Misleading Failures of Partial-input Baselines

Shi Feng
Computer Science
University of Maryland
shifeng@umiacs.umd.edu

Eric Wallance
Allen Institute for
Artificial Intelligence
ericw@allenai.org

Jordan Boyd-Graber
Computer Science, iSchool,
UMIACS, and LSC
University of Maryland
jbg@umiacs.umd.edu

Abstract

Recent work establishes dataset difficulty and removes annotation artifacts via partial-input baselines (e.g., hypothesis-only models for SNLI or question-only models for VQA). When a partial-input baseline gets high accuracy, a dataset is cheatable. However, the converse is not necessarily true: the failure of a partial-input baseline does not mean a dataset is free of artifacts. To illustrate this, we first design artificial datasets which contain trivial patterns in the full input that are undetectable by any partial-input model. Next, we identify such artifacts in the SNLI dataset—a hypothesis-only model augmented with trivial patterns in the premise can solve 15% of the examples that are previously considered "hard". Our work provides a caveat for the use of partial-input baselines for dataset verification and creation.

1 Dataset Artifacts Hurt Generalizability

Dataset quality is crucial for the development and evaluation of machine learning models. Large-scale natural language processing (NLP) datasets often use human annotations on web-crawled data, 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 train and evaluate models for natural language understanding. Importantly, a human inspection of individual examples cannot catch artifacts because they are only visible in aggregate on the dataset level. However, machine learning algorithms, which detect and exploit recurring patterns in large datasets by design, can just as easily use artifacts as real linguistic clues. As a result, models trained on these datasets can achieve high test accuracy by exploiting artifacts but fail to generalize, e.g., they fail under adversarial evaluation (Jia and Liang, 2017; Ribeiro et al., 2018).

The identification of dataset artifacts has changed model evaluation and dataset construction (Chen et al., 2016; Jia and Liang, 2017; Goyal et al., 2017). One key method is to use partial-input baselines, i.e., models that intentionally ignore portions of the input. Example use cases 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 successful partial-input baseline indicates that a dataset contains artifacts which make it easier than expected. On the other hand, examples where this 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 a 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 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 evade partial-input baselines; we use a hypothesis-plus-one-word model to solve 15% of the "hard" examples from SNLI (Bowman et al., 2015; Gururangan et al., 2018) where hypothesis-only models fail. Furthermore, we highlight some of the artifacts learned by this model using k-nearest neighbors in representation space. Section 5 discusses how partial-input baselines should be used in future dataset creation and analysis.

2 What are Partial-input Baselines?

A long-term goal of NLP is to solve tasks that we believe require a human-level understanding of language. The NLP community typically defines tasks with datasets: reproduce these answers given these inputs, and you have solved the underlying task. This task-dataset equivalence is only valid when the dataset 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, can lead to models that "cheat"—ones that achieve high test accuracy using patterns that do not generalize.

One frequent type of artifact, especially in classification datasets where each input contains multiple parts (e.g., a question and an image), is a strong correlation between a 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, i.e., using unintended patterns. Partial-input baselines have identified artifacts in many datasets, e.g., 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., 2019).

3 How Partial-input Baselines Fail

If a partial-input baseline fails, e.g., it gets close to chance accuracy, one might conclude that a dataset is difficult. For example, partial-input baselines are used to identify the "hard" examples in SNLI (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. A low accuracy from a partial-input baseline only means that the model failed to confirm a specific exploitable pattern in the part of the input that the model can see. This does not mean, however, that the dataset is free of artifacts—the full input might still contain very trivial patterns.

To illustrate how the 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 partial-input baselines cannot exploit, i.e., the patterns are only visible in the full input. As a result, a full-input 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 of artifacts that cannot be detected by a hypothesis-only baseline. Each SNLI example (training and testing) is copied three times, and the copies are assigned the labels Entailment, Neutral, and Contradiction, respectively. We then set each example's 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. Thus, the best accuracy from any hypothesis-only model is chance—the model 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 a partial-input baseline test but still contains artifacts. Obviously, a premise-only baseline can detect these artifacts; we address this in the next dataset 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-

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

Table 1: Each example in this dataset has the ground-truth label set as the premise. Every hypothesis occurs three times in the dataset, each time with a unique label and premise combination (not shown in this table). Therefore, a hypothesis-only baseline will only achieve chance accuracy, but 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 an artifact that requires both parts of the input. Each capital letter is a code word, and each label is derived from the combination of two code words. Each combination uniquely identifies a label, e.g., 𝔸 in the premise and 𝔹 in the hypothesis equals Entailment. However, a single code word cannot identify the label.

only baselines can detect the artifact. Each label is represented by the concatenation of two "code words", and this mapping is one-to-many: each label has three combinations of code words, and each combination uniquely identifies a label. Table 2 shows our code word configuration. The design of the code words ensures that a single 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 an artifact that requires both parts of the input. Table 3 shows an SNLI example modified accordingly. A full-input model can exploit the artifact and trivially achieve perfect accuracy, but a partial-input model cannot.

A more extreme version of this modified dataset has exactly the nine combinations in Table 2 as both the training set and the test set. Since a single code word cannot identify the label, neither hypothesis-only nor premise-only baselines can achieve more than chance accuracy. However, a full-input model can perfectly extract the label by combining the premise and the hypothesis.

Premise	𝔸 Animals are running
Hypothesis	𝔹 Animals are outdoors
Label	Entailment

Table 3: Each example in this dataset has a code word added to both the premise and the hypothesis. Following the configuration of Table 2, 𝔸 in the premise combined with 𝔹 in the hypothesis indicates the label is Entailment. A full-input model can easily exploit this artifact but partial-input models cannot.

4 Artifacts Evade Partial-input Baselines

Our synthetic dataset variants contain trivial artifacts that partial-input baselines fail to detect. Do real datasets such as SNLI have artifacts that are not detected by partial-input baselines?

We investigate this by providing additional information about the premise to a hypothesis-only model. In particular, we provide the last noun of the premise, i.e., we form a hypothesis-plus-one-word model. Since this additional information appears useless to humans (examples below), it is an artifact rather than a generalizable pattern.

We use a BERT-based (Devlin et al., 2019) classifier that gets 88.28% accuracy with the regular, full input. The hypothesis-only version reaches 70.10% accuracy.[1] With the hypothesis-plus-one-word model, the accuracy improves to 74.6%, i.e., the model solves 15% of the "hard" examples that are unsolvable by the hypothesis-only model.[2]

Table 4 shows examples that are only solvable with the one additional word from the premise. For both the hypothesis-only and hypothesis-plus-one-word models, we follow Papernot and McDaniel (2018) and Wallace et al. (2018) and retrieve training examples using nearest neighbor search in the final BERT representation space. In the first example, humans would not consider the hypothesis "The young boy is crying" as a contradiction to the premise "camera". In this case, the hypothesis-only model incorrectly predicts Entailment, however, the hypothesis-plus-one-word model correctly predicts Contradiction. This pattern—including one premise word—is an artifact that regular partial-input baselines cannot detect but can be exploited by a full-input model.

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

[2] We create the easy-hard split of the dataset using our model, not using the model from 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 4: We create a hypothesis-plus-one-word model that sees the hypothesis alongside the last noun in the premise. We show two SNLI test examples (highlighted) that are answered correctly using this model but are answered incorrectly using a hypothesis-only model. For each test example, we also show the training examples that are nearest neighbors in BERT's representation space. When using the hypothesis and the last noun in the premise (underlined), training examples with the correct label are retrieved; when using only the hypothesis, examples with the incorrect label are retrieved.

5 Discussion and Related Work

Partial-input baselines are valuable sanity checks for datasets, but as we illustrate, their implications should be understood carefully. This section discusses methods for validating and creating datasets in light of possible artifacts from the annotation process, as well as empirical results that corroborate the potential pitfalls highlighted in this paper. Furthermore, we discuss alternative approaches for developing robust NLP models.

Hypothesis Testing Validating datasets with partial-input baselines is a form of hypothesis-testing: one hypothesizes trivial solutions to the dataset (i.e., a spurious correlation between labels and a part of the input) and verifies if these hypotheses are true. While it is tempting to hypothesize other ways a model can cheat, it is infeasible to enumerate over all of them. In other words, if we could write down all the necessary tests for *test-driven development* (Beck, 2002) of a machine learning model, we would already have a rule-based system that can solve our task.

Adversarial Annotation Rather than using partial-input baselines as post-hoc tests, a natural idea is to incorporate them into the data generation process to reject bad examples. For example, the SWAG (Zellers et al., 2018) dataset consists of multiple-choice answers that are selected adversarially against an ensemble of partial-input and heuristic classifiers. However, since these classifiers can be easily fooled if they rely on superficial patterns, the resulting dataset may still contain artifacts. In particular, a much stronger model (BERT) that sees the full-input easily solves the dataset. This demonstrates that using partial-input baselines as adversaries may lead to datasets that are *just difficult enough* to fool the baselines but not difficult enough to ensure that no model can cheat.

Adversarial Evaluation Instead of validating a dataset, one can alternatively probe the model directly. For example, models can be stress tested using adversarial examples (Jia and Liang, 2017; Wallace et al., 2019) and challenge sets (Glockner et al., 2018; Naik et al., 2018). These tests can reveal strikingly simple model limitations, e.g., basic 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 Another technique for probing models is to use interpretation methods. Interpretations, however, have a problem of faithfulness (Rudin, 2018): they approximate (often locally) a complex model with a simpler, interpretable model (often a linear model). Since interpretations are inherently an approximation, they can never be completely faithful—there are cases where the original model and the simple model behave differently (Ghorbani et al., 2019). These

cases might also be especially important as they usually reflect the counter-intuitive brittleness of the complex models (e.g., in adversarial examples).

Certifiable Robustness Finally, an alternative approach for creating models that are free of artifacts is to alter the training process. In particular, model robustness research in computer vision has begun to transition from an empirical arms race between attackers and defenders to more theoretically sound robustness methods. For instance, convex relaxations can train models that are provably robust to adversarial examples (Raghunathan et al., 2018; Wong and Kolter, 2018). Despite these method's impressive (and rapidly developing) results, they largely focus on adversarial perturbations bounded to an L_∞ ball. This is due to the difficulties in formalizing attacks and defenses for more complex threat models, of which the discrete nature of NLP is included. Future work can look to generalize these methods to other classes of model vulnerabilities and artifacts.

6 Conclusion

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

Acknowledgments

This work was supported by NSF Grant IIS-1822494. Boyd-Graber and Feng are also supported by DARPA award HR0011-15-C-0113 under subcontract to Raytheon BBN Technologies. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsor.

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*.

Kent Beck. 2002. *Test-Driven Development by Example*. Addison-Wesley.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Amirata Ghorbani, Abubakar Abid, and James Y. Zou. 2019. Interpretation of neural networks is fragile. In *Association for the Advancement of Artificial Intelligence*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. 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*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of International Conference on Computational Linguistics*.

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)*.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified defenses against adversarial examples. In *Proceedings of the International Conference on Learning Representations*.

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 *NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning*.

Jesse Thomason, Daniel Gordan, and Yonatan Bisk. 2019. Shifting the baseline: Single modality performance on visual navigation & QA. In *Conference of the North American Chapter of the Association for Computational Linguistics*.

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. 2018. Interpreting neural networks with nearest neighbors. In *EMNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. In *Transactions of the Association for Computational Linguistics*.

Eric Wong and J. Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the International Conference of Machine Learning*.

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*.