slightly higher scores with the Dirty Set, seem to get a marginal benefit from the contamination. Given this lim-

[5]We follow (Brown et al., 2020), where N is defined as the 5th percentile of the distribution of dataset example sizes.

	Model	Before	Δ	Δ%
ARCT	RoB	0.831	-0.355	42.7%
	COM	0.795	-0.283	35.5%
ARC	RoB	0.435	-0.278	63.9%
	COM	0.422	-0.315	74.7%
PIQA	RoB	0.795	-0.489	61.5%
	COM	0.794	-0.508	64.0%
CSQA	RoB	0.738	-0.202	27.4%
	COM	0.727	-0.226	31.3%

Table 3: Results of adversarial attacks. RoB: RoBERTa-Large. COM: COMET(BART).

	N	Clean
ARCT	13	100%
ARC	10	98.81%
CSQA	8	94.92%
PIQA	8	86.78%

Table 4: Data contamination for n-grams of size N.[5]

ited impact, data contamination, also referred to as memorization in the literature, cannot provide the principal explanation for the results obtained in previous stress experiments.

An additional experiment was performed to verify the data contamination between task testsets. It was found that they do not share n-grams between themselves, and thus having no data contamination. This information becomes relevant in the next section.

5.5 Cross tasks

The results of the cross-task experiment are displayed in Table 7.

CSQA stands out positively as providing the best zero-shot approximation to the other tasks. This is in line with the results from previous experiments as it may be seen as further indicating that its higher capacity of generalization beyond the distribution of its training dataset benefits from a lower level of data spuriousness than in the other three tasks. It should be noted also that it is supported by a very general domain dataset, covering a broad range of commonsense dimensions and reasoning types: When applied to other tasks in a zero-shot manner, it does not stray that far from the scores of a specifically fine-tuned model.

	Dirty Set	Clean Set
ARC	0.714 (+0.279)	0.432 (-0.003)
CSQA	0.726 (-0.012)	0.739 (+0.001)
PIQA	0.835 (+0.040)	0.789 (-0.006)

Table 5: RoBERTa accuracy and deltas to best score with full testset.

	Dirty Set	Clean Set
ARC	0.643 (+0.221)	0.420 (+0.002)
CSQA	0.710 (-0.017)	0.727 (+0.000)
PIQA	0.819 (+0.025)	0.790 (-0.004)

Table 6: COMET(BART) accuracy and deltas to best score with full testset.

PIQA, in turn, stands out negatively, as it shows the only case where the zero-shot application of a model to a different task is not faring better than the random baseline of that task, namely when it is applied to solve ARC. While this inferior performance might be seen as being in line with its results in the previous experiments, and possibly another sign of the spuriousness of its dataset, the fact is that PIQA is not faring that bad in its other zero-shot applications, even providing the best zero-shot approximation to CSQA.

ARCT appears to provide the weakest contribution to solving other tasks, which is somewhat expected as it is not an ordinary commonsense task. While requiring commonsense reasoning, the task differs not only in the domain (narrower, covering social topics), but the task itself is different: not so much a Q&A task as the other three, but an argument mining task of warrant identification.

As discussed in the previous section, the testsets for the tasks have no data contamination between themselves, and as such, memorization cannot be the factor that explains these results.

5.6 Searching for possible shortcuts

Additional experiments were performed in the hopes of eventually finding of shortcuts, of two types, that could explain the observed behavior of the models: class imbalance and lexical cues. Class imbalance can be a simple way in which a model exploits the distribution of the dataset to inflate its performance. Lexical cues arise from the particular distribution of certain keywords in the training

	ARCT	ARC	PIQA	CSQA
ARCT	*0.831*	0.310	0.571	0.293
ARC	0.589	*0.435*	0.627	0.343
PIQA	0.597	**0.230**	*0.795*	0.552
CSQA	0.627	0.384	0.687	*0.738*
Random	0.5	0.25	0.5	0.2

Table 7: Cross-task results for RoBERTa-Large. Diagonal values from Table 1. Model trained in rows, tested zero-shot in columns. Values below random baseline in bold.

examples, such that their presence in a candidate answer provides a strong signal for that answer to be predicted in inference time. To detect lexical cues, we resort to the methodology defined in (Niven and Kao, 2019).

No strong shortcuts were found in both cases. The datasets do not suffer from class imbalance, and the lexical cues uncovered do not have enough productivity and coverage to explain the inflated behavior of the models in the previous experiments. We refer the interested reader to Appendix C for more details on these findings. Further searching for possible shortcuts is left for future work, possibly relying on other sort of clues and tools (Branco and Costa, 2008; Branco et al., 2014).

6 Conclusion

Commonsense is a quintessential challenge to Artificial Intelligence and deep learning techniques have been delivering important performance scores and progress in NLP tasks and benchmarks that require commonsense reasoning. In this paper, we set to assess how much of these gains are due to the capacity of generalization beyond the training datasets supporting these tasks and whether these advances may be suffering from the problem that recently has been uncovered in some other areas of NLP, namely that the outstanding models may owe much of their prowess to spurious shortcuts in the data rather than to robust generalizations that permit them to sustain or even approximate their level of performance beyond their benchmark datasets.

To pursue this goal, we focused on a number of tasks involving commonsense reasoning and their datasets that have been widely used in the literature and submit them to stress tests with the potential of helping to uncover data spuriousness problems.

In a first sign that such problems are most probably present, we found that most models do much better than expected without being supplied with all the segments in a given input, even approximating their performance when they are provided with the full input, namely in the case of ARCT, ARC and PIQA tasks, while CSQA shows the expected degraded performance.

Further experiments were consistent with this pattern. Again, the three tasks, ARCT, ARC and PIQA, are also clearly more sensitive to adversarial attacks than the latter. Also CSQA stands out in its generalization capacity in comparison to the lower capacity of the other three tasks to generalize as it provides much better approximation to other tasks in a zero shot setting.

Additionally, with the help of data contamination tests, these contrasts and problems were shown not to result from possible data contamination, leaving one with the most plausible justification that they are rather due to problems of data spuriousness that support shortcutted learning of commonsense reasoning, with reduced generalization capacity.

These results call for future research on a careful review and comparison between the methods and procedures used in the development of these datasets in order to learn how to avoid the pitfalls of producing datasets affected by data spuriousness. But the present results are already very useful in as much as they let one knows that the performance obtained with the datasets studied in the present paper, for which there appear a consistent indication of data spuriousness, should be taken with an extra grain of salt.

Acknowledgements

The research reported here was supported by PORTULAN CLARIN—Research Infrastructure for the Science and Technology of Language https://portulanclarin.net, funded by Lisboa 2020, Alentejo 2020 and FCT—Fundação para a Ciência e Tecnologia under the grant PINFRA/22117/2016.

References

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, pages 7432–7439.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

António Branco and Francisco Costa. 2008. A computational grammar for deep linguistic processing of portuguese: Lxgram. In *Technical Reports*, 17. University of Lisbon, Faculty of Sciences, Department of Informatics.

António Branco, Rodrigues, Silva João, João Ricardo, and Rui Vaz. 2014. Rolling out text categorization for language learning assessment supported by language technology. In *LNAI*, 8775.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Giacomo De Palma, Bobak Toussi Kiani, and Seth Lloyd. 2018. Deep neural networks are biased towards simple functions. *arXiv preprint arXiv:1812.10156*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

James B Freeman. 2011. *Argument Structure:: Representation and Theory*, volume 18. Springer Science & Business Media.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI'21*.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, volume 32, pages 125–136. Curran Associates, Inc.

Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. 2018. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

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

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. *arXiv preprint arXiv:2103.13009*.

Hugo Mercier and Dan Sperber. 2017. *The enigma of reason*. Harvard University Press.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3829–3839.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Sebastian Nagel. 2016. Cc-news.

Timothy Niven and Hung-Yu Kao. 2019. 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. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. 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. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics.

Ke Sun and Frank Nielsen. 2019. Lightlike neuromanifolds, occam's razor and deep learning. *arXiv preprint arXiv:1905.11027*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3878–3887, Online. Association for Computational Linguistics.

Stephen E. Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Guillermo Valle-Pérez, Chico Q Camargo, and Ard A Louis. 2018. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Lei Wu, Zhanxing Zhu, et al. 2017. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*.

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. 2018. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *AAAI*, pages 9733–9740.

Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

A Training Hyper-Parameters

The hyper-parameters found through the search and used to fine-tune each model are provided in Table 8.

B Dataset sizes

The number of examples for each dataset partition of each task is given in Table 9.

C Shortcut Exploration

The results of the two experiments with negative results searching for two different types of shortcuts, mentioned in Section 5.6, are presented here. The first experiment, which investigates class balance of each task dataset, is discussed in Section C.1. Section C.2 covers the search for possible lexical cues present in each task dataset.

C.1 Class Balance

Table 10 contains the statistics of target distribution for each task dataset split.

ARCT, PIQA and CSQA appear to be well balanced. PIQA and CSQA, while not being totally balanced, the difference from the unbalanced classes to random choice is so small that it is likely providing no real advantage to the models.

Task	Model	Batch Size	Learning Rate	Epochs
ARCT	RoBERTa-Large	16	1e-5	25
	GPT2-Medium	8	2e-3	18
	T5	8	2e-5	17
	BART-Large	16	2e-4	12
	COMET(BART)	8	1e-4	25
ARC	RoBERTa-Large	8	1e-4	16
	GPT2-Medium	4	1e-3	26
	T5	8	2e-5	12
	BART-Large	8	1e-4	27
	COMET(BART)	8	3e-5	22
PIQA	RoBERTa-Large	16	3e-3	28
	GPT2-Medium	8	1e-3	22
	T5	8	1e-5	9
	BART-Large	4	1e-3	19
	COMET(BART)	32	3e-4	16
CSQA	RoBERTa-Large	8	3e-4	13
	GPT2-Medium	8	1e-3	14
	T5	8	2e-5	5
	BART-Large	8	3e-4	18
	COMET(BART)	8	1e-4	14

Table 8: Hyper-parameters found through a search used in each experiment.

Task	Train	Dev	Test	Total
ARCT	2420	632	888	3940
ARC	1119	299	1172	3548
PIQA	16113	1838	3084	21035
CSQA	9741	1221	1140	12102

Table 9: Number of examples in each dataset partition for each task. ARC's numbers refer to its Challenge Set, which was used to carry out the experiments.

ARC is slightly unbalanced, albeit not by much. The candidate answer in the first position (labeled 0 in the table) has a diminished presence in the train split. In the same split, the remainder of the candidate answer positions are relatively well balanced and as such it seems that it would be difficult for a model to take a large advantage from this slight unbalance. In the development and test splits, the first answer position is equally underrepresented, and two other positions have a larger presence than the others, creating a slight unbalance.

In spite of this, the unbalance does not seem able to explain the results for ARC on the reported experiments. Since the model learns from the train split and since it is relatively well balanced, it is not expected to provide a useful signal for the model to be exploited during the testing phase.

C.2 Lexical Cues

This section provides a discussion regarding the exploration for lexical cues.

We implemented the metrics developed in (Niven and Kao, 2019) and make use of them to try to uncover lexical cues, namely applicability (α_k), productivity (π_k) and coverage (ξ_k) of cue k. These metrics provide a quantitative measure of how advantageous ngrams present in the dataset can be as shortcutting cues for the models. For this experiment, unigrams and bigrams are considered.

The tasks ARCT, ARC, PIQA and CSQA are framed as multiple choice problems, where the model must choose the correct answer from a set of candidate answers. We apply the metrics to the set of tokens present in each candidate answer, for each example.

The applicability α_k of a cue k (Equation 1) provides a measure of the number of examples where cue k occurs in one candidate answer, but not in any of the others:

$$\alpha_k = \sum_{i=1}^{n} \mathbb{1}\left[\exists j, k \in \mathbb{T}_j^{(i)} \wedge k \notin \mathbb{T}_{\neg j}^{(i)}\right] \quad (1)$$

$\mathbb{1}$ is the indicator function (outputs 1 if the input is true, 0 if not) and $\mathbb{T}_j^{(i)}$ represents the set of tokens in candidate answer j for example i.

Applicability provides the number of examples where a cue's presence is a direct signal to one of the candidate answers but it does not indicate whether that signal offers a correct answer.

Productivity π_k of a cue k (Equation 2) is a measure of the proportion of applicable examples where the cue predicts the correct answer,

$$\pi_k = \frac{\sum_{i=1}^{n} \mathbb{1}\left[\exists j, k \in \mathbb{T}_j^{(i)} \wedge k \notin \mathbb{T}_{\neg j}^{(i)} \wedge y_i = j\right]}{\alpha_k} \quad (2)$$

$y_i = j$ indicates that answer j, which cue k should belong to, is the correct choice for example i. A cue supplies a useful signal if its productivity is above random chance: $\pi_k > 1/m$, where m is the number of possible candidate answers for each example.

Productivity is useful, to understand how broad cue k's presence in the dataset is, to define its coverage ξ_k,

$$\xi_k = \frac{\alpha_k}{n} \quad (3)$$

where n is the total number of examples.

Coverage tells us the proportion of applicable examples with regards to the total number of examples.

ARCT Cues. Table 11 presents the top ten unigram and bigram cues in ARCT's dataset. The largest coverage is achieved with the cue "not", which incidentally was the most helpful cue found in the original ARCT dataset (before having bee cleaned from spurious cues by (Niven and Kao, 2019)), with a coverage of 0.64 and productivity of 0.61 during that revision and balancing of the dataset. In the revised dataset, "not" has a coverage of 0.38 and a productivity of 0.5 (random chance). It is expected that no high coverage useful cues remain in the cleaned dataset, as it was revised with the aid of these metrics. This conjecture is evidenced by the fact that none of the detected cues have a productivity greater than 1/2.

ARC Cues. In Table 12, the top ten unigram and bigram cues present in ARC's dataset are provided. Only 5 useful unigram cues are present in the top ten: "to", "and", "on", "for" and "an". The cues have a relatively low coverage, with the largest being just 0.13 (13% of the dataset), corresponding to cue "to". However, its productivity nears the random choice vlaue (0.25). The remainder of the cues have a coverage of 0.06 and 0.05, respectively, and the largest productivity is 0.41 for the cue "and". This cue is thus a strong signal but because it is such a common function word, it rarely is found in only one of the candidates, translating into a low application and subsequently low coverage due to that.

ARC therefore has useful unigram lexical cues present in its dataset, albeit their coverage and productivity are low, and as such it cannot explain the behavior of the performed experiments.

PIQA Cues. Table 13 displays the top 10 (in coverage) unigram and bigram cues for PIQA's dataset. Four useful unigram and five useful bigram cues are present in that table. Three of the four unigrams are function words. The cue with the largest coverage (0.10) is "a", with the remaining cues having their coverage between 0.07 and 0.01. Overall, the cues have low productivity, nearing the random choice value (0.5) and when accounting for their low coverage as well, it indicates that the dataset does not have a strong signal in the form of lexical cues for models to fully take advantage of.

CSQA Cues. Top 10 unigrams and bigrams for CSQA, in terms of coverage, are shown in Table 14.

As it is observed in ARC's and PIQA's dataset as well, CSQA features the presence of a few cues, although these show low productivity and coverage, such that the models performance cannot be attributed to their presence.

Five unigram and six bigram cues are useful ($\pi_k > 0.2$). The coverage for the bigram cues is just 0.01, while for the unigrams is in the range of 0.07 to 0.03. Only three bigrams have a productivity with a considerable gap from the random choice, with a productivity of 0.27 or greater.

In light of these results analysis, one might conjecture that no widespread, useful cues exist in the datasets. However, it could be that the datasets either:

- Contains other types of lexical cues, e.g. non-linear cues, such as a rule where, say, just for the sake of a rapid illustration, if the word "air" appears in the third position in the sentence, and "water" in the eight position, the

answer is 0.

- Shortcuts at the feature level, such as the presence of certain features in the word embeddings and hidden states, which provide strong signals for certain predictions. These require other types of analysis, different from the ones used in this paper and their detection and interpretation would be far from trivial.

Task	Split	Choice Number	Occurrences	Relative Frequency	Random Chance
ARCT	Train	0	1210	0.500	0.500
		1	1210	0.500	
	Development	0	316	0.500	
		1	316	0.500	
	Test	0	444	0.500	
		1	444	0.500	
ARC	Train	0	239	0.214	0.250
		1	296	**0.265**	
		2	291	**0.260**	
		3	293	**0.262**	
	Development	0	64	0.214	
		1	73	0.244	
		2	78	**0.261**	
		3	83	**0.278**	
		4	1	0.003	
	Test	0	266	0.227	
		1	311	**0.265**	
		2	310	**0.265**	
		3	285	0.243	
PIQA	Train	0	8053	0.500	0.500
		1	8060	0.500	
	Development	0	910	0.495	
		1	928	**0.505**	
CSQA	Train	0	1909	0.196	0.200
		1	1973	**0.203**	
		2	1946	0.200	
		3	1985	**0.204**	
		4	1928	0.198	
	Development	0	239	0.196	
		1	255	**0.209**	
		2	241	0.197	
		3	251	**0.206**	
		4	235	0.192	

Table 10: Class balance for each task dataset split. Relative frequency in bold indicates a frequency above random chance.

Unigrams			Bigrams		
Unigram	Coverage (ξ_k)	Productivity (π_k)	Bigram	Coverage (ξ_k)	Productivity (π_k)
(not,)	0.38	0.5	(is, not)	0.09	0.5
(do,)	0.12	0.5	(are, not)	0.07	0.5
(does,)	0.06	0.5	(do, not)	0.04	0.5
(can,)	0.06	0.5	(can, not)	0.03	0.5
(to,)	0.06	0.5	(does, not)	0.03	0.5
(and,)	0.05	0.5	(not, be)	0.03	0.5
(no,)	0.04	0.5	(is, a)	0.03	0.5
(a,)	0.04	0.5	(can, be)	0.02	0.5
(ca,)	0.04	0.5	(will, not)	0.02	0.5
(be,)	0.04	0.5	(not, a)	0.02	0.5
(more,)	0.03	0.5	(to, be)	0.02	0.5

Table 11: Top 10 unigram and bigram cues with regards to coverage, in descending order, for the ARCT dataset.

Unigrams			Bigrams		
Unigram	Coverage (ξ_k)	Productivity (π_k)	Bigram	Coverage (ξ_k)	Productivity (π_k)
(to,)	**0.13**	**0.26**	(of, the)	0.07	0.15
(in,)	0.13	0.25	(in, the)	0.06	0.24
(of,)	0.13	0.25	(to, the)	0.04	0.24
(a,)	0.11	0.22	(amount, of)	0.03	0.25
(the,)	0.09	0.25	(from, the)	0.03	0.27
(water,)	0.09	0.15	(in, a)	0.03	0.30
(from,)	0.07	0.23	(on, the)	0.03	0.22
(and,)	**0.06**	**0.41**	(the, same)	0.02	0.30
(on,)	**0.06**	**0.26**	(number, of)	0.02	0.16
(for,)	**0.05**	**0.29**	(the, amount)	0.02	0.24
(an,)	**0.05**	**0.29**	(of, a)	0.02	0.25

Table 12: Top 10 unigram and bigram cues with regards to coverage, in descending order, for the ARC dataset. In bold are cues whose productivity $\pi_k > 1/4$, indicating a useful cue.

Unigrams			Bigrams		
Unigram	Coverage (ξ_k)	Productivity (π_k)	Bigram	Coverage (ξ_k)	Productivity (π_k)
(a,)	**0.10**	**0.52**	(in, the)	0.03	0.41
(of,)	0.07	0.50	**(on, the)**	**0.03**	**0.54**
(to,)	0.07	0.49	**(of, the)**	**0.03**	**0.50**
(and,)	**0.07**	**0.52**	(with, a)	0.03	0.47
(in,)	0.06	0.47	**(use, a)**	**0.02**	**0.51**
(on,)	**0.06**	**0.53**	(to, the)	0.02	0.47
(the,)	0.05	0.40	**(in, a)**	**0.02**	**0.50**
(with,)	0.05	0.47	(and, then)	0.02	0.43
(it,)	0.05	0.48	**(into, the)**	**0.01**	**0.52**
(water,)	**0.04**	**0.52**	(top, of)	0.01	0.45
(your,)	0.04	0.45	(the, top)	0.01	0.47

Table 13: Top 10 unigram and bigram cues with regards to coverage, in descending order, for the PIQA dataset. In bold are cues whose productivity $\pi_k > 1/2$, indicating a useful cue.

Unigrams			Bigrams		
Unigram	Coverage (ξ_k)	Productivity (π_k)	Bigram	Coverage (ξ_k)	Productivity (π_k)
(store,)	**0.07**	**0.24**	(go, to)	0.01	0.20
(house,)	0.06	0.19	**(new, york)**	**0.01**	**0.21**
(to,)	0.06	0.17	(grocery, store)	0.01	0.20
(of,)	0.06	0.19	**(have, fun)**	**0.01**	**0.25**
(in,)	0.05	0.12	(talk, to)	0.01	0.06
(office,)	**0.03**	**0.23**	**(office, building)**	**0.01**	**0.25**
(city,)	**0.03**	**0.22**	**(friend, house)**	**0.01**	**0.27**
(room,)	**0.03**	**0.23**	(each, other)	0.01	0.10
(school,)	0.03	0.19	**(neighbor, house)**	**0.01**	**0.28**
(get,)	**0.03**	**0.23**	(living, room)	0.01	0.19
(park,)	0.03	0.16	**(music, store)**	**0.01**	**0.34**

Table 14: Top 10 unigram and bigram cues with regards to coverage, in descending order, for the CSQA dataset. In bold are cues whose productivity $\pi_k > 1/5$, indicating a useful cue.