Misleading Failures of Partial-input Baselines

Shi Feng[1] **Eric Wallace**[2] **Jordan Boyd-Graber**[1]
[1]University of Maryland [2]Allen Institute for Artificial Intelligence
`shifeng,jbg@umiacs.umd.edu` `ericw@allenai.org`

Abstract

Recent work establishes dataset difficulty and removes annotation artifacts via partial-input baselines (e.g., hypothesis-only or image-only models). While the success of a partial-input baseline indicates a dataset is cheatable, our work cautions the converse is not necessarily true. Using artificial datasets, we illustrate how the failure of a partial-input baseline might shadow more trivial patterns that are only visible in the full input. We also identify such artifacts in real natural language inference datasets. Our work provides an alternative view on the use of partial-input baselines in future dataset creation.

1 Dataset Artifacts Hurt Generalizability

Dataset quality is crucial to the development and evaluation of machine learning models. Large-scale natural language processing datasets often rely on crowdsourcing and web crawling, which can introduce *artifacts*. For example, crowdworkers might use specific words to contradict a given premise (Gururangan et al., 2018). These artifacts corrupt the intention of the datasets to model natural language understanding. Annotation artifacts are subtle patterns that are only visible in aggregate on the dataset level. Consequently, they evade human detection and machine learning algorithms, which detect and exploit recurring patterns in large datasets by design, can just as easily use artifacts as real linguistic clues. The resulting models achieve high test accuracy but fail to generalize: for example, they fail under adversarial evaluation (Jia and Liang, 2017; Ribeiro et al., 2018).

Identification of dataset artifacts has changed model evaluation and dataset construction (Chen et al., 2016; Jia and Liang, 2017; Goyal et al., 2017; Zellers et al., 2018). One key identification strategy is partial-input baselines: models that intentionally ignore portions of the input. Examples include hypothesis-only models for natural language inference (Gururangan et al., 2018), question-only models for visual question answering (Goyal et al., 2017), and paragraph-only models for reading comprehension (Kaushik and Lipton, 2018). A dataset is easier than expected if a partial-input baseline performs well. On the other hand, examples where the baseline fails are "hard" (Gururangan et al., 2018), and the failure of partial-input baselines is considered a verdict of a dataset's difficulty (Zellers et al., 2018; Kaushik and Lipton, 2018).

These partial-input analyses are valuable and indeed reveal dataset issues; however, they do not tell the whole story. Just as being free of one ailment is not the same as a clean bill of health, a baseline's failure only indicates that the dataset is not broken in one specific way. There is no reason that artifacts only infect part of the input—models can exploit patterns that are only visible in the full input.

After reviewing of partial-input baselines (Section 2), we construct variants of a natural language inference dataset to highlight the potential pitfalls of partial-input dataset validation (Section 3). Section 4 shows that real datasets have artifacts that cannot be detected by partial-input baselines; we use a hypothesis-plus-one-word model to solve some of the "hard" examples from SNLI (Bowman et al., 2015; Gururangan et al., 2018) where hypothesis-only models fail. We then use k-nearest neighbors to understand how the model learn to exploit these artifacts in the training data. Despite its potential pitfalls, partial-input baselines are still valuable sanity checks; we discuss how it should be used in future dataset creation in Section 5.

2 What are Partial-input Baselines?

A long-term goal of NLP is for models to tackle tasks that we believe require human-level understanding of language. The community typically defines tasks in terms of datasets: reproduce these answers given these inputs, and you have solved the underlying task. This equivalence is only valid when the data accurately represents the task. Unfortunately, verifying this equivalence via humans is fundamentally insufficient: humans reason about examples one by one, while models can discover recurring patterns. Patterns that are not part of the underlying task, or "artifacts" of the data collection process, lead to models that "cheat"—ones that achieve high test accuracy using trivial patterns that do not generalize.

One type of artifact observed in many datasets, specifically classification tasks where each input contains multiple parts (e.g., a question and an image), is a strong correlation between part of the input and the label. For example, a model can answer many VQA questions without looking at the image (Goyal et al., 2017). These artifacts can be detected using partial-input baselines: models that are restricted to using only part of the input.

Validating a dataset with a partial-input baseline has the following steps:

1. Decide which part of the input to use.
2. Reduce all examples in the training set and the test set.
3. Train a new model from scratch on the partial-input training set.
4. Test the model on the partial-input test set.

High accuracy from a partial-input model implies the *original* dataset is solvable (to some extent) in the wrong ways—using patterns that were not intended. This method has identified artifacts in datasets including SNLI (Gururangan et al., 2018; Poliak et al., 2018), VQA (Goyal et al., 2017), EmbodiedQA (Anand et al., 2018), visual dialogue (Massiceti et al., 2018), and visual navigation (Thomason et al., 2018).

3 How Partial-input Baselines Fail

If a partial-input baseline fails—for example, getting close to chance accuracy—one might conclude that the dataset is difficult; for example, partial-input baselines are used to identify the "hard" examples in SNLI and MULTINLI (Gururangan et al., 2018), verify that SQUAD is well constructed (Kaushik and Lipton, 2018) and that SWAG is challenging (Zellers et al., 2018).

Reasonable as it might seem, this kind of argument can be misleading. It is important to understand what exactly these results do and do not imply. Low accuracy from a partial-input baseline only means that the model failed to find exploitable patterns in the visible part of the input. This does not mean, however, that the dataset is free of artifacts—the full input might still contain very trivial patterns.

To illustrate how failures of partial-input baselines might shadow more trivial patterns that are only visible in the full input, we construct two variants of the SNLI dataset (Bowman et al., 2015). The datasets are constructed to contain trivial patterns that are visible in the full input but cannot be exploited by partial-input baselines, i.e., a full-input model can achieve perfect accuracy whereas partial-input models fail.

3.1 Label as Premise

In SNLI, each example consists of a pair of sentences: a premise and a hypothesis. The goal is to classify the semantic relationship between the premise and the hypothesis: either entailment, neutral, or contradiction.

Our first SNLI variant is an extreme example where we introduce artifacts to the dataset that cannot be detected by *some* partial-input baseline. Each SNLI example (training and testing) is copied three times, then each copy is then assigned the label Entailment, Neutral, and Contradiction, respectively. Finally, we set the premise to be the literal word of the associated label: "entailment", "neutral", or "contradiction" (Table 1). From the perspective of a hypothesis-only model, the three copies have identical inputs but conflicting labels, which prevents the model from fitting the training set. Thus the best accuracy from any hypothesis-only model is chance—the baseline fails due to high Bayes error. However, a full-input model can see the label in the premise and achieve perfect accuracy.

This serves as an extreme example of a dataset that passes one partial-input baseline test but still contains artifacts. Obviously, a premise-only baseline can detect these artifacts; we address this in the next variant.

3.2 Label Hidden in Premise and Hypothesis

The artifact we introduce in the previous dataset can be easily detected by a premise-only baseline. In this variant, we "encrypt" the label such that it is only visible if we combine the premise and the hypothesis, i.e., neither premise-only nor hypothesis-only baselines can detect the artifact. Each label

Old Premise	Animals are running
New Premise	Entailment
Hypothesis	Animals are outdoors
Label	Entailment

Table 1: Each example in this dataset has the groundtruth label as the premise. Because each hypothesis occurs in the dataset three times with a different label each time (not shown in this table), no hypothesis-only baseline can achieve better than chance accuracy. However, a full-input model can trivially solve the dataset.

Label	**Combinations**		
Entailment	A+B	C+D	E+F
Contradiction	A+F	C+B	E+D
Neutral	A+D	C+F	E+B

Table 2: We "encrypt" the labels to mimic the exploitable patterns that requires both parts of the input. Each capital letter is a code word, and each label can be represented as one of three combinations of two code words. Each combination uniquely identifies a label—for example, *A* in the premise and *B* in the hypothesis equals Entailment. However, a single code word cannot—one cannot infer the label by only seeing *A* in the premise.

is represented by the concatenation of two code words, and the mapping is one-to-many: each label has three combinations, and each combination uniquely identifies a label. The design of the code words (Table 2) ensure that one code word cannot uniquely identify a label—you need both.

We put one code word in the premise and the other in the hypothesis. These encrypted labels mimic the exploitable patterns that require both parts of the input. The most extreme version of this dataset has the nine combinations in Table 2 as both the training set and the test set.

Because a single code word cannot identify the label, neither hypothesis-only nor premise-only baselines can achieve more than chance accuracy (one-third chance). However, a full-input model can still easily learn to extract the label by combining the premise and the hypothesis and achieve perfect accuracy.

4 Artifacts Undetected by Partial-input Baselines

Our synthetic datasets are trivially solvable but partial-input baselines fail to detect the artifacts. Do real datasets such as SNLI have artifacts that cannot be detected by partial-input baselines?

Additional information about the premise should make it easier to solve examples that are unsolvable for a hypothesis-only model. If the added features appear useless to humans but allow the hypothesis-only model to improve accuracy, they are artifacts instead of generalizable patterns.

We showcase using a very limited premise feature—only the last noun—to form a hypothesis-plus-one-word model. We start with a BERT-based classifier that gets 88.28% accuracy with regular, full input. The hypothesis-only version reaches 70.10% accuracy.[1] With hypothesis-plus-one-word, the accuracy improves to 74.6% and the model solves 15% of the "hard" examples, all of which are unsolvable by the hypothesis-only model.[2]

In Table 3 we show examples that are only solvable with the additional one word from the premise. Following Papernot and McDaniel (2018), we extract training examples by nearest neighbor search in the final BERT representation space, for both hypothesis-only and hypothesis-plus-one-word models. In the first example, humans would not judge "The young boy is crying" as a contradiction to "camera", which is the premise seen by the hypothesis-plus-one-word model; without the additional word, nearest neighbor search returns examples with the incorrect Entailment label, but with the additional word "camera" as premise, we get instead training examples with label Contradiction. This added pattern by including one premise word is an artifact that regular partial-input baselines cannot detect, but it can be exploited by a full-input model.

5 Discussion and Related Work

Partial-input baselines are valuable sanity checks for complex NLP datasets, but as we illustrated, their implications should be understood carefully. Going one step further, we discuss not only methods for creating datasets with fewer artifacts but also empirical results that corroborate the potential pitfalls we suggest in this paper. We also discuss some alternative approaches to robust NLP models.

As we illustrate with synthetic and real datasets, each partial-input test can only verify that the dataset is not broken in one specific way. A more

[1] Gururangan et al. (2018) report 67.0% using a simpler hypothesis-only model.

[2] The easy-hard split of the dataset is done with our own model, not the one released by Gururangan et al. (2018).

Label	Premise	Hypothesis
Contradiction	A young boy hanging on a pole smiling at the camera.	The young boy is crying.
Contradiction	A boy smiles tentatively at the camera.	a boy is crying.
Contradiction	A happy child smiles at the camera.	The child is crying at the playground.
Contradiction	A girl shows a small child her camera.	A boy crying.
Entailment	A little boy with a baseball on his shirt is crying.	A boy is crying.
Entailment	Young boy crying in a stroller.	A boy is crying.
Entailment	A baby boy in overalls is crying.	A boy is crying.
Entailment	Little boy playing with his toy train.	A boy is playing with toys.
Entailment	A little boy is looking at a toy train.	A boy is looking at a toy.
Entailment	Little redheaded boy looking at a toy train.	A little boy is watching a toy train.
Entailment	A young girl in goggles riding on a toy train.	A girl rides a toy train.
Contradiction	A little girl is playing with tinker toys.	A little boy is playing with toys.
Contradiction	A toddler shovels a snowy driveway with a shovel.	A young child is playing with toys.
Contradiction	A boy playing with toys in a bedroom.	A boy is playing with toys at the park.

Table 3: SNLI test examples (highlighted) that are unsolvable for the hypothesis-only model but can be solved when a single word in the premise (underlined) is added. We also show the training examples that are nearest neighbors to the test example in BERT's representation space. Underlines indicate parts of the input that are visible to the model. With the additional last noun in the premise, training examples with the same label are retrieved; with only the hypothesis, examples with the incorrect label are returned.

complete validation of the dataset requires us to list more ways that a model can cheat, but it is impossible to list all of them. Can we prevent the model from cheating by creating datasets with fewer artifacts?

Adversarial Annotation A natural next step is to incorporate these baselines into the data generation process. One notable example of a dataset that uses adversarial annotation is SWAG (Zellers et al., 2018), where multiple-choice answers are selected adversarially against an ensemble of classifiers. However, since the adversaries (trained normally) can be easily fooled if they rely on superficial patterns, these supposedly challenging examples still contain artifacts, which can be exploited by a stronger model, e.g. BERT. This annotation paradigm leads to datasets that are *just difficult enough* to fool the baselines but not enough to ensure that no model can cheat.

Adversarial Evaluation Switching our focus from dataset to models, adversarial evaluation is vital to understanding a system's capabilities, as strikingly simple model limitations can be overlooked (Belinkov and Bisk, 2018; Jia and Liang, 2017). For instance, simple paraphrases can fool textual entailment and visual question answering systems (Iyyer et al., 2018; Ribeiro et al., 2018), while common typos drastically degrade neural machine translation quality (Belinkov and Bisk, 2018).

Interpretations We can also try to understand directly what the model is doing using interpretations. But there is a problem of faithfulness (Rudin, 2018). The nature of interpretation is that we approximate (often locally) a complex model (often neural networks) with a much simpler, inherently interpretable model (often linear models). Because the interpretation is an approximation, it can never be completely faithful: there must be cases where the original model and the simple model behave differently, and these cases might be especially important as they usually reflect the counter-intuitive brittleness of the complex models (e.g., in adversarial examples).

Certifiable Robustness In computer vision, the research on robustness is transitioning from an empirical arm race between attacks and defenses to more theoretically sound **certifiable and provable robustness** methods. Despite their strong empirical results and theoretical guarantees, direct adaptation of these methods to natural language tasks is still an open problem due to the discrete nature of text inputs.

6 Conclusion

Partial-input baselines are valuable sanity checks of dataset difficulty, but their implications need to be analyzed carefully. We illustrate in both synthetic and real datasets how these experiments can shadow trivial, exploitable patterns that require the full input. Our work provides an alternative view on the use of partial-input baselines in future dataset creation.

References

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. 2018. Blindfold baselines for embodied qa. In *NeurIPS Visually-Grounded Interaction and Language Workshop*.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *Proceedings of the International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of Empirical Methods in Natural Language Processing*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the Association for Computational Linguistics*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Computer Vision and Pattern Recognition*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke S. Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of Empirical Methods in Natural Language Processing*.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of Empirical Methods in Natural Language Processing*.

Daniela Massiceti, Puneet K. Dokania, N. Siddharth, and Philip H.S. Torr. 2018. Visual dialogue without vision or dialogue. In *NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*.

Nicolas Papernot and Patrick D. McDaniel. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv: 1803.04765*.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *7th Joint Conference on Lexical and Computational Semantics (*SEM)*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the Association for Computational Linguistics*.

Cynthia Rudin. 2018. Please stop explaining black box models for high stakes decisions. In *NIPS 2018 Workshop on Critiquing and Correcting Trends in Machine Learning*.

Jesse Thomason, Daniel Gordan, and Yonatan Bisk. 2018. Shifting the baseline: Single modality performance on visual navigation & qa. *arXiv preprint arXiv:1811.00613*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of Empirical Methods in Natural Language Processing*.