

NeurIPS 2021 Datasets and Benchmarks Track

<https://neurips.cc/Conferences/2021/CallForDatasetsBenchmarks>

Motivation

- There are no good models without good data
 - many algorithms are only evaluated on toy problems
 - data that is plagued with bias, which could lead to biased models or misleading results
- Lack of papers on this topic
 - ~5 accepted papers per year focus on proposing new datasets
 - ~10 accepted papers per year focus on systemic benchmarking
- Datasets and benchmarks require their own publishing and reviewing guidelines

<https://neuripsconf.medium.com/announcing-the-neurips-2021-datasets-and-benchmarks-track-644e27c1e66c>

<https://research.google/pubs/pub49953/>

Not only NIPS: ACL-2021 Workshop on Benchmarking

- What important technologies and underlying sciences need to be fostered, now and in the future?
- In each case, are there existing tasks/benchmarks that move the field in the right direction?
- Where are there gaps?
- For the gaps, are there initial steps that are accessible, attractive, and cost effective?
- How large should a benchmark be?
 - How much data do we need to measure significant differences?
 - How much data do machines need to obtain good performance?
 - How much data do babies need to learn language?

Related publications

- [Datasheets for Datasets](#) (Cited by 409, 11-08-2021)
 - The machine learning community currently has no standardized process for documenting datasets, which can lead to severe consequences in high-stakes domains. To address this gap, we propose datasheets for datasets. In the electronics industry, every component, no matter how simple or complex, is accompanied with a datasheet that describes its operating characteristics, test results, recommended uses, and other information. By analogy, we propose that every dataset be accompanied with a datasheet that documents its motivation, composition, collection process, recommended uses, and so on. Datasheets for datasets will facilitate better communication between dataset creators and dataset consumers, and encourage the machine learning community to prioritize transparency and accountability.
- [Confident Learning: Estimating Uncertainty in Dataset Labels](#) (Cited by 64, 11-08-2021)
 - Learning exists in the context of data, yet notions of confidence typically focus on model predictions, not label quality. Confident learning (CL) is an alternative approach which focuses instead on label quality by characterizing and identifying label errors in datasets, based on the principles of pruning noisy data, counting with probabilistic thresholds to estimate noise, and ranking examples to train with confidence. Whereas numerous studies have developed these principles independently, here, we combine them, building on the assumption of a class-conditional noise process to directly estimate the joint distribution between noisy (given) labels and uncorrupted (unknown) labels.
- [Data and its \(dis\)contents: A survey of dataset development and use in machine learning research](#) (Cited by 27, 11-08-2021)
 - Datasets have played a foundational role in the advancement of machine learning research. They form the basis for the models we design and deploy, as well as our primary medium for benchmarking and evaluation. Furthermore, the ways in which we collect, construct and share these datasets inform the kinds of problems the field pursues and the methods explored in algorithm development. However, recent work from a breadth of perspectives has revealed the limitations of predominant practices in dataset collection and use. In this paper, we survey the many concerns raised about the way we collect and use data in machine learning and advocate that a more cautious and thorough understanding of data is necessary to address several of the practical and ethical issues of the field.
- ["Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI](#) (Cited by 25, 11-08-2021)
 - AI models are increasingly applied in high-stakes domains like health and conservation. Data quality carries an elevated significance in high-stakes AI due to its heightened downstream impact, impacting predictions like cancer detection, wildlife poaching, and loan allocations. Paradoxically, data is the most under-valued and de-glamorised aspect of AI. In this paper, we report on data practices in high-stakes AI, from interviews with 53 AI practitioners in India, East and West African countries, and USA. We define, identify, and present empirical evidence on Data Cascades—compounding events causing negative, downstream effects from data issues—triggered by conventional AI/ML practices that undervalue data quality. Data cascades are pervasive (92% prevalence), invisible, delayed, but often avoidable. We discuss HCI opportunities in designing and incentivizing data excellence as a first-class citizen of AI, resulting in safer and more robust systems for all.

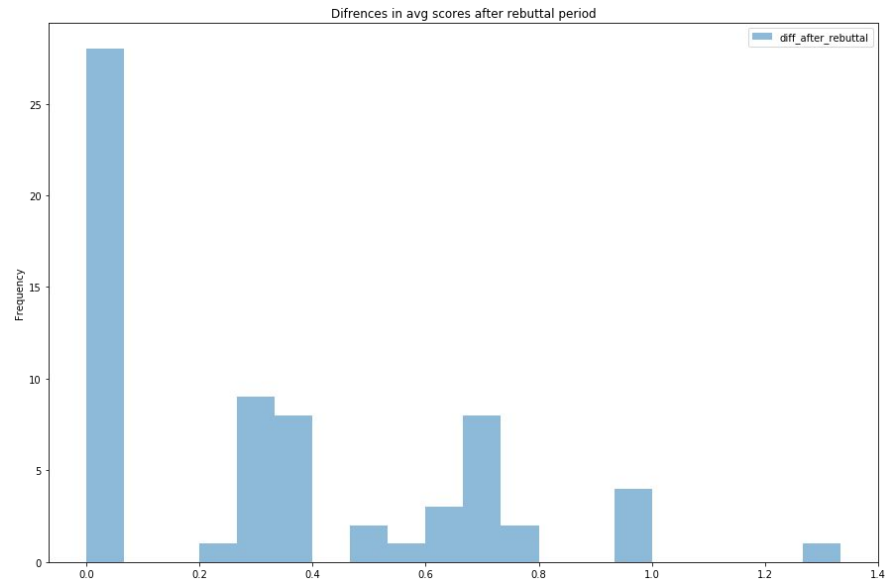
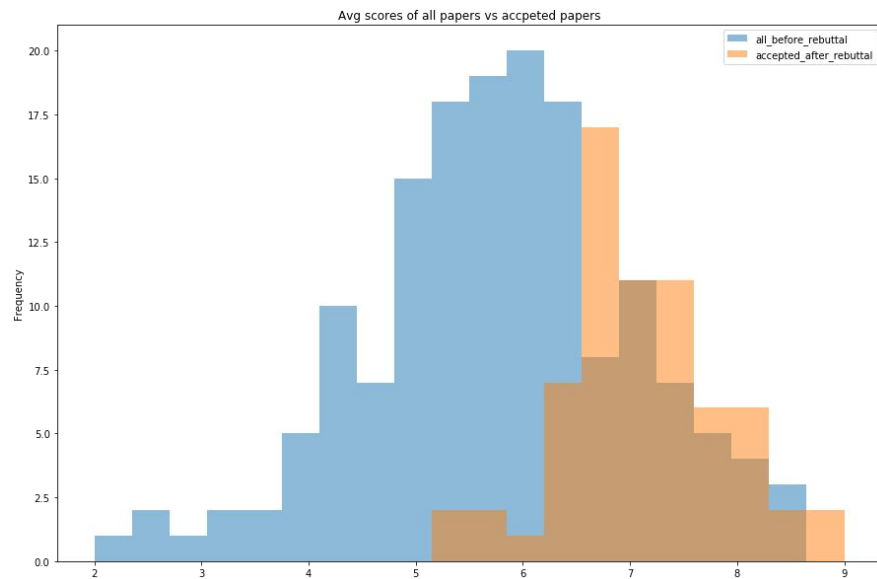
Submission guidelines

- [Paper checklist](#)
 - negative societal impacts
 - ethics review guidelines
 - ...
- For new datasets
 - Dataset documentation and intended uses. Recommended documentation frameworks include [datasheets for datasets](#), [dataset nutrition labels](#), [data statements for NLP](#), and [accountability frameworks](#).
 - ...
- For new benchmarks
 - The supplementary materials must ensure that all results are easily reproducible. Where possible, use a reproducibility framework such as the [ML reproducibility checklist](#), or otherwise guarantee that all results can be easily reproduced, i.e. all necessary datasets, code, and evaluation procedures must be accessible and documented.

Submission summary

- Total submissions: 165
- Number of accepted papers: 67
- Acceptance rate: 40.06 %
- Max reviewer avg score: 9 ([link](#))
- Min reviewer avg score: 2
- Max avg score improvement after rebuttal: +1.33 ([pub](#)), +1 ([pub 1](#), [pub 2](#), [pub 3](#))
- Threshold before rebuttal period to be sure that paper will be accepted: 6.4

Statistics - avg reviewers scores



Papers with interesting ideas

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks (8.33 avg reviewers score)

We identify label errors in the test sets of 10 of the most commonly-used computer vision, natural language, and audio datasets, and subsequently study the potential for these label errors to affect benchmark results. Errors in test sets are numerous and widespread: we estimate an average of 3.3% errors across the 10 datasets, where for example 2916 label errors comprise 6% of the ImageNet validation set. Putative label errors are identified using **confident learning** algorithms and then human-validated via crowdsourcing (54% of the algorithmically-flagged candidates are indeed erroneously labeled). Traditionally, machine learning practitioners choose which model to deploy based on test accuracy — our findings advise caution here, proposing that judging models over correctly labeled test sets may be more useful, especially for noisy real-world datasets

<https://openreview.net/pdf?id=XccDXrDNLeK>

<https://arxiv.org/pdf/1911.00068.pdf>

<https://github.com/cgnorthcutt/cleanlab>

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks

Table 1: Test set errors are prominent across common benchmark datasets. Errors are estimated using confident learning (CL) and validated by human workers on Mechanical Turk.

Dataset	Modality	Size	Model	Test Set Errors				
				CL guessed	MTurk checked	validated	estimated	% error
MNIST	image	10,000	2-conv CNN	100	100 (100%)	15	-	0.15
CIFAR-10	image	10,000	VGG	275	275 (100%)	54	-	0.54
CIFAR-100	image	10,000	VGG	2,235	2,235 (100%)	585	-	5.85
Caltech-256	image	29,780	Wide ResNet-50-2	2,360	2,360 (100%)	458	-	1.54
ImageNet*	image	50,000	ResNet-50	5,440	5,440 (100%)	2,916	-	5.83
QuickDraw	image	50,426,266	VGG	6,825,383	2,500 (0.04%)	1870	5,105,386	10.12
20news	text	7,532	TFIDF + SGD	93	93 (100%)	82	-	1.11
IMDB	text	25,000	FastText	1,310	1,310 (100%)	725	-	2.9
Amazon Reviews	text	9,996,437	FastText	533,249	1,000 (0.2%)	732	390,338	3.9
AudioSet	audio	20,371	VGG	307	307 (100%)	275	-	1.35














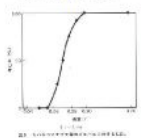






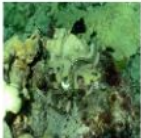

*Because the ImageNet test set labels are not publicly available, the ILSVRC 2012 validation set is used.

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks

Table 2: Mechanical Turk validation confirming the existence of pervasive label errors and categorizing the types of label issues.

Dataset	Test Set Errors Categorization					
	non-errors	errors	non-agreement	correctable	multi-label	neither
MNIST	85	15	2	10	-	3
CIFAR-10	221	54	32	18	0	4
CIFAR-100	1650	585	210	318	20	37
Caltech-256	1902	458	99	221	115	23
ImageNet	2524	2916	598	1428	597	293
QuickDraw	630	1870	563	1047	20	240
20news	11	82	43	22	12	5
IMDB	585	725	552	173	-	-
Amazon Reviews	268	732	430	302	-	-
AudioSet	32	275	-	-	-	-

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks

	MNIST	CIFAR-10	CIFAR-100	Caltech-256	ImageNet	QuickDraw
correctable	 given: 5 corrected: 3	 given: cat corrected: frog	 given: lobster corrected: crab	 given: dolphin corrected: kayak	 given: white stork corrected: black stork	 given: tiger corrected: eye
multi-label	(N/A)	(N/A)	 given: hamster also: cup	 given: laptop also: people	 given: mantis also: fence	 given: hat also: flying saucer
neither	 given: 6 alt: 1	 given: deer alt: bird	 given: rose alt: apple	 given: house-fly alt: ladder	 given: polar bear alt: elephant	 given: pineapple alt: raccoon
non-agreement	 given: 4 alt: 9	 given: deer alt: frog	 given: spider alt: cockroach	 given: yo-yo alt: frisbee	 given: eel alt: flatworm	 given: bandage alt: roller coaster

<https://openreview.net/pdf?id=XccDXrDNLeK>

<https://labelerrors.com/>

Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks

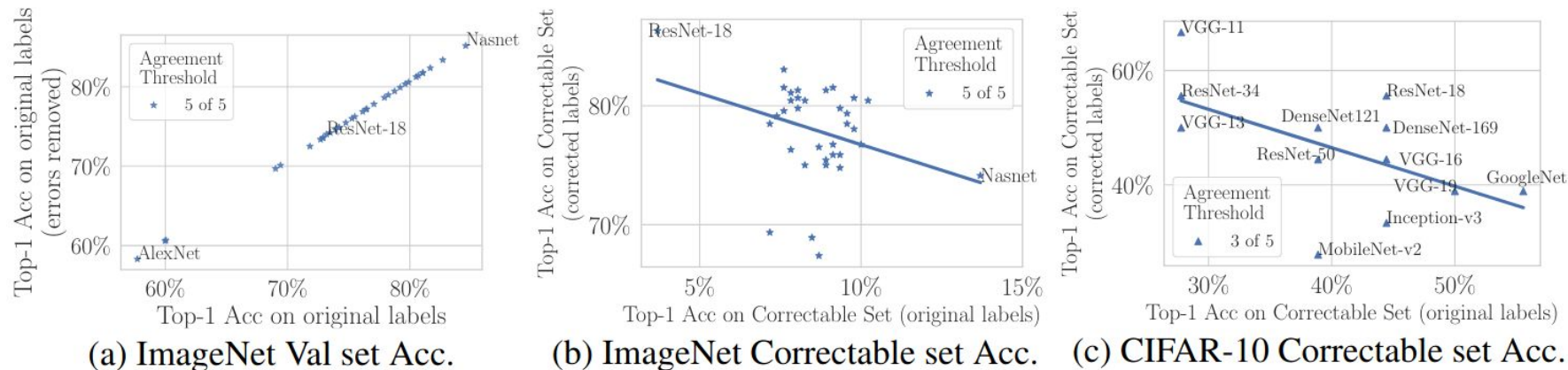


Figure 3: Benchmark ranking comparison of 34 models pre-trained on ImageNet and 13 pre-trained on CIFAR-10 (more details in Tables S2 and S1 and Fig. S2, in the Appendix). Benchmarks are unchanged by removing label errors (a), but change drastically on the Correctable set with original (erroneous) labels versus corrected labels, e.g. Nasnet: 1/34 \rightarrow 29/34, ResNet-18: 34/34 \rightarrow 1/34.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification (8 avg reviewers score)

Constructing benchmarks that test the abilities of modern natural language understanding models is difficult – pre-trained language models exploit artifacts in benchmarks to achieve human parity, but still fail on adversarial examples and make errors that demonstrate a lack of common sense. In this work, we propose gamification as a framework for data construction. The goal of players in the game is to compose questions that mislead a rival AI, while using specific phrases for extra points. The game environment leads to enhanced user engagement and simultaneously gives the game designer control over the collected data, allowing us to collect high-quality data at scale. Using our method we create CommonsenseQA 2.0, which includes 14,343 yes/no questions, and demonstrate its difficulty for models that are orders-of-magnitude larger than the AI used in the game itself.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification

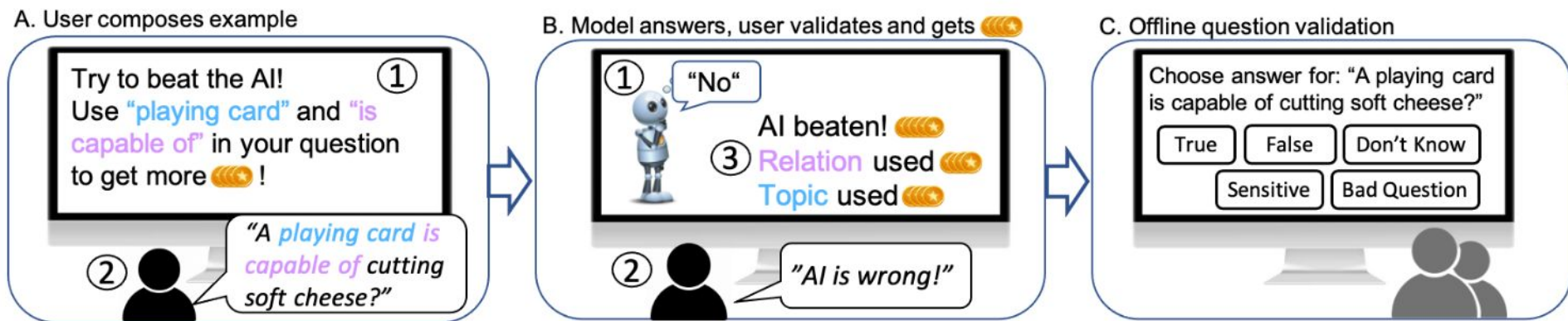


Figure 1: An overview of our approach for data collection through gamification.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification



Figure 6: Distribution of the relational prompt words in questions. Each image displays a topic prompt, the area of each image is proportional to the frequency of the corresponding relational prompt in the dataset.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification

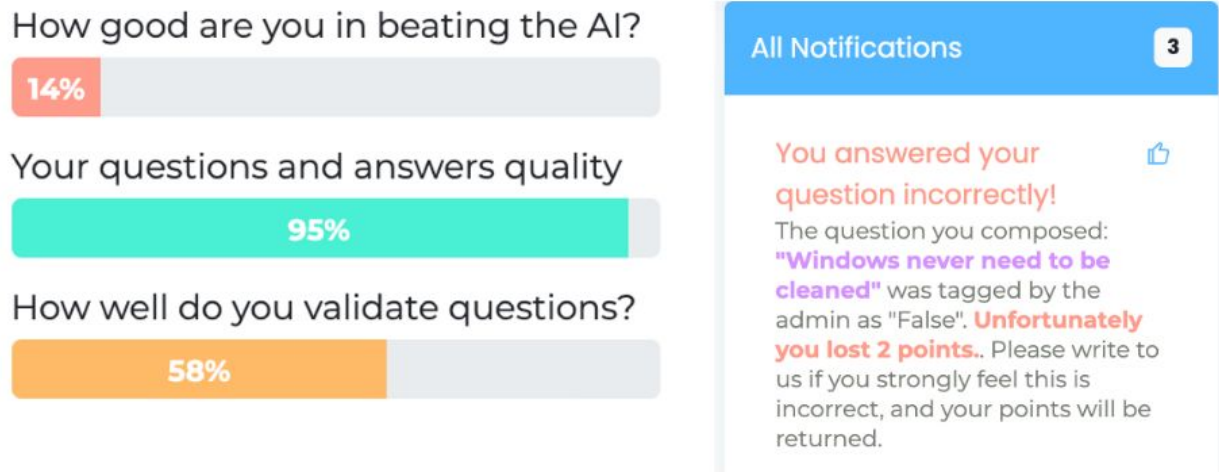


Figure 3: Feedback given to users during the game. Left: metrics on average daily player performance. Right: notification on a bad question that leads to point deduction.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification

The fun factor At the end of each playing session, players were encouraged to leave feedback. We selected sentiment words out of the 100 most frequently used words in the comments, shown in Fig. 5. We find that users enjoy the game and mostly use positive sentiment words. Fun is an important factor in encouraging high engagement, allowing us to select annotators that are better at beating the AI, while maintaining a low average cost.

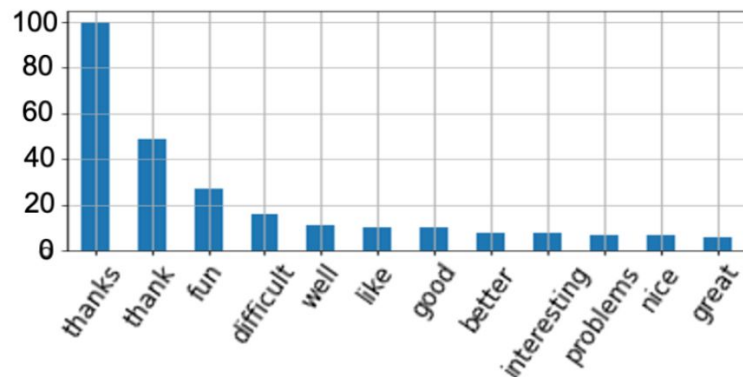


Figure 5: Sentiment words in comments.

CommonsenseQA 2.0: Exposing the Limits of AI through Gamification

	<i>Dev</i>	<i>Test</i>
GPT-3	58.4	52.9
T5-LARGE	53.8	54.6
UNICORN-LARGE	56.4	54.9
T5-11B	68.5	67.8
UNICORN-11B	69.9	70.2
Human	94.1	-

Table 4: Development and test accuracies (%) on CSQA2.

Programming Puzzles (8 avg reviewers score)

Abstract

We introduce a new type of programming challenge called programming *puzzles*, as an objective and comprehensive evaluation of program synthesis, and release an open-source dataset of Python Programming Puzzles (P3). Each puzzle is defined by a short Python program f , and the goal is to find an input x which makes f output True. The puzzles are objective in that each one is specified entirely by the source code of its verifier f , so evaluating $f(x)$ is all that is needed to test a candidate solution x . They do not require an answer key or input/output examples, nor do they depend on natural language understanding. The dataset is comprehensive in that it spans problems of a range of difficulties and domains, ranging from trivial string manipulation problems that are immediately obvious to human programmers (but not necessarily to AI), to classic programming puzzles (e.g., Towers of Hanoi), to interview/competitive-programming problems (e.g., dynamic programming), to longstanding open problems in algorithms and mathematics (e.g., factoring). The objective nature of P3 readily supports self-supervised bootstrapping. We develop baseline enumerative program synthesis and GPT-3 solvers that are capable of solving easy puzzles—even without access to any reference solutions—by learning from their own past solutions. Based on a small user study, we find puzzle difficulty to correlate between human programmers and the baseline AI solvers.

Programming Puzzles

To illustrate, `(lambda s: "Hello " + s == "Hello world")2` is an example of a Python programming puzzle, with the answer `"world"`. Some puzzles have multiple answers and some puzzles,

```
def f1(s: str): #find a string with 1000 o's but no consecutive o's.
    return s.count("o") == 1000 and s.count("oo") == 0

def f2(x: List[int]): #find the *indices* of the longest monotonic subsequence
    s = "Dynamic programming solves this classic job-interview puzzle!!!"
    return all(s[x[i]] <= s[x[i+1]] and x[i] < x[i+1] for i in range(25))

def f3(d: int): #find a non-trivial integer factor
    n = 100433627766186892221372630609062766858404681029709092356097
    return 1 < d < n and n % d == 0
```

Figure 1: Programming puzzles ranging from trivial to longstanding open algorithmic challenges in multiple domains. `f1("ox" * 1000) == True` where `s * n` means `n` repetitions of string `s`; dynamic programming efficiently finds a list of 26 increasing indexes to solve `f2` while brute force would take exponentially long; and `f3` requires advanced computational number theory algorithms.

Programming Puzzles

Table 2: Solved problems per domain with up to 1M tries per puzzle for enumerative and 10K for GPT-3. The first row also shows the number of available P3 problems in that domain. Bootstrapping models, that learn from new solutions as they are found, are in grayed lines. The score in the last column is the macro-average of the success rates across the different domains. We report the results of the run with the highest score across our three trials per enumerative model.

Model	Algebra	Basic	Chess	Classic	CodeForces	Compression	Conway's	Game Theory	Games
Uniform	0/4	7/21	0/5	1/22	7/24	0/3	0/2	0/2	0/5
Random forest	0	10	0	4	8	0	0	0	0
B. Random forest	0	13	0	5	7	0	0	0	0
Transformer	0	10	0	4	7	0	0	0	0
B. Transformer	0	13	0	3	8	0	0	0	0
GPT-3 (Medium)	0	6	0	1	7	0	0	0	1
B. GPT-3	0	13	0	2	7	0	0	0	1

Model	Graphs	ICPC	IMO	Lattices	N. Theory	Probability	Study	Trivial inverse	Tutorial	Score
Uniform	1/11	0/3	0/6	0/2	3/16	1/5	9/30	22/34	2/5	13.9
Random forest	2	0	0	0	4	1	7	25	3	17.7
B. Random forest	2	0	0	0	6	1	10	26	3	20.0
Transformer	2	0	0	0	6	1	12	24	3	19.0
B. Transformer	4	0	0	0	7	1	13	26	3	21.6
GPT-3 (Medium)	4	0	0	0	1	1	17	18	5	19.7
B. GPT-3	6	0	0	0	5	1	18	30	4	25.2

Programming Puzzles

The first finding is that success in puzzles correlates with programming experience. For our retrospective study analysis, we split the participants by the median years of Python programming experience. We had 10 *beginners* with less than three years of experience, and 11 *experienced* participants with at least three years. We find that 9 of the 30 puzzles were solved by all beginners, while 17 of the puzzles were solved by all experienced participants. Also, beginners spent on average 194 seconds per puzzle, while experienced spent only 149 seconds on average. The average solving time provides a useful proxy to the perceived difficulty of each puzzle. Overall, we see that puzzles are easier for experienced programmers, indicating their value for evaluating programming proficiency.

The second finding is that experienced human programmers outperformed our AI baselines. There were over 10 puzzles that none of our baselines solved, while the average number of puzzles solved by experienced programmers was greater than 25, and one participant solved all 30. However, bear in mind that these results are with a 6 minute limit. No human would solve the puzzles within seconds, and we would expect experienced programmers to eventually solve virtually all study puzzles.

The Benchmark Lottery (5.75 avg reviewers score - not accepted)

Abstract

The world of empirical machine learning (ML) strongly relies on benchmarks in order to determine the relative effectiveness of different algorithms and methods. This paper proposes the notion of *a benchmark lottery* that describes the overall fragility of the ML benchmarking process. The benchmark lottery postulates that many factors, other than fundamental algorithmic superiority, may lead to a method being perceived as superior. On multiple benchmark setups that are prevalent in the ML community, we show that the relative performance of algorithms may be altered significantly simply by choosing different benchmark tasks, highlighting the fragility of the current paradigms and potential fallacious interpretation derived from benchmarking ML methods. Given that every benchmark makes a statement about what it perceives to be important, we argue that this might lead to biased progress in the community. We discuss the implications of the observed phenomena and provide recommendations on mitigating them using multiple machine learning domains and communities as use cases, including natural language processing, computer vision, information retrieval, recommender systems, and reinforcement learning.

The Benchmark Lottery

- Section 2 discusses how benchmarks can influence long-term research directions in a given (sub-)field, and describes the life cycle of a benchmark.
- Section 3 introduces the *task selection bias* and using established benchmarks as examples shows how relative performance of algorithms is affected by the task selection process.
- Section 4 takes another view of the task selection bias and proposes *community bias* as a higher-level process that influences task selection. We show that forces from the broader research community directly impact the task selection process and as a result, play a substantial role in creating the lottery.
- Section 5 posits that benchmarks are stateful entities and that participation in a benchmark differs vastly depending upon its state. We also argue continual re-use of the same benchmark may be problematic.
- Section 6 discusses *rigging the lottery*, the issue that some communities (e.g. recommender systems and reinforcement learning) face, where the lack of well-established community-driven sets of benchmarks or clear guidelines may inadvertently enable researchers to fit benchmarks to model. We highlight the potential drawbacks of such an approach.
- Finally, in Section 7 we provide recommendations for finding a way out of the lottery by building better benchmarks and rendering more accurate judgments when comparing models.

The Benchmark Lottery

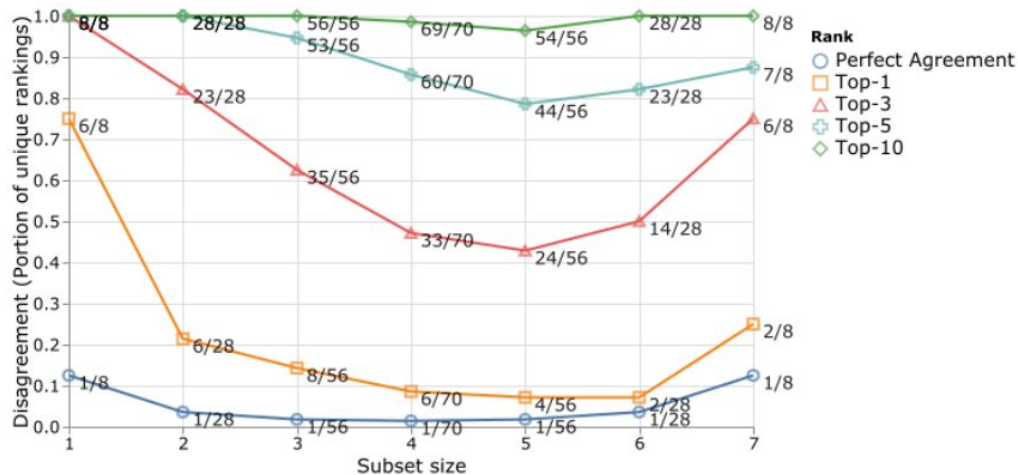


Figure 1: Disagreement of model rankings on the SuperGLUE benchmark as a function of the number of selected benchmark tasks. The x -axis represents the number of tasks in each sub-selection of tasks

Ranking inconsistency. Figure 1 gives a concise overview of the number of unique Top- k rankings produced obtained from fixed-size subsets of tasks. For example among the 70 different possibilities of selecting 4 out of 8 tasks, there are 6 distinct model ranking orders produced for Top-1 (i.e. there are 6 different possible top models). Moreover, when considering Top-3 or even Top-5, almost 60 out of 70 rankings do not agree with each other. Overall, the rankings become highly diverse as the subset of tasks selected from the benchmark is varied. This forms the core of the empirical evidence

The Benchmark Lottery

A SuperGLUE: Ranking of models on different combinations of tasks

Figure 4 shows the performance of different models on different combinations of tasks in terms of their rank in the list of all models. The very top row in the heatmap is the ranking on the SuperGLUE (considering all tasks) and models on the x-axis are sorted based on their rank on "All" tasks. There is no strong pattern observable in the plot. For instance, the top ranked model on "All" does not newsreel perform best on other combinations of tasks.

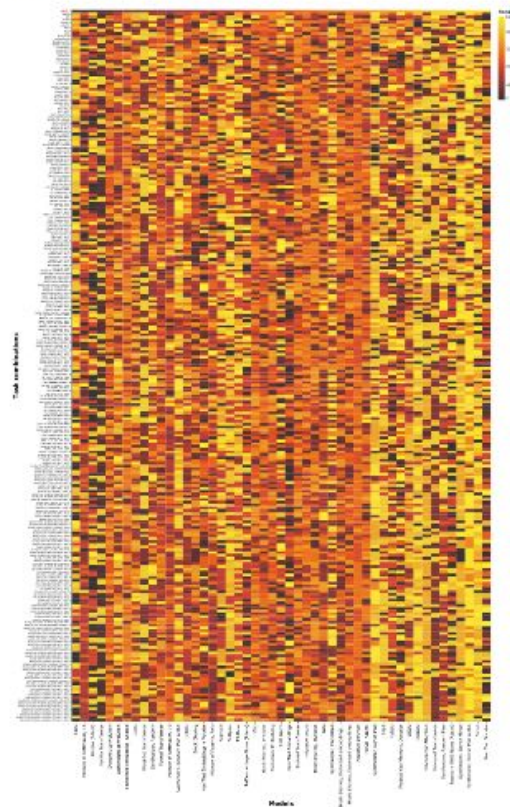


Figure 4: Performance of different models on different combinations of tasks in in terms of their rank on the SuperGLUE benchmark. Models are sorted on the x -axis based on their rank when evaluated on "ALL" tasks. We can observe that there no clear and strong pattern or correlation in different combinations of tasks compared to the full benchmark, indicating that there is no "best" model, while most of the time, the top-ranked model is simply taken as the absolute winner.

Thank you!