

The Data Layer of AI

Twenty-six articles on the data behind modern artificial intelligence, one per newsletter issue

The articles

Part I	The Data Problem	3
01	The Bottleneck Moved	3
02	The Logarithm Problem	8
03	The Test Set Was Wrong	13
04	Failures That Arrive Late	18
05	“High Quality” Is Not a Property of Data	24
Part II	The Science of Annotation	30
06	Disagreement Is Data	30
07	Agreement Is Not Truth	36
08	The Arithmetic of Trust	41
09	The Expert Premium	46
Part III	After Simple Labeling	52
10	Annotation Is Moving Up the Stack	52
11	Stop Labeling Everything. Carefully.	58
12	What the Human in the Loop Is For	64
Part IV	The Generative AI Data Layer	70
13	RLHF Is a Data Problem	70
14	Whose Preferences?	75
15	When AI Grades AI	81
16	Evaluation Has Become Data Generation	87
17	Red Teaming Is Dataset Construction	93
Part V	Synthetic Data	99
18	The Synthetic Data Paradox	99
19	Model Collapse, Read Carefully	105
Part VI	Seeing and Moving	111
20	Time Is the Hardest Label	111
21	Machines That Move	117
Part VII	The Business of Data	123
22	Cost per Label Is the Wrong Unit	123
23	Build, Buy or Partner	129
24	Where Did This Data Come From?	136
Part VIII	India	142
25	Not a Translation Problem	142
26	The Voice Gap	148

The Bottleneck Moved

Compute and architecture still matter. But the scarce input in applied AI is increasingly the data that tells a model what good looks like.

7 MINUTE READ

In 2022, a team at OpenAI reported a result that should have unsettled anyone who believed model size was the main story. Human evaluators were shown outputs from two language models and asked which they preferred. One model had 175 billion parameters. The other had 1.3 billion, less than one hundredth the size. The evaluators preferred the small one.

The difference between the two was not architecture, and it was not compute. The small model had been fine-tuned on a few tens of thousands of examples of human demonstration and human preference. The technique was reinforcement learning from human feedback, and the paper, known as InstructGPT, became the template for how raw language models are turned into usable products.

Read as a data story, the result says something specific. A hundredfold deficit in parameters was more than offset by roughly 77,000 carefully collected prompts, written and ranked by about 40 contractors. That is a small dataset by any modern standard. It is also a dataset that could not be scraped. Somebody had to design the task, recruit and screen the people, write the instructions, and decide what to do when they disagreed.

This is what it means to say the bottleneck moved.

Three eras, three scarce inputs

It helps to be precise about what was scarce, and when.

In the supervised era that followed ImageNet, the scarce input was labelled examples. Progress in vision, speech and translation tracked the availability of large annotated datasets, and annotation was a volume business: millions of images, each needing a class or a box.

Self-supervised pretraining changed that. Models such as BERT and the GPT family learn by predicting missing or next tokens in raw text, which needs no labels at all. For several years the scarce inputs were compute and raw text, and the scaling-law literature formalised the trade. Kaplan and colleagues showed in 2020 that

language-model loss falls predictably as parameters, data and compute rise. Hoffmann and colleagues refined this in 2022, arguing that the large models of the day were undertrained and that, for a fixed compute budget, model size and training tokens should grow in roughly equal proportion.

The third era began when pretrained models became good enough to be useful and not yet good enough to be trusted. A raw language model completes text. It does not reliably follow instructions, decline harmful requests or know when to stop. Closing that gap is called post-training, and it runs on human-made data of a new kind: demonstrations of good behaviour, comparisons between outputs, and test cases that define success. The volume is small. In the InstructGPT case it was tens of thousands of items, set against the hundreds of billions of tokens used in pretraining.

The Hoffmann result deserves a second look. It is usually cited as a statement about compute. It is equally a statement about data: a bigger model is only worth building if there is proportionally more text to train it on.

FIGURE 1 · FRAMEWORK

Three eras, three scarce inputs

ERA	HOW MODELS LEARN	SCARCE INPUT	HUMAN-MADE DATA
Supervised c. 2012–2018	From labelled examples, one task at a time	Labels, in volume	Millions of simple labels: classes, boxes, transcripts
Self-supervised pretraining c. 2018–2022	By predicting missing or next tokens in raw data	Compute and raw text	Little direct labelling; filtering and curation of crawls
Post-training and evaluation 2022 onward	A pretrained model is adapted with feedback and tested for each use	Judgement: demonstrations, preferences, expert review, test cases	Thousands of difficult decisions, each with more influence on behaviour

Synthesis by the editors. Era boundaries are approximate and overlap.

The supply side is tightening

Two lines of evidence suggest that the supply of raw text is not unlimited.

The first is arithmetic. Villalobos and colleagues at Epoch AI estimated the total stock of public human-written text and compared it with the growth rate of training datasets. Their conclusion, published at ICML in 2024, is that if current trends continue, models will be trained on datasets roughly equal in size to the available stock of public human text at some point between 2026 and 2032. The authors are careful to say this is not a hard wall; synthetic data, transfer from other modalities and better data efficiency can all extend the runway. But each of those is, itself, a data-engineering problem.

The second is consent. The Data Provenance Initiative audited 14,000 web domains underlying three widely used training corpora. Its 2024 study found a rapid rise in restrictions aimed at AI crawlers: more than 5% of all tokens in the C4 corpus, and more than 28% of tokens from its most actively maintained, critical sources, were fully restricted by robots.txt, and 45% of C4 was restricted by terms of service. The open web that pretraining depended on is closing, unevenly and fast.

Neither finding means AI progress stops. Both mean that the marginal unit of useful data is getting more expensive to obtain, and more valuable once obtained.

Where practitioners say projects break

The research literature on why applied AI fails points the same way, from a different direction.

Sambasivan and colleagues at Google interviewed 53 AI practitioners working on high-stakes applications in India, East and West Africa, and the United States. Ninety-two per cent reported at least one “data cascade”, a compounding failure that originated in data and surfaced downstream. Nearly half reported two or more. The paper’s title, taken from an interviewee, is its own summary: everyone wants to do the model work, not the data work.

A 2024 RAND study, based on interviews with 65 experienced data scientists and engineers, reached a compatible conclusion. Among the root causes of failed AI projects it identified were organisations lacking the data needed to train an effective model, and teams misunderstanding what problem the data was supposed to help solve.

Analyst forecasts go further. Gartner predicted in early 2025 that, through 2026, organisations will abandon 60% of AI projects that are unsupported by what it calls AI-ready data. That is a forecast rather than a measurement, and should be read as one. But it reflects what the firm’s clients report.

KEY TAKEAWAY

The claim is not that data matters more than models. It is that for most organisations the model is now something they can rent, while the data that adapts, evaluates and corrects it is something they must make.

The strongest objection

The best counterargument comes from Rich Sutton's 2019 essay "The Bitter Lesson". Its thesis is that, across seventy years of AI research, methods that exploit general-purpose computation have repeatedly beaten methods that encode human knowledge. On this view, careful data work is another form of hand-crafting, destined to be overtaken by scale.

The objection has force, and the history supports it. Hand-built features lost to learned ones. Curated linguistic resources lost to raw text.

But notice what the bitter lesson is actually about. It is a claim about how knowledge should enter a system: through learning, not through hand-coded rules. It says nothing against the importance of what the system learns from. InstructGPT is fully consistent with it. Nobody wrote rules for helpfulness. People produced examples and comparisons, and a general method learned from them.

A second objection is that compute remains a real constraint, and it does. Frontier training runs are limited by chips and power. The claim here is narrower. For the much larger population of organisations that are not training frontier models, compute is available by the hour. A dataset that captures how their domain works is not.

What the shift looks like in practice

The volume of human-made data per model has fallen. Its unit value has risen, and its character has changed.

The supervised era asked annotators questions with answers: is this a cat, where is the pedestrian. The post-training era asks questions that require judgement: which of these two answers is more helpful, is this medical summary safe to act on, did the agent complete the task or only appear to. The people who can answer are fewer. The instructions are harder to write. The disagreements are more informative.

Evaluation has followed the same path. A model that can do many things needs many tests, and each test is a dataset someone has to build, protect from contamination and keep current.

This is why large AI developers have been spending heavily on the organisations that produce such data. In June 2025, Meta agreed to pay about 14.3 billion dollars for a 49% stake in Scale AI, a data-labelling company, according to Reuters and other outlets. Whatever one thinks of the price, nobody pays it for a commodity.

What this changes

FOR AI BUILDERS

Treat post-training and evaluation data as designed artefacts with owners, versions and specifications, in the same way as code.

Before scaling a model, ask whether the limiting factor is capacity or signal. A small amount of well-specified feedback can outweigh a large increase in parameters.

Assume that freely scrapeable data will keep shrinking. Plan for licensed, commissioned and first-party sources.

FOR ENTERPRISES

The foundation model is increasingly a purchased input. Differentiation comes from data that encodes your domain, your edge cases and your standards.

Budget for data work as a continuing operating function, not a one-off project cost that ends at launch.

Ask of every AI initiative: what data would tell us this system is wrong, and who is producing it?

THE INDIKA PERSPECTIVE

The useful question for an enterprise is no longer “who will label our data” but “what is the smallest amount of expert-made data that changes model behaviour, and how do we produce it reliably”. That reframes the work from throughput to design: task specification, annotator selection, disagreement handling and evaluation become the product. The rest of this series takes those pieces one at a time.

RESEARCH & REFERENCES

Ouyang, L. et al. (2022). “Training language models to follow instructions with human feedback.” NeurIPS.

Kaplan, J. et al. (2020). “Scaling Laws for Neural Language Models.” arXiv preprint.

Hoffmann, J. et al. (2022). “Training Compute-Optimal Large Language Models.” NeurIPS.

Villalobos, P. et al. (2024). “Position: Will we run out of data? Limits of LLM scaling based on human-generated data.” ICML.

Longpre, S. et al. (2024). “Consent in Crisis: The Rapid Decline of the AI Data Commons.” NeurIPS Datasets and Benchmarks.

Sambasivan, N. et al. (2021). “Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI.” ACM CHI.

Ryseff, J., De Bruhl, B. and Newberry, S. (2024). “The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed.” RAND Corporation.

Gartner (2025). “Lack of AI-Ready Data Puts AI Projects at Risk.” Press release, 26 February. Forecast.

Sutton, R. (2019). “The Bitter Lesson.” Essay.

Devlin, J. et al. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” NAACL.

Reuters (2025). Reporting on Meta’s investment in Scale AI, June.

The Logarithm Problem

Why the tenth million examples teach a model far less than the first, and what that does to the case for “just collect more”.

6 MINUTE READ

In 2017, researchers at Google and Carnegie Mellon ran an experiment designed to answer a simple question. If you give a vision model 300 times more training data than ImageNet, how much better does it get?

They had the means to find out: an internal dataset called JFT-300M, with around 300 million images and noisy, automatically generated labels. The answer was that performance kept improving, which was the headline. The detail was in the shape of the curve. Performance on downstream vision tasks rose **logarithmically** with the volume of training data.

A logarithmic relationship is easy to state and easy to underestimate. It means each fixed gain in performance requires multiplying the data, not adding to it. Going from one million to ten million examples buys roughly what going from ten million to a hundred million buys. The second step costs ten times as much.

Both sides of the “more data versus better data” argument cite this paper. They are both right, and the reason is worth working through.

The case for volume

The volume argument has a distinguished history. In 2001, Banko and Brill showed that on a language task, simple algorithms trained on a billion words outperformed sophisticated ones trained on a million, and that the ranking of algorithms changed as data grew. In 2009, three Google researchers gave the idea its name in an essay called “The Unreasonable Effectiveness of Data”, arguing that for problems involving human behaviour, a lot of messy data beats a little clean data and an elegant model.

More recent evidence is stronger still. CLIP was trained on 400 million image–text pairs collected from the web with no manual labelling. ALIGN used 1.8 billion pairs of images and noisy alt-text, and its authors stated their conclusion plainly: scale can make up for noise. Rolnick and colleagues showed that deep networks can learn accurately even when correct labels are outnumbered many times over by randomly wrong ones, provided there are enough correct examples in absolute terms.

So the volume position is not naive. Under the right conditions, quantity does substitute for quality.

The conditions

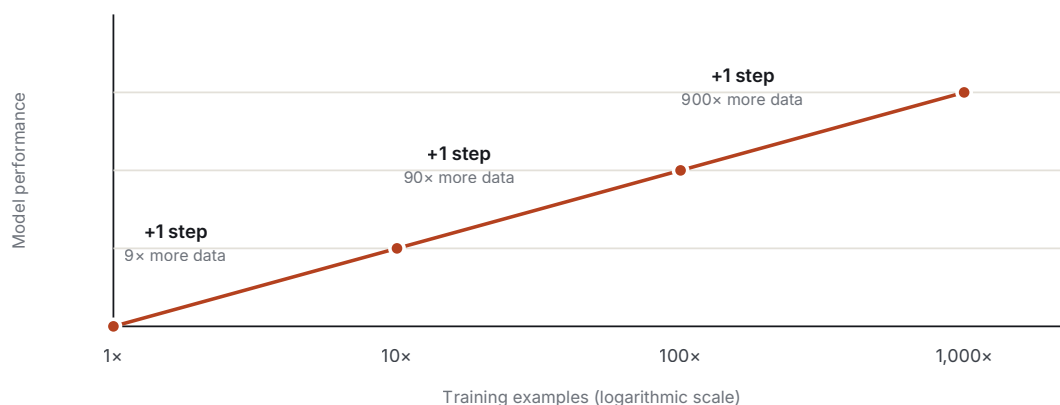
There are two conditions, and they are the whole argument.

The first is that the noise is random. Rolnick's result concerns labels corrupted uniformly at random. Random errors point in all directions and cancel as data grows. Systematic errors do not. If annotators consistently confuse two classes, or if a web-scraped caption source consistently describes one group of people in a particular way, more data reinforces the error. Work on real annotation noise bears this out: when Wei and colleagues re-labelled CIFAR with human annotators, they found that real-world label noise is structured by the content of the image and is harder for models to cope with than the synthetic noise most robustness methods are tested against.

The second is that data is nearly free. A logarithmic return is acceptable when the marginal example costs almost nothing, as with crawled web pages. It is ruinous when each example needs a radiologist, a native speaker of Maithili, or a robot arm in a real kitchen.

FIGURE 2 · CONCEPTUAL ILLUSTRATION

Equal gains, tenfold costs



Illustrative. The log-linear shape follows Sun et al. (2017) and Udandara et al. (2024); the axis values are schematic, not measurements.

The long tail makes this sharper. A 2024 study by Udandara et al. examined 34 multimodal models across five pretraining datasets and asked how well each model handled a concept as a function of how often that concept appeared in its training data. The relationship was log-linear: linear gains in performance required exponentially more examples of the concept. The models

were not generalising to rare concepts so much as performing in proportion to exposure. For anything rare, which in most enterprise settings means most of what matters, undirected volume is the slowest possible route.

What selection buys

If volume gives logarithmic returns, the natural question is whether choosing examples carefully can do better. The research answer is a qualified yes.

Toneva and colleagues tracked how often each training example was learned and then forgotten during training. Many examples were never forgotten once learned, and a large share of those could be removed from the training set without hurting accuracy. Not all data is doing work.

Swayamdipta and colleagues took this further with what they called dataset cartography. By measuring a model's confidence in each example across training, and how much that confidence varied, they sorted datasets into three regions. Easy examples helped optimisation but added little. Ambiguous examples, where the model wavered, contributed most to generalisation outside the training distribution. Hard examples, which the model rarely got right, were frequently labelling errors.

That taxonomy is practically useful. It says the examples most worth paying for are those the current model is uncertain about, and that the examples it consistently fails on deserve an audit before they deserve more training.

Sorscher and colleagues supplied the theory. Their 2022 analysis showed that with a good enough measure of example difficulty, error can in principle fall exponentially with the size of a pruned dataset, not as a power law. They were also candid about the gap between theory and practice: most existing pruning metrics performed poorly at ImageNet scale.

The same idea shows up in small language models. Microsoft's phi-1, a 1.3-billion-parameter model, reached competitive results on Python coding benchmarks after training on a few billion tokens of filtered and synthetic "textbook quality" data, a fraction of what comparable models consumed. The authors' explanation was the data.

KEY TAKEAWAY

"More versus better" is a false contest. Volume wins when examples are free and errors are random. Selection wins when examples are costly, errors are systematic, or the cases that matter are rare. Most enterprise AI sits in the second regime.

Where the selection argument overreaches

Curation is not costless, and its advocates sometimes write as if it were.

Every selection method needs a model to do the selecting, and the choice of examples then carries that model's blind spots. Data chosen as informative for one architecture is not always informative for the next. Pruning removes redundancy, but it can also remove the rare subgroups that happened to look redundant to the scoring function. And a curated dataset is a smaller dataset, which is a weaker defence against distribution shift nobody anticipated.

There is also an honest asymmetry between frontier labs and everyone else. For a lab pretraining a foundation model on trillions of tokens, heavy filtering has been important but so has sheer scale, and nobody has shown that a small curated corpus can replace broad pretraining. The selection argument is strongest downstream: in fine-tuning, in evaluation and in domain adaptation, where datasets are small enough for each example to be examined and expensive enough to justify it.

A way to decide

A useful test is to ask what the next thousand examples would cost and what they would teach.

If they are nearly free and drawn from the same distribution the model will face, collect them. If they are expensive, ask first which thousand. A model's own uncertainty, its error analysis by subgroup, and its failures in production are all better guides to what to collect next than a target number.

A second test is to ask whom the model fails. Aggregate accuracy can keep rising while performance for a small but important group stays flat, because new data contains that group in the same small proportion as before. If the failures concentrate in a particular region, device, language or customer type, the remedy is data from that group, and no increase in general volume will supply it at a reasonable cost.

A third test is to ask what kind of errors the data contains. Random noise can be outvoted with redundancy or scale. Systematic noise needs a change in guidelines, annotators or sources, and no quantity of additional data will substitute for that.

What this changes

FOR AI BUILDERS

Plot a learning curve before commissioning more data. If accuracy is flat on a log axis, volume is no longer the constraint.

Use model confidence and training dynamics to split data into easy, ambiguous and consistently wrong. Buy more of the second and audit the third.

Measure rare-case performance separately. Aggregate accuracy hides the tail, and the tail is where the logarithm bites hardest.

FOR ENTERPRISES

Question any data plan stated only as a volume target. Ask what the examples are for and how they were chosen.

Expect a data partner to reduce the number of labels you pay for, not to maximise it.

Where your important cases are rare, budget to collect them deliberately. They will not arrive in proportion to their importance.

THE INDIKA PERSPECTIVE

A data operation that understands the logarithm sells differently. Its value is in knowing which examples not to produce: finding the ambiguous region of a dataset, auditing the consistently wrong, and directing expensive expert attention to the rare cases a model has barely seen. Throughput still matters, but as a constraint to be managed, not as the measure of the work.

RESEARCH & REFERENCES

Sun, C. et al. (2017). "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era." IEEE ICCV.

Banko, M. and Brill, E. (2001). "Scaling to Very Very Large Corpora for Natural Language Disambiguation." ACL.

Halevy, A., Norvig, P. and Pereira, F. (2009). "The Unreasonable Effectiveness of Data." IEEE Intelligent Systems.

Radford, A. et al. (2021). "Learning Transferable Visual Models From Natural Language Supervision." ICML.

Jia, C. et al. (2021). "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision." ICML.

Rolnick, D. et al. (2017). "Deep Learning is Robust to Massive Label Noise." arXiv preprint.

Wei, J. et al. (2022). "Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations." ICLR.

Udandara, V. et al. (2024). "No 'Zero-Shot' Without Exponential Data: Pretraining Concept Frequency Determines Multimodal Model Performance." NeurIPS.

Toneva, M. et al. (2019). "An Empirical Study of Example Forgetting during Deep Neural Network Learning." ICLR.

Swayamdipta, S. et al. (2020). "Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics." EMNLP.

Sorscher, B. et al. (2022). "Beyond neural scaling laws: beating power law scaling via data pruning." NeurIPS.

Gunasekar, S. et al. (2023). "Textbooks Are All You Need." Microsoft Research, arXiv preprint.

The Test Set Was Wrong

The benchmarks used to rank a decade of models contain thousands of mislabelled examples. Correcting them changes which model wins.

7 MINUTE READ

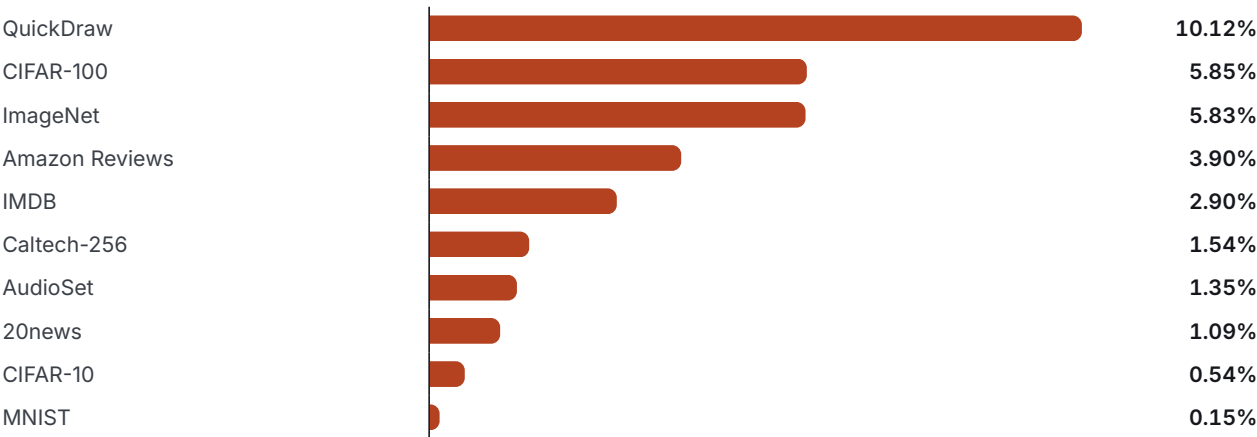
For most of the past decade, the ImageNet validation set has been the yardstick of computer vision. Fifty thousand images, each with a single correct label, against which thousands of models have been scored and ranked. Papers were accepted, architectures adopted and careers advanced on differences of a fraction of a percentage point.

In 2021, three researchers at MIT went looking for mistakes in it. They used an algorithm to flag images whose labels looked statistically suspicious, then had human reviewers check the candidates. They estimated that about 6% of the ImageNet validation set is mislabelled: 2,916 images.

ImageNet was not unusual. Across ten of the most widely used benchmarks in vision, text and audio, the same team estimated an average test-set error rate of at least 3.3%. Google’s QuickDraw came in above 10%.

FIGURE 3 · DATA VISUALISATION

Estimated label errors in ten benchmark test sets



Share of test-set labels estimated to be erroneous. Source: Northcutt, Athalye and Mueller (2021), Table 1. Figures are the authors’ estimates after human validation of algorithmically flagged candidates.

The finding that matters most is not the error rate. It is what happened when the errors were corrected.

When the ranking flips

A benchmark exists to order models. If model A scores higher than model B, we conclude that A is better and act accordingly. That inference assumes the answer key is right.

Northcutt, Athalye and Mueller tested the assumption directly. They re-evaluated a range of standard models on the examples whose labels had been corrected. On those examples, the ordering of models changed. Larger, higher-capacity models, which scored best against the original labels, did worse than smaller models once the labels were fixed.

Their explanation was that large models had enough capacity to fit the systematic patterns in the labelling mistakes, and were rewarded for it by a test set that contained the same mistakes. The authors quantified how fragile the standard ranking is. On ImageNet with corrected labels, ResNet-18 outperforms ResNet-50 if the prevalence of originally mislabelled test examples increases by just 6%.

A practitioner choosing between two models on a noisy benchmark may be selecting the one that has better learned the benchmark's errors. In a deployment where real data is noisier than the curated benchmark, which is most deployments, the model that looked worse may be the better choice.

Not all errors are errors

Before treating 6% as a simple defect rate, it is worth looking at what the mistakes are.

A parallel line of work examined ImageNet's labels in detail. Beyer and colleagues, in a paper titled "Are we done with ImageNet?", found that a large share of the apparent errors arose because images contain several objects while the benchmark allows one label. A photograph of a desk with a laptop, a keyboard and a mug has a single ground-truth class, and a model that names a different visible object is marked wrong. Tsipras and colleagues reached a similar conclusion: much of the remaining gap between top models and perfect accuracy reflected the labelling procedure, not the models' perception.

A later study from Google Research reviewed every remaining mistake made by the best models of the time. Many were arguably not mistakes at all, and a meaningful fraction involved fine-grained distinctions, such as closely related dog breeds, where the original label itself was in doubt.

So "label error" covers at least three different things:

- **Plain mistakes.** The label is wrong and any informed reviewer would agree.

- **Underspecified tasks.** The image is multi-object, the text is ambiguous, and the single-label format cannot represent it.
- **Expertise gaps.** The correct label exists but the original annotators could not reliably determine it.

Each calls for a different remedy. The first needs correction. The second needs a different task design. The third needs different annotators.

Why test errors cost more than training errors

There is a reasonable intuition that a few per cent of noise is tolerable. For training data, that intuition has research behind it: deep networks are often robust to moderate amounts of random label noise, and methods such as confident learning can identify and down-weight suspect examples.

Test data is different, for three reasons.

It is small. A training set has millions of examples to average over. A typical evaluation set has thousands, and the subsets that matter, such as rare classes or critical failure modes, may have dozens. A handful of wrong labels in a subset of fifty changes the measured result materially.

It is the instrument. Training noise makes a model somewhat worse. Test noise makes every measurement of every model wrong, by an unknown amount, in an unknown direction. Every decision that depends on the metric inherits the error: which model to ship, whether a fine-tune helped, whether last quarter's regression was real.

It sets a false ceiling. If 5% of labels are wrong, a perfect model scores 95%. Teams then spend effort chasing the last few points of headroom that do not exist, or worse, reward the model that reproduces the labellers' mistakes.

Medical AI has learned this the expensive way. In work on diabetic retinopathy, Krause and colleagues at Google showed that the choice of reference standard substantially changes a model's measured performance. When labels came from a majority vote of individual graders, some disease was missed that became evident when retinal specialists discussed the same images and adjudicated. Evaluated against the adjudicated standard, error rates looked different, and tuning the model on a small set of adjudicated labels improved it. The model had not changed. The ruler had.

KEY TAKEAWAY

A model can be no better than you are able to measure. An evaluation set deserves a higher labelling standard than the training set it judges, and usually receives a lower one.

The case for not worrying

There are fair objections to making too much of this.

The first is that the error estimates are themselves produced by a noisy process. Candidate errors were flagged by an algorithm and confirmed by crowd workers, who can be wrong too. The authors acknowledge this and describe their figures as estimates. The exact percentage for any dataset should be held loosely.

The second is that benchmark rankings have been broadly reliable in practice. When Recht and colleagues built a fresh ImageNet test set following the original protocol, accuracy fell by 11 to 14 percentage points across models, but the ordering of models was largely preserved. Models that did better on the old test did better on the new one. Benchmarks contain errors and still carry signal.

Both objections are sound, and both leave the main point standing. Rankings are stable when differences between models are large. They become unreliable when differences are small, and small differences are what most real model-selection decisions turn on. A team choosing between two fine-tunes separated by half a point on an internal test set of two thousand examples is deciding inside the noise.

What good practice looks like

None of this requires exotic tooling. It requires treating the evaluation set as the most important dataset an organisation owns.

Audit it. Run a label-error detection pass over the test set, and manually review the examples where strong models disagree with the label. Disagreement between a good model and the answer key is the cheapest available signal of a labelling problem.

Label it differently. Use more annotators per item, more senior annotators, and adjudication for disagreements. The economics favour this: evaluation sets are small, so a high per-item cost is a low total cost.

Record the ambiguity. Where an item has more than one defensible answer, store that fact. A test set that forces a single label onto an ambiguous item is measuring agreement with one annotator's coin flip.

Protect it. An evaluation set that has been used to tune a model, or has leaked into training data, no longer measures what it did. Keep it separate, control who can see it and replace part of it periodically. A later article in this series takes up contamination in detail.

Report uncertainty. A metric computed on a few thousand noisy labels has an error bar. Decisions should be made with it visible.

What this changes

FOR AI BUILDERS

Before trusting a small improvement, estimate the label error rate of the evaluation set. If the improvement is smaller than the error rate, it is not yet a result.

Review every test item where your best model confidently disagrees with the label. Expect to find both model errors and label errors.

Keep a small adjudicated “platinum” subset labelled to a higher standard, and check that conclusions hold on it.

FOR ENTERPRISES

Ask vendors and internal teams how the evaluation data was labelled, by whom, and with what agreement. Accuracy claims without this are unanchored.

Fund evaluation data separately from training data, with a higher quality bar and independent ownership.

Treat an evaluation set as a controlled asset. If it leaks into training, or goes stale, the numbers it produces stop meaning anything.

THE INDIKA PERSPECTIVE

The practical lesson is that evaluation data and training data are different products with different economics. Training data rewards coverage and volume at acceptable quality. Evaluation data rewards a small number of items labelled as well as they can be, with ambiguity recorded and experts adjudicating. Organisations that commission both under one specification usually get a test set that is too noisy to trust and a training set that cost more than it needed to.

RESEARCH & REFERENCES

Northcutt, C., Athalye, A. and Mueller, J. (2021). “Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks.” NeurIPS Datasets and Benchmarks.

Northcutt, C., Jiang, L. and Chuang, I. (2021). “Confident Learning: Estimating Uncertainty in Dataset Labels.” Journal of Artificial Intelligence Research.

Beyer, L. et al. (2020). “Are we done with ImageNet?” arXiv preprint, Google Research.

Tsipras, D. et al. (2020). “From ImageNet to Image Classification: Contextualizing Progress on Benchmarks.” ICML.

Vasudevan, V. et al. (2022). “When does dough become a bagel? Analyzing the remaining mistakes on ImageNet.” NeurIPS.

Recht, B. et al. (2019). “Do ImageNet Classifiers Generalize to ImageNet?” ICML.

Krause, J. et al. (2018). “Grader Variability and the Importance of Reference Standards for Evaluating Machine Learning Models for Diabetic Retinopathy.” Ophthalmology.

Rolnick, D. et al. (2017). “Deep Learning is Robust to Massive Label Noise.” arXiv preprint.

Russakovsky, O. et al. (2015). “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision.

Failures That Arrive Late

Data problems rarely announce themselves where they start. They surface months later, somewhere else, looking like a model problem.

7 MINUTE READ

A deep-learning system for detecting diabetic eye disease had performed at the level of specialist physicians in published studies. In 2018 and 2019, researchers from Google Health followed it into eleven clinics in Thailand to see how it worked in routine care.

The model's accuracy was not the problem. The photographs were. Nurses were capturing retinal images in rooms where lighting could not always be controlled, on a schedule that left little time for retakes. The system had been built to reject images it judged too poor to grade, a sensible safeguard, and it rejected roughly a fifth of them. Patients whose images were rejected were told to travel to another facility on another day. Where internet connections were slow, uploads stalled and screening queues lengthened.

Nothing about this was visible in the model's validation metrics. The training and test images had been collected under better conditions and graded by ophthalmologists. The assumption that deployment images would resemble them was made, implicitly, at the moment the dataset was assembled, and it was tested for the first time in a clinic.

Researchers have a name for this pattern.

Cascades

In 2021, Nithya Sambasivan and colleagues at Google published a study of how data problems unfold in practice. They interviewed 53 AI practitioners building high-stakes systems, in areas such as health, agriculture, credit and public safety, across India, East and West Africa and the United States.

They defined a **data cascade** as a compounding event that causes negative downstream effects from data issues. The properties they documented are what make cascades distinctive:

- They are **opaque**. There are no clear indicators, tools or metrics to detect them as they begin.

- They are **delayed**. Effects appear well after the decision that caused them, often in a different part of the system.
- They are **compounding**. Small early shortcuts interact and grow.
- They are **common**. Ninety-two per cent of the practitioners had experienced at least one. Forty-five per cent had experienced two or more.

The consequences reported were not marginal. Teams described discarding whole datasets and starting collection again, harm to the people the system was meant to serve, and strained relationships with partners.

FIGURE 4 · PROCESS MAP

How a data cascade travels

STAGE	WHAT HAPPENS	IS THE PROBLEM VISIBLE?	WHAT A FIX COSTS HERE
Collection	Data gathered under convenient conditions; the assumption that deployment will match goes unrecorded	No	A conversation with someone who knows the deployment setting
Labelling	Labels are accurate for the data as collected	No	A revised guideline and a small pilot
Training	Model converges normally	No	A re-labelling or re-sampling pass
Validation	Test set drawn from the same source; score is good	No	A new, independent evaluation set
Deployment	Real conditions differ; performance drops or the system rejects inputs	Yes, as a “model problem”	Diagnosis, then re-collection, re-labelling, retraining and revalidation
Aftermath	Trust, partners and users are affected	Yes	All of the above, plus the harm already done

Schematic based on the pattern described by Sambasivan et al. (2021). Costs are qualitative.

Four triggers

The study traced cascades to four recurring causes. Each is recognisable to anyone who has shipped a model.

The physical world is brittle. Models trained on clean data met hardware that drifted, sensors that were moved, cameras with fingerprints on the lens, and environments that changed with the seasons. The Thai clinic lighting is this trigger exactly.

Domain expertise was missing where data was made. Practitioners took decisions about what to collect, how to label it and what to discard without the subject knowledge to see the consequences. In some cases the people who did have the knowledge, such as field workers or clinicians, were treated as data collectors and not consulted on what the data meant.

Incentives pointed elsewhere. Data work was undervalued relative to model work, inside teams and in the wider research culture. The study's title is a direct quotation from an interviewee. Data collectors in the field were sometimes paid or assessed in ways that rewarded volume over care.

Documentation did not cross boundaries. Datasets passed between organisations with their assumptions unrecorded. Recipients did not know how fields had been defined, which records had been excluded, or why.

None of these is a labelling error in the narrow sense. A dataset can have perfectly accurate labels and still carry all four.

The same story in other fields

The medical-imaging literature contains several well-documented cascades, each discovered late.

Zech and colleagues trained a pneumonia detector on chest X-rays from multiple hospital systems and found it performed worse on data from a hospital it had not seen. Investigating why, they found the network could identify which hospital an image came from with high accuracy, using cues unrelated to disease. Because pneumonia prevalence differed sharply between sites, knowing the hospital was a useful shortcut. The model had learned, in part, to recognise the scanner.

During the pandemic, DeGrave and colleagues examined published models for detecting COVID-19 from radiographs. Many had been trained on datasets where positive and negative cases came from different sources. The models had learned to separate the sources, using markers such as text annotations on the films and patient positioning. Reported accuracy was high and largely illusory.

Oakden-Rayner and colleagues described a subtler version, which they called hidden stratification. A model can perform well on a labelled class overall while failing on an unlabelled subtype within it. In one of their examples, a pneumothorax detector appeared accurate largely because most positive images showed a chest drain, the visible sign of a case already treated. On the untreated cases that matter clinically, performance was much worse.

In each case the flaw was in how the data had been assembled. In each case the standard evaluation, drawn from the same flawed source, could not see it.

KEY TAKEAWAY

A validation score certifies that a model has learned its dataset. It says nothing about whether the dataset resembles the world. Cascades live in that gap.

Why the delay is the expensive part

Software engineering has long observed that defects cost more to fix the later they are found. Data defects follow the same logic with an added complication: a model does not fail loudly when its data is wrong. It trains, converges and reports a good number.

Sculley and colleagues, in an influential 2015 paper on technical debt in machine-learning systems, made a related point. Data dependencies cost more than code dependencies because they are harder to detect. A change in an upstream data source can silently alter a model's behaviour with no error and no failed build.

The practical consequence is that the cost of a data decision is not paid when it is made. It is paid after the model has been built on it, the product has been built on the model, and users have built their work on the product. By then, repair means re-collection, re-labelling, re-training, re-validation and, in regulated settings, re-approval.

Is this just a call for more process?

A reasonable sceptic would say that every engineering discipline produces papers lamenting that people do not document enough. Teams have deadlines. Exhaustive data validation before any modelling begins would mean many projects never start.

That is fair. The answer is not to front-load every possible check. Sambasivan's own conclusion is more specific: most cascades the practitioners described were avoidable through earlier, modest interventions. The study argues for a shift from treating data quality as "goodness of fit" to a model, measured at the end, toward what it calls goodness of data, attended to throughout.

A second objection is that foundation models change the picture, since a team fine-tuning a pretrained model handles far less data than one training from scratch. The volume is smaller. The exposure is not. A pretrained model adapted on a small, unrepresentative sample inherits that sample's assumptions with fewer examples to dilute them, and the evaluation set, typically drawn from the same sample, will again fail to notice.

Shortening the delay

If the defining feature of a cascade is the time between cause and discovery, the remedy is to shorten it.

Test against deployment conditions early. Before a dataset is finalised, collect a small sample from the real environment, with the real hardware, operators and constraints, and evaluate on it. The Thai clinics would have surfaced the image-quality issue with a few hundred photographs.

Put domain experts upstream. The person who knows that a chest drain signals a treated case should be involved when the labelling scheme is designed, not when the paper is reviewed.

Evaluate by slice. Aggregate metrics conceal exactly the failures described above. Results broken out by site, device, subgroup and subtype expose them.

Write down the assumptions. What was included, what was excluded, how categories were defined, what conditions the data was collected under. Documentation frameworks such as datasheets for datasets exist for this purpose, and their value is greatest at organisational boundaries.

Watch the inputs after launch. Production data drifts. Monitoring the distribution of what the model receives, not only what it outputs, is how the next cascade is caught early.

What this changes

FOR AI BUILDERS

For every dataset, state in writing how deployment data is expected to differ from it. If the answer is “it won’t”, test that claim first.

Look for shortcuts: can the model predict the data source, the device or the series site? If so, it may be using that to predict the label.

Hold out data by site, time or device, not only at random. Random splits hide distribution shift by construction.

FOR ENTERPRISES

Make data work visible in planning and review. If only model metrics are reported upward, only model work will be rewarded.

When data crosses a team or vendor boundary, require documentation of how it was collected, filtered and labelled.

Pilot in the least favourable realistic conditions, not the most favourable. The purpose of a pilot is to find the cascade.

THE INDIKA PERSPECTIVE

Cascades are an organisational problem before they are a technical one: the people who make data and the people who use it are separated, and the assumptions do not travel between them. A data partner earns its place by closing that gap. That means asking where and how the model will run before a single item is labelled, bringing domain knowledge into guideline design, and handing over datasets with their assumptions written down.

RESEARCH & REFERENCES

Sambasivan, N. et al. (2021). “Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI.” ACM CHI.

Beede, E. et al. (2020). “A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy.” ACM CHI.

Zech, J. et al. (2018). “Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study.” PLOS Medicine.

DeGrave, A., Janizek, J. and Lee, S.-I. (2021). “AI for radiographic COVID-19 detection selects shortcuts over signal.” Nature Machine Intelligence.

Oakden-Rayner, L. et al. (2020). "Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging." ACM CHIL.

Geirhos, R. et al. (2020). "Shortcut learning in deep neural networks." Nature Machine Intelligence.

Sculley, D. et al. (2015). "Hidden Technical Debt in Machine Learning Systems." NeurIPS.

Gebru, T. et al. (2021). "Datasheets for Datasets." Communications of the ACM.

Paleyes, A., Urma, R.-G. and Lawrence, N. (2022). "Challenges in Deploying Machine Learning: A Survey of Case Studies." ACM Computing Surveys.

“High Quality” Is Not a Property of Data

A dataset cannot be good in general. It can only be good for a purpose, and most quality metrics never ask which one.

7 MINUTE READ

In 2018, Joy Buolamwini and Timnit Gebru evaluated three commercial systems that classified gender from photographs of faces. For lighter-skinned men, the worst error rate among the three was 0.8%. For darker-skinned women, error rates ran as high as 34.7%.

The labels in the underlying training and benchmark data were not the issue. A photograph labelled as a man was, as far as anyone could tell, a photograph of a man. The issue was who was in the photographs. When the authors examined two standard face benchmarks of the time, they found that 79.6% of the faces in one and 86.2% in the other were lighter-skinned.

By the most common definition of data quality, the percentage of labels that are correct, those datasets were excellent. By any definition that mattered to a darker-skinned woman being classified by the resulting system, they were poor.

This is the central difficulty with the phrase “high-quality training data”. It is used as though quality were intrinsic to the data, like weight. It is not.

An older and better definition

The data-management field settled this question before modern machine learning existed. In 1996, Richard Wang and Diane Strong published a study that asked data consumers, not data producers, what quality meant to them. Their definition has been standard since: data quality is **fitness for use**.

Two things follow from that definition. First, quality cannot be assessed without knowing the use. Second, accuracy is only one of several things users care about. Wang and Strong identified fifteen dimensions in four groups: intrinsic qualities such as accuracy and believability, contextual qualities such as relevance, completeness and timeliness, representational qualities such as consistency and interpretability, and accessibility.

Machine learning rediscovered the same idea through failure. The pattern in the previous article in this series, in which a model validates well and then breaks in deployment, is what fitness for use looks like when it is absent.

Six questions that define quality for training data

Translating the idea into something a team can apply, the research literature supports at least six separate questions. A dataset can score well on some and badly on others.

FIGURE 5 · FRAMEWORK

Six questions that define training-data quality

DIMENSION	THE QUESTION	HOW IT CAN BE MEASURED
Label correctness	Are the labels right?	Audit against expert-adjudicated reference items; label-error detection
Coverage	Are the conditions of use present?	Example counts per defined slice: device, region, language, layout
Subgroup adequacy	Are important minorities within a class represented?	Per-slice evaluation; search for hidden strata
Consistency	Would the same item get the same label again?	Inter- and intra-annotator agreement, per category
Provenance	Do we know source, consent, licence and processing?	Completeness of documentation against a datasheet
Freshness	Does it still describe the world?	Drift between training data and recent production inputs

Synthesis by the editors, drawing on Wang and Strong (1996), Gebru et al. (2021), Koh et al. (2021) and Oakden-Rayner et al. (2020).

Are the labels correct? This is the dimension everyone measures. It matters, and methods such as confident learning and multi-annotator review address it. But it is the dimension on which most professionally produced datasets are already adequate.

Does it cover the conditions of use? Coverage means the range of inputs the model will meet is present in sufficient number: the lighting, the accents, the document layouts, the patient populations. The WILDS benchmark, assembled by Koh and colleagues from ten real-world datasets, documents consistent performance drops when models meet hospitals, regions, cameras or time periods that were absent from training. Coverage failures are invisible to accuracy audits because the examples that would reveal them are not there to audit.

Are subgroups represented where it counts? Coverage in aggregate is not enough. Hidden stratification, described by Oakden-Rayner and colleagues, occurs when a clinically important subtype is rare within a class and the model fails on it while the class-level metric looks fine.

Is it labelled consistently? Two annotators applying the same guidelines should produce compatible labels, and the same annotator should do so over time. Inconsistent labelling is different from wrong labelling: each individual label may be defensible, while the dataset as a whole teaches the model a boundary that moves.

Do we know where it came from? Provenance covers source, collection method, consent, licence and processing history. A large audit of more than 1,800 text datasets by the Data Provenance Initiative found licence information omitted for more than 70% of datasets on popular hosting platforms and miscategorised for more than half. Data of unknown origin carries legal exposure and, less obviously, technical risk: without knowing how it was filtered, nobody can say what is missing.

Is it still true? Language, products, fraud patterns and road layouts change. Lazaridou and colleagues showed that language models degrade measurably on text from periods after their training data ends. A dataset is a photograph of a distribution that keeps moving.

Why accuracy dominates anyway

If quality has six dimensions, why do most specifications mention one?

Partly because accuracy is easy to measure. You sample labelled items, have someone check them, and report a percentage. The other dimensions require knowing something about deployment, which the people producing the data often do not.

Partly because of how the work is bought. Annotation is commonly procured against an accuracy threshold in a service agreement. Coverage is the client's responsibility, defined by whatever raw data the client hands over. The vendor is accountable for labelling what it was given, correctly. Nobody is accountable for whether what was given was the right thing to label.

And partly because the research culture long treated datasets as fixed. Benchmark datasets were inputs to be downloaded, not artefacts to be examined. The data-centric AI movement, and benchmarks such as DataPerf that hold the model fixed and score improvements to the data, are a deliberate correction.

KEY TAKEAWAY

A specification that says “98% label accuracy” has defined one-sixth of data quality, and the sixth that was least likely to be the problem.

Is this unmeasurable?

The obvious objection is that “fitness for use” is a slogan, not a metric. Accuracy at least produces a number. If quality depends on context, every dataset needs a bespoke assessment and nothing can be compared or contracted.

The objection is weaker than it looks, because most of the dimensions can be measured once the use is stated.

Coverage can be measured by defining the slices that matter, such as device type, language, region or demographic group, and counting examples in each. Subgroup adequacy can be measured by evaluating on each slice separately. Consistency can be measured by inter-annotator and intra-annotator agreement. Provenance can be measured by completeness of documentation. Freshness can be measured by comparing the distribution of recent production inputs against the training set.

Tooling for parts of this is mature. Google’s data-validation work for production pipelines, and Amazon’s equivalent, check incoming data against an expected schema and distribution automatically. Documentation standards such as datasheets for datasets and data cards give a structure for recording the rest. An international standard series for data quality in analytics and machine learning, ISO/IEC 5259, now exists as well.

What cannot be avoided is the first step: someone has to say what the model is for, where it will run and on whom. That is uncomfortable because it is a product decision, not a data decision. But a dataset specification that omits it is not neutral. It has assumed that deployment will resemble whatever data happened to be available.

The limit of the argument

Fitness for use should not become an excuse for ignoring label accuracy. In domains where labels are hard, such as medical imaging or legal classification, accuracy may well be the binding constraint, and it is the subject of several later articles in this series.

There is also a real tension between dimensions. Maximising coverage of rare subgroups usually means collecting data from harder sources, which lowers label accuracy and agreement. Insisting on full provenance excludes data that would have improved coverage. Quality is a set of trade-offs, and a team that claims to have maximised every dimension has not looked closely at any of them.

Writing a better specification

A workable data specification answers the six questions explicitly.

It states the deployment conditions and the slices that matter. It sets minimum counts per slice, not only a total. It sets an accuracy target and says how it will be measured and by whom. It sets an agreement target, and says what happens to items on which annotators disagree. It requires documentation of source, consent and processing. And it sets a review date, after which the data is assumed stale until checked.

Most of that fits on a page. Writing the page is the hard part, because it forces the conversation about purpose that the word “quality” was being used to avoid.

What this changes

FOR AI BUILDERS

Define evaluation slices before collecting data, and track example counts per slice alongside total volume.

Report agreement and coverage next to accuracy in every dataset summary. A single number invites a single-dimension view.

Compare the distribution of live inputs to the training set on a schedule. Freshness is a measurable property.

FOR ENTERPRISES

Replace “high-quality data” in contracts and plans with the specific properties you need and how each will be verified.

Clarify who owns coverage. If the vendor labels what it is given and the client provides what it has, nobody is checking representativeness.

Require dataset documentation as a deliverable, with the same standing as the data.

THE INDIKA PERSPECTIVE

Asked for high-quality data, the right first response is a question: quality for what? A data partner that takes the accuracy target and starts labelling is doing what was asked and may be missing the larger risk. The more valuable service is helping a client define slices, find the gaps in what it already holds, and decide which dimension is the binding one for this model in this setting.

RESEARCH & REFERENCES

Buolamwini, J. and Gebru, T. (2018). “Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification.” Conference on Fairness, Accountability and Transparency (FAT*), PMLR.

Wang, R. and Strong, D. (1996). “Beyond Accuracy: What Data Quality Means to Data Consumers.” Journal of Management Information Systems.

Koh, P. W. et al. (2021). “WILDS: A Benchmark of in-the-Wild Distribution Shifts.” ICML.

Oakden-Rayner, L. et al. (2020). “Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging.” ACM CHIL.

Longpre, S. et al. (2024). “A large-scale audit of dataset licensing and attribution in AI.” Nature Machine Intelligence.

Lazaridou, A. et al. (2021). “Mind the Gap: Assessing Temporal Generalization in Neural Language Models.” NeurIPS.

Mazumder, M. et al. (2023). “DataPerf: Benchmarks for Data-Centric AI Development.” NeurIPS Datasets and Benchmarks.

Breck, E. et al. (2019). "Data Validation for Machine Learning." MLSys.

Schelter, S. et al. (2018). "Automating Large-Scale Data Quality Verification." Proceedings of the VLDB Endowment.

Gebru, T. et al. (2021). "Datasheets for Datasets." Communications of the ACM.

Pushkarna, M., Zaldivar, A. and Kjartansson, O. (2022). "Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI." ACM FAccT.

ISO/IEC 5259 series (2024). Artificial intelligence: Data quality for analytics and machine learning.

Disagreement Is Data

When two careful people label the same item differently, the usual response is to pick a winner. The evidence says that often throws away the most useful information in the dataset.

7 MINUTE READ

Take a sentence pair of the kind used in natural-language inference tasks. The premise: a man is standing outside a café holding a cup. The hypothesis: the man has bought a coffee. Does the first statement imply the second?

Ask one hundred people and you will not get one answer. Some will say yes, because that is the obvious reading. Some will say it is undetermined, because the cup might hold tea, or might be empty, or might belong to someone else. Both groups are reading carefully. They differ on how much everyday inference the task permits.

The conventional pipeline handles this by collecting three or five labels, taking the majority, and recording a single “gold” answer. The minority view is discarded as noise.

In 2019, Ellie Pavlick and Tom Kwiatkowski tested whether that was justified. They collected many more judgements per item than usual on inference datasets and looked at the distributions. For a substantial share of items, disagreement did not shrink as more annotators were added. The judgements clustered around two different answers. They also found that supplying more context did not make the disagreement go away. Their conclusion was that the disagreement was a property of the items, not of careless annotators.

A follow-up study, ChaosNLI, collected one hundred labels each for several thousand examples from popular inference benchmarks and found the same thing at scale: many items have a spread of human opinion, and models trained on single labels were poor at predicting that spread.

Four reasons people disagree

“Disagreement” covers different phenomena. Separating them is the first practical step, because each has a different remedy.

FIGURE 6 · FRAMEWORK

Four sources of annotator disagreement

SOURCE	WHAT IT LOOKS LIKE	HOW TO TELL	WHAT TO DO
Error	One annotator slipped or misread	They agree once shown the item again	Redundancy, review, retraining
Underspecified guideline	Annotators filled a gap in the instructions differently	The split follows a case the guideline does not cover	Revise the guideline; re-label affected items
Ambiguous item	The item supports more than one reading	Annotators understand each other and still differ	Record the distribution; use soft labels
Different perspective	The judgement depends on who is judging	Splits track annotator background or belief	Define whose view the data should represent; recruit to match

Synthesis by the editors, after Aroyo and Welty (2015), Plank (2022) and Uma et al. (2021).

Error. Someone was tired, rushed or misread the item. The label is wrong and they would agree if shown. Redundancy and review fix this.

Underspecified guidelines. The instructions did not say whether a partially hidden pedestrian counts, or whether sarcasm is “negative”. Annotators filled the gap differently. Each label is reasonable under some reading of the instructions. The fix is to the guideline, and re-labelling without changing it reproduces the split.

Ambiguity in the item. The café sentence. The item supports more than one reading and no guideline of reasonable length would settle every such case. The honest label is a distribution.

Difference in perspective. The annotators bring different knowledge, experience or values, and the task depends on them. Whether a remark is offensive is the standard example.

The first two are defects in the process. The last two are information about the world.

What the perspective evidence shows

The research on subjective tasks is by now extensive, and uncomfortable for anyone who assumes a neutral annotator exists.

Sap and colleagues showed in 2019 that tweets written in African American English were substantially more likely to be labelled offensive by annotators than other tweets, and that models trained on those labels reproduced the pattern. When annotators were primed to consider the dialect of the author, the rate fell. The labels had encoded the annotators’ unfamiliarity as a property of the text.

A later study by the same group, “Annotators with Attitudes”, measured annotators’ own beliefs and identities and found that they predicted toxicity ratings in systematic ways. Annotators who scored higher on measures of racist beliefs were less likely to rate anti-Black language as toxic and more likely to rate African American English as toxic. Work at Google on rater identity found comparable effects: raters who belonged to a group targeted by a comment rated it differently from those who did not.

In these settings, majority vote does not converge on truth. It converges on the view of whichever group supplies the majority of annotators.

Medicine shows a structurally similar pattern for a different reason. Studies of diabetic retinopathy grading found meaningful variation between individual ophthalmologists reading the same image. Here the disagreement reflects difficulty and expertise, and the remedy that worked was adjudication: specialists discussing cases until they reached a consensus that was better than any individual vote.

What to do with a split

Aroyo and Welty named the underlying assumption in a 2015 paper whose title is its argument: “Truth Is a Lie”. They listed seven myths of annotation, among them that each item has one correct answer, that disagreement is bad, and that more detailed guidelines always help. Barbara Plank later proposed a term now widely used, **human label variation**, to describe disagreement that should be modelled and not removed.

Several practical methods follow.

Keep the individual labels. The cheapest change is to stop discarding them. A dataset that stores every annotator’s judgement, with an annotator identifier, can support majority vote and everything else. One that stores only the aggregate cannot be un-aggregated.

Train on distributions. Peterson and colleagues collected about fifty human judgements each for ten thousand CIFAR-10 test images and trained classifiers on the resulting distributions instead of single labels. The models generalised better to new datasets and were more robust to adversarial perturbation. Where humans were unsure, the model learned to be unsure, and that turned out to be useful.

Model annotators, not only items. Davani, Díaz and Prabhakaran trained models that predicted each annotator’s label separately before combining them. Performance matched or beat majority-vote training, and the model’s internal spread gave a better estimate of when it was likely to be wrong.

Evaluate against the spread. If the test set forces a single answer onto items where humans split sixty-forty, forty per cent of the “error” assigned to a model is disagreement with a coin flip. Evaluating on the distribution, or separately on high- and low-agreement items, gives a truer picture.

KEY TAKEAWAY

When annotators split, ask why before asking who is right. A split caused by a vague guideline needs a better guideline. A split caused by an ambiguous item is the label.

The case for a single answer

There are good reasons the single-label convention persists.

Many downstream systems need a decision. A content-moderation queue removes a post or leaves it. A triage model flags a scan or does not. A distribution over labels has to be collapsed at some point, and someone has to choose the threshold.

Collecting enough labels per item to estimate a distribution is expensive. Five annotators cost five times what one costs, and most of the time, on most items, they agree.

And embracing disagreement can become an excuse. A team with sloppy guidelines can declare its noise to be “human label variation” and avoid the work of fixing them. Not every split is meaningful. Sometimes one annotator is simply wrong.

These objections argue for discipline, not for discarding the idea. The decision has to be made eventually, but it is better made by a system that knows which cases are contested. Redundant labelling can be targeted: collect more judgements on items where the first two annotators disagree, and stop early where they concur. And the way to tell error from ambiguity is to look, which is the purpose of the taxonomy above.

A working procedure

For teams that want to act on this without rebuilding their pipeline, a modest version is enough.

First, pilot the guideline on a small sample with several annotators, and read the disagreements. Most early disagreement is the second type, underspecification, and a round or two of guideline revision removes much of it.

Second, once the guideline is stable, classify the disagreement that remains. If annotators converge after discussion, it was error or oversight. If they understand each other and still differ, it is ambiguity or perspective.

Third, decide by design what the dataset should represent. For a factual extraction task, the answer may be adjudicated truth. For toxicity, it may be the distribution of views across a defined population, with the population specified and the annotator pool recruited to match it.

Fourth, record who the annotators were, in aggregate. A dataset of subjective judgements without a description of who judged is missing its most important metadata.

What this changes

FOR AI BUILDERS

Store per-annotator labels and annotator IDs. Aggregation should be a choice made at training time, not a loss made at collection time.

Split evaluation results by agreement level. A model that fails on high-agreement items has a real problem; one that “fails” on contested items may not.

Where disagreement is legitimate, try soft labels. Calibration often improves at little cost.

FOR ENTERPRISES

For subjective tasks, specify whose judgement the system should reflect, and confirm the annotator pool matches.

Treat persistent disagreement as a finding about your policy or taxonomy. It often reveals a category boundary the business has not actually decided.

Ask data suppliers for disagreement rates and how splits were resolved, not only a final accuracy figure.

THE INDIKA PERSPECTIVE

An annotation operation that measures success by agreement has an incentive to suppress disagreement: tighter guidelines, more similar annotators, quiet removal of contested items. That produces clean numbers and a dataset that misrepresents the world. The better practice is to treat disagreement as a signal to be triaged. Some of it exposes a guideline that needs work. Some of it is the most valuable thing the annotators have told you.

RESEARCH & REFERENCES

Pavlick, E. and Kwiatkowski, T. (2019). “Inherent Disagreements in Human Textual Inferences.” Transactions of the Association for Computational Linguistics.

Nie, Y., Zhou, X. and Bansal, M. (2020). “What Can We Learn from Collective Human Opinions on Natural Language Inference Data?” EMNLP.

Aroyo, L. and Welty, C. (2015). “Truth Is a Lie: Crowd Truth and the Seven Myths of Human Annotation.” AI Magazine.

Plank, B. (2022). “The ‘Problem’ of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation.” EMNLP.

Sap, M. et al. (2019). “The Risk of Racial Bias in Hate Speech Detection.” ACL.

Sap, M. et al. (2022). “Annotators with Attitudes: How Annotator Beliefs And Identities Bias Toxic Language Detection.” NAACL.

Goyal, N. et al. (2022). “Is Your Toxicity My Toxicity? Exploring the Impact of Rater Identity on Toxicity Annotation.” ACM CSCW.

Peterson, J. et al. (2019). “Human Uncertainty Makes Classification More Robust.” IEEE ICCV.

Davani, A. M., Díaz, M. and Prabhakaran, V. (2022). "Dealing with Disagreements: Looking Beyond the Majority Vote in Subjective Annotations." Transactions of the Association for Computational Linguistics.

Uma, A. et al. (2021). "Learning from Disagreement: A Survey." Journal of Artificial Intelligence Research.

Krause, J. et al. (2018). "Grader Variability and the Importance of Reference Standards for Evaluating Machine Learning Models for Diabetic Retinopathy." Ophthalmology.

Agreement Is Not Truth

Inter-annotator agreement is the most widely reported quality metric in data annotation. It measures whether people gave the same answer, which is a different thing from whether the answer was right.

7 MINUTE READ

Two annotators label one hundred medical images as either “abnormal” or “normal”. They give the same label to ninety of them. Is that good?

Most people would say yes. Ninety per cent agreement sounds like a reliable process. Now add one detail. In all ninety cases of agreement, both annotators said “abnormal”. In the ten cases of disagreement, one said abnormal and the other normal, five each way. Neither annotator ever agreed with the other that an image was normal.

The standard statistic for this situation is Cohen’s kappa, which corrects observed agreement for the agreement expected by chance. Each annotator called 95% of images abnormal, so two people labelling independently at those rates would agree about 90.5% of the time. Observed agreement, at 90%, is slightly below that. Kappa comes out at roughly minus 0.05.

By the conventional reading, these two annotators agree no better than chance. By the raw percentage, they agree nine times in ten. Both numbers are computed correctly from the same table.

FIGURE 7 · ANNOTATED EXAMPLE

Same agreement, very different kappa

Skewed labels

	B: abnormal	B: normal
A: abnormal	90	5
A: normal	5	0

Observed agreement **90%**
Agreement expected by chance **90.5%**
Cohen’s kappa **−0.05**

Balanced labels

	B: abnormal	B: normal
A: abnormal	45	5
A: normal	5	45

Observed agreement **90%**
Agreement expected by chance **50%**
Cohen’s kappa **0.80**

Worked example, 100 items and two annotators in each case. Kappa = (observed – chance) ÷ (1 – chance).

This is a known result, described by Feinstein and Cicchetti in 1990 as the paradox of high agreement but low kappa. It is also a good entry point to a larger problem. Agreement metrics are treated as measures of data quality. They measure something narrower, and they have properties that surprise people who rely on them.

What agreement statistics do

Cohen introduced kappa in 1960 to answer a specific question. If two raters assign items to categories, how much of their agreement exceeds what would occur if they were assigning at random according to their own base rates? Fleiss extended the idea to multiple raters. Krippendorff's alpha generalises further, to any number of raters, missing data and ordinal or interval scales.

All three subtract chance agreement, and that subtraction is the source of both their value and their oddities. When one category is very common, chance agreement is high and there is little room above it. The statistic becomes unstable, and small changes in a handful of rare-category items move it a great deal. Since most real annotation tasks are imbalanced, with fraud, defects, tumours and toxic comments all being rare, this is the normal case.

Then there is the question of what counts as a good value. The most cited scale comes from Landis and Koch in 1977: values above 0.80 are “almost perfect”, 0.61 to 0.80 “substantial”, and so on. It is cited in thousands of papers as though it were a standard. The authors themselves described the divisions as clearly arbitrary, offered as a convenience for discussing one example. Krippendorff recommends 0.80 for reliable conclusions and 0.667 as the lowest level for tentative ones, and has been explicit that even these depend on what is at stake.

Artstein and Poesio's survey of agreement in computational linguistics remains the best single treatment. Its recommendation is that researchers report the coefficient, the method, the number of annotators and categories, and the distribution, because the bare number cannot be interpreted without them. That advice is rarely followed.

Reliable is not the same as valid

Measurement theory distinguishes reliability from validity. A bathroom scale that reads three kilograms heavy every time is perfectly reliable and wrong. Agreement is a reliability measure. It says annotators behaved consistently with each other. It cannot say whether they were consistently correct.

There are at least three common ways to obtain high agreement on wrong labels.

Shared blind spots. If all annotators lack the same piece of expertise, they will make the same mistake. In the diabetic retinopathy literature, majority votes of general graders missed subtle findings that a panel of retinal specialists identified on discussion. The majority agreed. It was wrong in a consistent direction.

Guidelines that buy agreement. The quickest way to raise kappa is to add rules: when in doubt, choose category A; ignore anything smaller than twenty pixels; treat sarcasm as literal. Agreement rises because annotators are now applying the same convention, whether or not the convention matches what the model needs to learn.

Anchoring on a suggestion. When annotators correct machine-generated pre-labels instead of labelling from scratch, they tend to accept what they are shown. Studies of pre-annotation in linguistic and clinical text have found it speeds work and raises agreement, while also noting the risk that annotators reproduce the system's errors. Two people who both accepted the same wrong suggestion are in full agreement.

In each case the agreement metric improves as the data gets worse.

Low agreement that is fine

The converse also holds. Agreement can be low on a dataset that is doing its job.

As the previous article in this series argued, some tasks do not have single answers. If the task is to rate how offensive a remark is, and the annotators were recruited to reflect a varied population, low agreement is an accurate measurement of a real spread in opinion. Raising it by homogenising the annotator pool would make the data less representative.

Reidsma and Carletta made a sharper version of the point in 2008. They simulated annotation with different kinds of disagreement and trained classifiers on the results. Disagreement that looked like random noise did relatively little harm to a learner even when agreement was modest. Disagreement that was systematic, where one annotator consistently differed in a patterned way, could mislead a learner even when overall agreement was high. Their conclusion was that the agreement coefficient alone does not tell you whether data is safe to learn from. The structure of the disagreement does.

KEY TAKEAWAY

Agreement tells you whether annotators understood the task the same way. It does not tell you whether they understood it correctly, and no threshold on kappa converts one into the other.

In defence of the metric

None of this means agreement should not be measured. The case for it is strong.

It is cheap and it requires no ground truth. In most annotation projects there is no independent source of correct answers to compare against, and agreement is the only quality signal available at scale. It is an excellent diagnostic of guidelines: when agreement on a category collapses, the definition of that category is almost always the cause. And low agreement on a task that ought to have clear answers is real evidence of a problem, even if high agreement is not proof of its absence.

The trouble is with how it is used: as a single headline number, compared with a borrowed threshold, and reported as though it certified correctness.

Using agreement properly

A few habits make agreement statistics far more informative.

Report it per category. An overall kappa of 0.75 can conceal one category at 0.95 and another at 0.30. The second is where the model will fail.

Report prevalence alongside it. With the distribution of labels visible, a reader can tell whether a low kappa reflects real confusion or the imbalance effect. For rare classes, positive and negative agreement reported separately are more interpretable than a single chance-corrected figure.

Look at the confusion pattern. Which pairs of categories are confused, and by which annotators? Random scatter suggests difficulty. A consistent pattern from one person suggests a different reading of the guidelines.

Measure validity separately. Seed the work with items whose answers have been established by expert adjudication, and score annotators against them. This is a different measurement from agreement, and it is the only one that speaks to correctness.

Check agreement with experts, not only between peers. If generalist annotators agree with each other at 0.85 and with a domain expert at 0.55, the first number describes a shared misunderstanding.

Measure the same annotator twice. Intra-annotator agreement, obtained by quietly re-presenting a sample of items some weeks later, shows whether one person applies the guideline consistently over time. Someone who disagrees with their own earlier labels cannot be expected to agree with anyone else's, and the usual cause is a guideline that leaves too much to memory and mood.

Be suspicious of sudden improvement. If agreement jumps after a guideline revision or after introducing pre-labels, find out whether annotators became more accurate or more alike.

What this changes

FOR AI BUILDERS

Never accept a dataset on a single agreement figure. Ask for per-class values, label prevalence and the confusion matrix between annotators.

Maintain an expert-adjudicated reference set and track accuracy against it as a separate metric from agreement.

When pre-labels are introduced, measure how often annotators override them on items known to be wrong.

FOR ENTERPRISES

Treat agreement thresholds in vendor agreements as a floor on process consistency, not a guarantee of correctness.

In specialised domains, require that agreement be measured against qualified experts as well as among annotators.

Ask what was done to reach the reported agreement. Guideline shortcuts that raise the number may lower the value of the data.

THE INDIKA PERSPECTIVE

An honest quality report separates two questions that the industry usually merges: did annotators agree, and were they right. The first is measured by agreement statistics, with their caveats stated. The second needs an independent reference, built by people with the expertise to adjudicate. Reporting both, per category, is more work and produces less flattering numbers. It is also the only version a model team can act on.

RESEARCH & REFERENCES

Cohen, J. (1960). "A Coefficient of Agreement for Nominal Scales." Educational and Psychological Measurement.

Fleiss, J. (1971). "Measuring nominal scale agreement among many raters." Psychological Bulletin.

Krippendorff, K. (2018). Content Analysis: An Introduction to Its Methodology, 4th edition. SAGE.

Landis, J. R. and Koch, G. (1977). "The Measurement of Observer Agreement for Categorical Data." Biometrics.

Feinstein, A. and Cicchetti, D. (1990). "High agreement but low kappa: I. The problems of two paradoxes." Journal of Clinical Epidemiology.

Artstein, R. and Poesio, M. (2008). "Inter-Coder Agreement for Computational Linguistics." Computational Linguistics.

Reidsma, D. and Carletta, J. (2008). "Reliability Measurement without Limits." Computational Linguistics.

Krause, J. et al. (2018). "Grader Variability and the Importance of Reference Standards for Evaluating Machine Learning Models for Diabetic Retinopathy." Ophthalmology.

Fort, K. and Sagot, B. (2010). "Influence of Pre-annotation on POS-tagged Corpus Development." ACL Linguistic Annotation Workshop.

Lingren, T. et al. (2014). "Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical trial announcements." Journal of the American Medical Informatics Association.

The Arithmetic of Trust

Majority voting rests on a two-century-old theorem with two assumptions. Annotation pipelines violate both, and the mathematics of what to do about it has existed since 1979.

7 MINUTE READ

In 1785 the Marquis de Condorcet proved a theorem about juries. If each juror is more likely than not to reach the correct verdict, and jurors decide independently, then the probability that the majority is correct rises toward certainty as the jury grows.

Modern data annotation runs on this theorem, mostly without knowing it. Collect three labels per item, take the majority, call it ground truth. The reasoning is Condorcet's: individuals err, crowds converge.

The arithmetic is worth seeing, because it shows both the strength of the idea and where it stops.

FIGURE 8 · DATA VISUALISATION

How often a majority vote is right

EACH ANNOTATOR IS RIGHT	1 VOTE	3 VOTES	5 VOTES	7 VOTES	9 VOTES
60% of the time	60.0%	64.8%	68.3%	71.0%	73.3%
70%	70.0%	78.4%	83.7%	87.4%	90.1%
80%	80.0%	89.6%	94.2%	96.7%	98.0%
90%	90.0%	97.2%	99.1%	99.7%	99.9%

Calculated from the binomial distribution for a two-class task. Assumes annotators are equally accurate and err independently. Correlated errors make the true figures lower.

Three annotators who are each right 80% of the time produce a correct majority 89.6% of the time. Five produce 94.2%. Redundancy works, and it works quickly.

Now look at the top row. Annotators who are right 60% of the time need nine votes to reach 73%. And at any level of individual skill, the calculation assumes that annotators' mistakes are independent. If they all misread the same guideline, or lack the same expertise, or accepted the same machine-generated suggestion, adding more of them adds confidence without adding accuracy.

Condorcet's theorem also has a less quoted corollary. If each juror is worse than chance, the majority converges on the wrong answer.

Not all annotators are equal

In 1979, two statisticians, Philip Dawid and Allan Skene, addressed a problem from clinical medicine. Several clinicians had each recorded a patient's condition, their records disagreed, and there was no independent way to know who was right. Simple voting treated every clinician as equally reliable. Dawid and Skene proposed something better.

Their model gives each rater a **confusion matrix**: the probability that this rater says each category, given the true category. The true labels are unknown, but the model can be fitted by alternating two steps. Given a current guess at the true labels, estimate each rater's confusion matrix. Given those matrices, re-estimate the true labels, weighting each rater by their estimated reliability. Repeat until stable.

The result is an estimate of the most likely true label for every item, and an estimate of every rater's pattern of error, with no ground truth required.

Three ideas in that paper still organise the field.

Reliability is per annotator. A vote from a careful expert and a vote from someone guessing should not count the same.

Reliability is per category. An annotator may be excellent at identifying one class and poor at another. A confusion matrix captures that; a single accuracy score does not.

Bias is recoverable. This is the least intuitive and most useful point. Suppose an annotator consistently labels "severe" whenever the truth is "moderate". Their accuracy is poor. But their labels are highly informative, because the error is predictable and can be inverted. Ipeirotis, Provost and Wang developed this distinction for crowdsourcing: an annotator who is biased but consistent is far more valuable than one who is unbiased but random, and simple accuracy scores cannot tell them apart.

What the crowdsourcing literature added

The arrival of online labour markets in the 2000s turned Dawid and Skene's clinical problem into an industrial one, and produced a large body of evidence.

Snow and colleagues, in a much-cited 2008 study, compared non-expert labels from Amazon Mechanical Turk with expert labels across five language tasks. For tasks such as rating the emotional tone of headlines, an average of four non-expert labels per item matched the quality of an expert's. The paper's title asked whether cheap and fast could also be good, and for those tasks the answer was yes.

Sheng, Provost and Ipeirotis examined the trade-off between labelling more items and re-labelling the same ones. When individual labels are noisy, they found, collecting additional labels for existing items can improve a model more than collecting new items, and selectively re-labelling the items most likely to be wrong is better still.

Later models added more structure. Whitehill and colleagues modelled item difficulty alongside annotator ability, since a disagreement on a hard item says less about an annotator than a disagreement on an easy one. Raykar and colleagues trained the classifier and the annotator model jointly, so that the model's own predictions helped decide whom to trust.

A sober note comes from a 2017 benchmark by Zheng and colleagues, which compared seventeen truth-inference algorithms across several real crowdsourced datasets. Its title asked whether the problem was solved, and the answer was no: no algorithm was best across all datasets, and on some tasks plain majority voting was competitive with far more elaborate methods. When annotators are of similar quality and enough labels are collected, the sophistication buys little. The models earn their keep when quality is uneven, redundancy is thin or errors are systematic.

When the independence assumption fails

Every method above still assumes that, given the true label, annotators err independently. Three developments have made that harder to believe.

Shared training. Professional annotation teams are trained together on the same guidelines by the same leads. That is good for consistency and bad for independence. Their errors are correlated by design.

Shared pre-labels. When a model proposes a label and humans confirm it, several humans confirming the same proposal are not several independent opinions.

Shared tools. In 2023, Veselovsky, Ribeiro and West re-ran a text summarisation task on Mechanical Turk and estimated, using keystroke patterns and a text classifier, that between 33% and 46% of crowd workers had used a large language model to produce their answers. Workers who paste a task into the same chatbot will agree with one another closely. A redundancy scheme would read that as high confidence.

The implication is that redundancy now needs diversity to be worth anything. Three labels from three annotators with different backgrounds, working without a shared suggestion, are stronger evidence than five labels from a homogeneous team correcting the same pre-label.

KEY TAKEAWAY

Majority vote is the right method when annotators are equally skilled and independently wrong. Those are assumptions to be tested, and in a modern pipeline they usually fail.

What voting has going for it

Weighted aggregation has costs that its advocates understate.

It is harder to explain. “Three people looked and two said yes” is something a client, an auditor or a regulator can follow. “The posterior probability under a fitted latent-class model is 0.83” is not.

It needs data. Estimating a confusion matrix for each annotator requires each annotator to have labelled enough items that overlap with others. In a fast-moving project with a rotating workforce, the estimates can be too noisy to help.

And it can entrench the majority. An unsupervised model learns reliability from agreement, so an annotator who is right when the group is wrong, perhaps the only domain expert in the pool, can be scored as unreliable and down-weighted. Without some ground truth to anchor the model, it measures conformity.

The last point is the reason no production system should rely on aggregation alone.

A practical trust system

In practice, good operations combine several signals.

Gold items. A fraction of each annotator’s queue consists of items with known answers, established by experts. This gives a direct measure of accuracy and anchors everything else.

Per-annotator, per-class statistics. Track confusion patterns, not only overall accuracy. An annotator who systematically confuses two categories needs targeted retraining, and their past labels in those categories can be flagged for review.

Adaptive redundancy. Send each item to two annotators. If they agree, accept it. If they disagree, send it to a third, or to a senior reviewer. This concentrates the labelling budget on the hard items, which is where it changes outcomes.

Diversity by design. For the items that get multiple labels, route them to annotators who are unlikely to share errors: different teams, different sites, with pre-labels hidden from at least one.

Weighted aggregation where the data supports it. With enough overlap and a gold anchor, a Dawid–Skene–style model will outperform voting. Use it there.

What this changes

FOR AI BUILDERS

Log every individual label with an annotator identifier and a timestamp. Without that, none of these methods is available later.

Estimate per-annotator confusion matrices against gold items and use them to weight votes and to target retraining.

Check for correlated errors: compare agreement among annotators who saw the same pre-label with agreement among those who did not.

FOR ENTERPRISES

Ask how redundancy is allocated. Uniform triple-labelling of everything is a sign that budget is being spent on easy items.

Ask how the supplier knows its annotators' errors are independent, especially where AI assistance is used.

Require a gold-standard programme with expert-set answers, and ask to see accuracy against it by annotator cohort.

THE INDIKA PERSPECTIVE

Trust in an annotation workforce should be measured, specific and revisable: this person, on this category, at this level of confidence, as of this month. That is more demanding than a headcount and a consensus rule, and it changes what a data operation can promise. It can route hard items to the people who are demonstrably good at them, catch drift in an individual before it contaminates a batch, and say with evidence how much a given label should be believed.

RESEARCH & REFERENCES

Dawid, A. P. and Skene, A. M. (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm." *Journal of the Royal Statistical Society, Series C (Applied Statistics)*.

Snow, R. et al. (2008). "Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks." *EMNLP*.

Sheng, V., Provost, F. and Ipeirotis, P. (2008). "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers." *ACM KDD*.

Ipeirotis, P., Provost, F. and Wang, J. (2010). "Quality Management on Amazon Mechanical Turk." *ACM SIGKDD Workshop on Human Computation*.

Whitehill, J. et al. (2009). "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise." *NeurIPS*.

Raykar, V. et al. (2010). "Learning From Crowds." *Journal of Machine Learning Research*.

Zheng, Y. et al. (2017). "Truth Inference in Crowdsourcing: Is the Problem Solved?" *Proceedings of the VLDB Endowment*.

Paun, S. et al. (2018). "Comparing Bayesian Models of Annotation." *Transactions of the Association for Computational Linguistics*.

Veselovsky, V., Ribeiro, M. H. and West, R. (2023). "Artificial Artificial Intelligence: Crowd Workers Widely Use Large Language Models for Text Production Tasks." *arXiv preprint*.

Condorcet, M. de (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*.

The Expert Premium

A specialist's hour costs many times a generalist's. Whether that is expensive depends entirely on what the generalist's label is worth.

7 MINUTE READ

To train one of the first deep-learning systems for detecting diabetic eye disease, a Google team needed labels for 128,175 retinal photographs. They did not use a crowd. They used a panel of 54 ophthalmologists and senior ophthalmology residents licensed in the United States, and had each image graded between three and seven times.

Published in JAMA in 2016, the resulting system showed high sensitivity and specificity for referable disease on two validation sets. It is among the most cited results in medical AI. It is also an annotation project of unusual cost: several hundred thousand individual judgements by physicians.

Eight years earlier, a different and equally cited study had reached what looks like the opposite conclusion. Snow and colleagues showed that for several language tasks, four non-expert workers recruited online could match the label quality of one expert, at a small fraction of the price.

Both results are sound. The interesting question is what separates the tasks in the first study from the tasks in the second, because that boundary determines the economics of every data budget in a specialised field.

Three kinds of task

A useful way to sort annotation tasks is by what a non-expert lacks.

Nothing. Is this headline positive or negative? Is there a car in this image? Any attentive adult has the knowledge required. Individual error comes from inattention, and inattention is roughly random, so redundancy cancels it. This is the regime of Snow's result.

Training. Draw a box around every pedestrian, including partly hidden ones, following a forty-page specification. The required knowledge is the specification itself. It can be taught in days, and performance then depends on diligence and practice. Most commercial annotation sits here.

Years. Does this retinal image show moderate or severe non-proliferative retinopathy? Is this contract clause an uncapped indemnity? Is this code change safe to merge? The required knowledge cannot be written into guidelines of any practical length, because it is the professional training itself.

In the third category, the logic of redundancy breaks. Four non-experts do not average out to one expert, because their errors are not random. They are the same error: the one that comes from not knowing. As the previous article showed, majority voting amplifies shared mistakes.

The evidence from three professions

Medicine. The Google retinopathy work was followed by a study of the labels themselves. Krause and colleagues found that even among ophthalmologists, individual graders varied considerably, and that an adjudicated consensus of retinal specialists identified disease that a majority vote of individual graders missed. Expertise turned out to have levels, and the reference standard mattered at each one. Work on chest radiography by Majkowska and colleagues took the same approach, building reference standards through adjudication by panels of radiologists because single-reader labels were not reliable enough to evaluate against.

Law. The Contract Understanding Atticus Dataset was built to train models to find important clauses in commercial contracts. Its more than 13,000 annotations were produced by dozens of legal experts, and its authors describe the cost of expert annotators as the main bottleneck for specialised domains. LegalBench, a later benchmark of 162 legal reasoning tasks, was assembled through contributions from legal professionals for the same reason: nobody else could write the tasks or verify the answers.

What happens without that expertise in the loop is now documented. A Stanford study led by Magesh evaluated commercial AI legal research tools marketed as reliable, and found they produced incorrect or misgrounded answers between 17% and 33% of the time. Detecting those errors required lawyers reading each response against the cited authorities.

Finance. FinanceBench tested language models on questions about public company filings, with answers and supporting evidence annotated by hand. A leading model equipped with a retrieval system answered incorrectly or declined to answer 81% of a sample of questions. Again, establishing that required someone who could read a filing and know the right answer.

In each field, expertise is needed twice: once to create training signal, and again, more stringently, to evaluate whether the model learned it.

The yield calculation

The standard objection to expert annotation is price. A specialist's time costs many multiples of a generalist's. On a cost-per-label basis the comparison looks hopeless.

Cost per label is the wrong unit. The quantity that matters is cost per correct label, which depends on yield.

FIGURE 9 · FRAMEWORK

What the non-expert lacks decides who should label

A NON-EXPERT LACKS	TYPICAL TASKS	PATTERN OF ERROR	DOES REDUNDANCY HELP?	SENSIBLE DESIGN
Nothing	Sentiment, object presence, simple relevance	Random lapses of attention	Yes; a few votes approach expert quality	Generalists with modest redundancy
Training	Detailed bounding, entity tagging to a long specification	Guideline misreadings, reducible with practice	Partly; training helps more	Trained, stable teams with gold-item checks
Years of practice	Clinical grading, legal clause review, code safety	Shared and systematic: the same wrong answer	No; votes reinforce the shared error	Experts for adjudication, hard items and evaluation sets

Synthesis by the editors, after Snow et al. (2008), Gulshan et al. (2016) and Krause et al. (2018).

Consider a task where a generalist is right 60% of the time and wrong in a consistent direction on the rest. Three generalist votes do not rescue it. The 40% of items on which they are wrong are items where they tend to be wrong together. The dataset is delivered at low cost, with a large fraction of labels incorrect and no indication of which.

The downstream costs then arrive. The model learns the systematic error. Evaluation, if it uses labels from the same pool, cannot detect it. The problem surfaces in deployment, as the cascades article described, and the remedy is to re-label with experts anyway, having also paid for the first attempt and the failed model.

On tasks of the first and second kind, generalists have a yield close to that of experts and the cheap option is correct. On tasks of the third kind, the cheap option's yield can be low enough that its true cost per usable label exceeds the expert's.

KEY TAKEAWAY

The question is never “can we afford experts?” It is “on which items does an expert give a different answer from a trained generalist?” Those items, and only those, are worth the premium.

Making expertise go further

Expert time is scarce as well as expensive. In many health systems there are too few radiologists to read clinical scans, let alone to label training data. The practical problem is allocation, and several designs help.

Tiering. Trained generalists handle the full volume and flag items they find uncertain or that fall into predefined hard categories. Experts see only the flagged subset. This works when generalists know what they do not know, which has to be verified, not assumed.

Adjudication, not labelling. Experts are most valuable resolving disagreements, as the retinopathy studies showed. Two trained annotators label independently, and an expert sees only the items where they differ.

Experts write, others apply. Much of an expert’s knowledge can be partially transferred into guidelines, worked examples and decision trees. An hour of a specialist’s time spent improving a guideline can raise the quality of thousands of generalist labels. An hour spent labelling produces a few dozen.

Reserve experts for evaluation. As the article on test sets argued, evaluation data is small and deserves the highest standard. If the expert budget is limited, the test set is the place to spend it first.

Use the model to route. Active learning and uncertainty estimates can direct expert attention to the examples a model finds hardest. A later article in this series examines how well that works.

Where the premium is overpaid

The argument can be pushed too far, and vendors of expert annotation have reasons to push it.

Credentials are not skill at annotation. A senior physician may be an inconsistent labeller, impatient with guidelines and prone to applying personal conventions. Expert disagreement is real, as Krause’s work shows, and a single expert label is not ground truth. Expert panels need the same agreement monitoring and adjudication as any other annotators.

Many tasks in specialised fields are not specialised tasks. Extracting a patient's date of birth from a clinical note does not need a clinician. Classifying a legal document as a lease or a licence may not need a lawyer. Decomposing a workflow often shows that a small share of the decisions require expertise and the rest require care.

And the boundary moves. As models improve, tasks that needed training can be pre-labelled, and tasks that needed experts can be drafted for expert review. The premium does not disappear, but it concentrates on a smaller and harder set of judgements.

A test before you commit

Before commissioning a large annotation project in a specialised domain, a small experiment settles most of the question.

Take two hundred representative items. Have them labelled by trained generalists and, independently, by two or three qualified experts with adjudication. Compare.

If generalists match the adjudicated expert labels at a high rate, the task is of the first or second kind and expert annotation is unnecessary for the bulk of it. If they diverge, look at where. Random scatter suggests a training or guideline problem. A consistent pattern on particular item types identifies exactly which items need an expert. The experiment costs little, and it replaces an argument about hourly rates with a measurement of yield.

What this changes

FOR AI BUILDERS

Run the generalist-versus-expert comparison on a sample before fixing the labelling design. Let measured divergence decide the allocation.

Decompose specialised tasks. Route sub-decisions by the knowledge they require, not the domain they belong to.

Build evaluation sets with adjudicated expert labels even when training data is produced more cheaply.

FOR ENTERPRISES

Compare suppliers on cost per verified-correct label for your task, not on price per item.

In regulated or high-stakes domains, require evidence of annotator qualifications and of how expert disagreement is resolved.

Treat your own senior practitioners' time as a data asset. A few hours spent on guidelines and adjudication can be worth more than any external labelling budget.

THE INDIKA PERSPECTIVE

Domain expertise is the least scalable input in AI data and the one most often wasted, either by not using it where it is needed or by spending it on work that does not require it. The operational skill is in the routing: knowing, with evidence, which items need a specialist, getting those items in front of one, and using everything the specialist decides to improve what the rest of the workforce does next.

RESEARCH & REFERENCES

- Gulshan, V. et al. (2016). "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs." *JAMA*.
- Snow, R. et al. (2008). "Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks." *EMNLP*.
- Krause, J. et al. (2018). "Grader Variability and the Importance of Reference Standards for Evaluating Machine Learning Models for Diabetic Retinopathy." *Ophthalmology*.
- Majkowska, A. et al. (2020). "Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation." *Radiology*.
- Hendrycks, D. et al. (2021). "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review." *NeurIPS Datasets and Benchmarks*.
- Guha, N. et al. (2023). "LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models." *NeurIPS Datasets and Benchmarks*.
- Magesh, V. et al. (2025). "Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools." *Journal of Empirical Legal Studies*.
- Islam, P. et al. (2023). "FinanceBench: A New Benchmark for Financial Question Answering." *arXiv preprint*.
- Singhal, K. et al. (2023). "Large language models encode clinical knowledge." *Nature*.

Annotation Is Moving Up the Stack

Models now draw most of the boxes and write most of the first-pass labels. The human work that remains is smaller in volume, harder in kind, and more consequential per decision.

7 MINUTE READ

When Meta's research group built the dataset behind its Segment Anything Model, it needed outlines, known as masks, for objects in millions of photographs. The final dataset contains 1.1 billion masks across 11 million images, far more than any previous segmentation dataset.

Human annotators drew almost none of them. According to the paper, 99.1% of the masks were generated fully automatically.

It would be easy to read that as the end of manual annotation. The details of how the dataset was built suggest something more interesting.

How a data engine works

The team describes a three-stage process it calls a data engine.

In the first stage, professional annotators labelled masks by clicking on objects, with an early version of the model proposing an outline for them to refine. As the model was retrained on their output, it improved, and the average time to annotate a mask fell from 34 seconds to 14. This stage produced 4.3 million masks.

In the second stage, the model automatically labelled the objects it was confident about, and annotators were asked to add the ones it had missed. Their work became harder: the remaining objects were the less obvious ones, and average time per mask went back up to 34 seconds. This stage added 5.9 million masks.

In the third stage, the model ran alone and generated the rest.

Three things stand out. Human effort was front-loaded: about 10 million human-involved masks made the next billion possible. The human task changed as the model improved, from drawing everything to finding what the machine had failed

to see. And the time per human-labelled item did not fall in the long run. It fell while the model was absorbing easy work and rose again once only hard work remained.

FIGURE 10 · PROCESS MAP

The Segment Anything data engine, stage by stage

STAGE	WHAT PEOPLE DID	MASKS PRODUCED	AVERAGE HUMAN TIME PER MASK
1. Assisted-manual	Clicked on objects; refined the model's proposed outline	4.3 million	Fell from 34 to 14 seconds as the model improved
2. Semi-automatic	Added the objects the model had missed	5.9 million more	Returned to 34 seconds; only harder objects remained
3. Fully automatic	Nothing; the model ran alone	Total reaches 1.1 billion	None

99.1% of final masks generated automatically

0.9% human-involved

Source: Kirillov et al. (2023), "Segment Anything", Section 4.

That pattern, a shrinking volume of human work that grows harder per item, appears wherever AI-assisted annotation has been studied.

Language models as annotators

Text annotation has moved in the same direction, with large language models acting as first-pass labellers.

A widely cited 2023 study in PNAS by Gilardi, Alizadeh and Kubli compared ChatGPT with crowd workers on classification tasks over tweets and news articles: relevance, stance, topic and framing. With no task-specific training, the model's accuracy exceeded that of crowd workers on most tasks, and its consistency across repeated runs exceeded that of both crowd workers and trained annotators. The cost per annotation was under a third of a US cent, more than an order of magnitude cheaper than the crowd.

Earlier work had already shown that annotation could be partly programmed. The Snorkel system lets subject-matter experts write labelling functions, simple rules or heuristics that each label some examples imperfectly, and then combines them statistically. A case study at Google reported that this approach let teams build classifiers using organisational knowledge in place of large volumes of hand labels.

So for a large class of tasks, the first-pass label is now a machine's. The evidence on what follows is more mixed.

What the optimistic reading leaves out

Performance varies by task, unpredictably. Follow-up studies testing language models across many annotation tasks found accuracy ranging from excellent to poor, with no reliable way to tell in advance which a new task would be. Pangakis and colleagues, after testing across a range of social-science tasks, titled their paper with its recommendation: automated annotation with generative AI requires validation. That means human-labelled data for every new task, to check the machine against.

The comparison baseline was weak. Crowd workers on general-purpose platforms are not the standard a professional annotation operation should be measured against. Showing that a model beats anonymous, lightly screened workers on tweet classification says little about how it compares with trained annotators on a client's own taxonomy, and less about expert tasks.

Consistency is not correctness. A model gives the same answer every time, which looks like reliability. As the agreement article argued, it is reliability in the narrow sense. A model that systematically mislabels a category will do so with perfect self-agreement.

Humans anchor on suggestions. When people review machine output instead of labelling from scratch, they tend to accept what they see. Studies of pre-annotation in corpus linguistics found gains in speed and consistency along with a measurable pull toward the pre-annotator's errors. The broader literature on automation bias, reviewed by Parasuraman and Manzey, shows that this tendency affects experts as well as novices and is not eliminated by training or instructions to be careful.

The annotators may already be using the model. Veselovsky and colleagues estimated that a third to nearly half of crowd workers on one text task used a language model to produce their answers. Where that happens, "human review" of machine output is partly the machine reviewing itself.

KEY TAKEAWAY

AI assistance makes the average label cheaper and the average human decision harder. The risk shifts from annotators making random mistakes to annotators approving systematic ones.

The new job description

If the machine produces the first label, the human contribution moves to four places.

Specification. Somebody decides what the categories are, what counts as an instance, how edge cases resolve. With a language model as annotator, this takes the form of a prompt and a set of examples, but it is the same intellectual work as writing an annotation guideline, and the quality of everything downstream depends on it.

Verification. A sample of machine labels must be checked against human judgement, per category, on an ongoing basis. This is a measurement task. It needs labels produced independently of the machine, by people who have not seen its answer.

Exception handling. The items the model is unsure about, or that fall outside what it has seen, go to people. As in the second stage of the Segment Anything engine, these are by construction the hard cases.

Adjudication. When model and human disagree, or two models disagree, somebody with the authority and knowledge to decide has to decide.

Each of these is more skilled than drawing boxes. An annotator in this system needs to understand the task well enough to notice when a plausible-looking output is wrong, which is a harder cognitive demand than producing an answer unaided.

Designing against anchoring

The central design problem of AI-assisted annotation is that review is cheaper than labelling only if reviewers stay critical, and the evidence says they tend not to. Several countermeasures have support.

Blind a sample. Have a proportion of items labelled from scratch by humans who never see the machine's output. Compare. This is the only clean estimate of the machine's error rate and of how much reviewers are anchoring.

Seed known errors. Insert items where the pre-label is deliberately wrong and measure how often reviewers catch them. A reviewer who approves planted errors is not reviewing.

Show uncertainty, and hide certainty. Route low-confidence items to full human labelling with no suggestion shown. Reserve the review-and-approve mode for items where the model is reliably right.

Rotate the suggestion source. If two different models pre-label, and their disagreements are surfaced to the human, the reviewer is prompted to think instead of confirm.

Watch the override rate. If reviewers change almost nothing, either the model is excellent or the review is nominal. The seeded-error test distinguishes the two.

Does the human share go to zero?

The strongest form of the automation argument is that these are transitional problems. Models are improving quickly. If a model can label at expert level, review becomes unnecessary.

For some tasks that is already close to true, and for those tasks human annotation at volume should stop. The Segment Anything result is a case where a model, bootstrapped by people, reached a point where its automatic masks were judged to be of high quality in the authors' own review.

But notice what cannot be automated by construction. A model cannot be the independent check on itself: verification requires a signal from outside. A model cannot define what the organisation wants the categories to be. And the cases a model handles worst are, by definition, the cases where its labels are least trustworthy and a human's are most needed.

There is also a moving frontier. Each time models master a class of labels, the valuable data becomes whatever lies just beyond: rarer cases, finer distinctions, more specialised domains, judgements about quality instead of category. The second stage of the data engine is the general pattern.

What this changes

FOR AI BUILDERS

For any model-assisted pipeline, maintain a blind, human-only labelled sample as the reference. Without it you have no error estimate.

Measure reviewer performance with seeded errors, not with agreement with the pre-label.

Route by confidence: automatic acceptance where the model is verified reliable, unassisted human labelling where it is not.

FOR ENTERPRISES

Expect AI assistance to lower unit cost and to raise the skill required of the remaining human workforce. Price and staffing should reflect both.

Ask suppliers what share of labels are machine-generated, how they are verified, and what measures counter automation bias.

Validate LLM-based labelling on your own task before relying on it. Published results on other tasks do not transfer.

THE INDIKA PERSPECTIVE

The useful description of modern annotation is a control system: a model that proposes, people who specify and verify, and instrumentation that shows whether the people are still exercising judgement. The labour content per label falls. The design content rises. An operation that still measures itself in items per hour is measuring the part of the work that machines have taken over.

RESEARCH & REFERENCES

- Kirillov, A. et al. (2023). "Segment Anything." IEEE/CVF ICCV.
- Gilardi, F., Alizadeh, M. and Kubli, M. (2023). "ChatGPT outperforms crowd workers for text-annotation tasks." Proceedings of the National Academy of Sciences.
- Ratner, A. et al. (2017). "Snorkel: Rapid Training Data Creation with Weak Supervision." Proceedings of the VLDB Endowment.
- Bach, S. et al. (2019). "Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale." ACM SIGMOD.
- Pangakis, N., Wolken, S. and Fasching, N. (2023). "Automated Annotation with Generative AI Requires Validation." arXiv preprint.
- Ziems, C. et al. (2024). "Can Large Language Models Transform Computational Social Science?" Computational Linguistics.
- Tan, Z. et al. (2024). "Large Language Models for Data Annotation and Synthesis: A Survey." EMNLP.
- Fort, K. and Sagot, B. (2010). "Influence of Pre-annotation on POS-tagged Corpus Development." ACL Linguistic Annotation Workshop.
- Parasuraman, R. and Manzey, D. (2010). "Complacency and Bias in Human Use of Automation: An Attentional Integration." Human Factors.
- Veselovsky, V., Ribeiro, M. H. and West, R. (2023). "Artificial Artificial Intelligence: Crowd Workers Widely Use Large Language Models for Text Production Tasks." arXiv preprint.

Stop Labeling Everything. Carefully.

Active learning promises the same model for a fraction of the labels. The research says the promise is real, conditional, and easier to break than most introductions admit.

7 MINUTE READ

Suppose you have a million unlabelled images and a budget to label ten thousand. Which ten thousand?

The default answer is a random sample. It is simple and unbiased, and it wastes most of the budget. If 95% of the images are easy cases that a model would get right after seeing a few hundred, then roughly 9,500 of the labels will teach it almost nothing.

Active learning is the alternative. Train a model on a small labelled seed. Use it to score the unlabelled pool. Send the most informative examples to annotators. Retrain, and repeat. The idea goes back to the early 1990s, and Burr Settles's 2009 survey remains the standard reference for its many variants.

The core intuition is sound and the published gains can be large. It is also among the techniques with the widest gap between how often it is recommended and how reliably it works. That gap is worth understanding before building a labelling budget around it.

How the selection works

Most methods fall into two families.

Uncertainty sampling picks examples the current model is least sure about: those nearest its decision boundary, or where its predicted probabilities are most evenly spread. The logic is that a confident prediction is probably right and a label would confirm what the model already knows.

Diversity sampling picks examples that are unlike those already labelled, to cover the input space. The core-set approach of Sener and Savarese is the best-known formulation: choose points such that every unlabelled example is close to some labelled one.

Practical systems combine the two, selecting batches that are both uncertain and varied, since the hundred most uncertain examples are often a hundred near-copies of the same hard case.

FIGURE 11 · FRAMEWORK

When model-guided selection tends to pay off

MORE LIKELY TO HELP	LESS LIKELY TO HELP
The cases that matter are rare in the raw data	Classes are balanced and most examples are informative
The unlabelled pool is very large relative to the budget	The pool is small enough to label a representative sample
Labels are expensive, so each wasted one matters	Labels are cheap and fast
Uncertain items are checked for being answerable before labelling	The top of the uncertainty ranking is sent out unfiltered
A random stream is labelled alongside for evaluation and comparison	All labelled data, including the test set, is model-selected
Selection uses several models or a general-purpose encoder	Selection depends on one task model that will soon be replaced

Synthesis by the editors, after Lowell et al. (2019), Karamcheti et al. (2021) and Munjal et al. (2022).

Where it has worked

The clearest successes are in settings with three features: a large unlabelled pool, a heavily skewed distribution, and expensive labels.

Autonomous driving is the standard example. Fleets record far more footage than could ever be annotated, nearly all of it uneventful. Companies in the field have described mining that footage for the scenes their models handle worst, using model disagreement and uncertainty as triggers, and labelling those. The Segment Anything data engine described in the previous article applies similar logic: annotators were directed to the objects the model had missed.

Rare-class problems benefit for the same reason. If one transaction in a thousand is fraudulent, a random sample of ten thousand contains about ten positives. A model-guided sample can contain hundreds.

In these cases, the alternative to active selection is not a slightly less efficient random sample. It is a dataset that barely contains the phenomenon of interest.

Where it has not

In 2019, Lowell, Lipton and Wallace published a paper with a deliberately plain title: “Practical Obstacles to Deploying Active Learning”. They ran standard methods across several text classification and sequence labelling tasks, with several model types, and reported three findings that anyone planning to use the technique should know.

Gains were inconsistent. Active learning beat random sampling in some combinations of dataset and model and lost in others. There was no way to know in advance which would occur, and a practitioner cannot run both to find out, because the point is to label only once.

The data is tied to the model that chose it. A dataset selected by one model was often no better than random, and sometimes worse, when used to train a different model. This matters because the labelled data usually outlives the model. Teams change architectures every year or two. A dataset shaped by the uncertainties of the previous one carries its blind spots forward.

The sample is no longer representative. An actively selected set over-represents hard and unusual cases. It cannot be used to estimate real-world performance, class frequencies or anything else that depends on the natural distribution.

Other studies add to the list. Karamcheti and colleagues investigated why active learning underperformed on visual question answering and traced it to what they called collective outliers: groups of examples that the model is highly uncertain about because they are unanswerable, mislabelled or require knowledge the model cannot acquire. Uncertainty sampling sought these out avidly. The budget went to examples that could not be learned from.

Munjal and colleagues re-ran a range of deep active-learning methods under controlled conditions and found that many reported improvements over random sampling shrank or vanished once the random baseline was properly tuned and regularised. Part of the published advantage had been an artefact of weak baselines.

There is also a cold-start problem. In the first rounds, the model has seen too little to have meaningful uncertainty. Its selections are close to noise, and can be worse than random because they are biased noise.

KEY TAKEAWAY

Uncertainty sampling cannot distinguish “hard but learnable” from “hard because the label is wrong or the question is unanswerable”. Without a filter, it spends the budget on the second kind.

Reconciling the two pictures

How can the same technique be indispensable in one setting and unreliable in another?

The successes share a property. In driving and fraud, the natural distribution is so skewed that almost any intelligent selection beats random. The gain comes from finding rare things, and it survives a change of model because the rare things remain rare and relevant whatever the architecture.

The failures cluster in balanced academic benchmarks, where most examples carry some information and random sampling is already a reasonable strategy. There, the gain depends on fine distinctions about which examples are marginally more useful to a particular model, and those distinctions do not transfer.

The practical reading is that active learning is most trustworthy when it is used as a search tool for under-represented cases, and least trustworthy when it is used as a fine optimiser of labelling efficiency on data that is already reasonably balanced.

Using it without getting hurt

A few design choices address most of the documented failure modes.

Always keep a random stream. Label a random sample alongside the selected one, perhaps a fifth to a third of the budget. It provides an unbiased evaluation set, an estimate of the true distribution, a guard against the selector's blind spots, and a baseline that shows whether selection is helping.

Never evaluate on selected data. The test set must come from the natural distribution, or from the deployment environment directly. A model evaluated on its own hardest cases will look worse than it is. One evaluated on a set chosen by a previous model will look arbitrary.

Filter before you send. Examples flagged as most uncertain should pass a quick check for being answerable and in scope. Route the remainder to a separate queue: some will be labelling errors or guideline gaps, which are worth knowing about for other reasons.

Combine uncertainty with diversity. Batches selected purely on uncertainty are redundant. Clustering candidates and sampling across clusters is a simple and effective correction.

Start with a random or diverse seed. Do not use model-guided selection until the model is good enough for its uncertainty to mean something.

Select with an ensemble or a strong general model. Disagreement among several different models is a more architecture-neutral signal than one model's confidence. Using a large pretrained model's embeddings for diversity selection

also reduces dependence on the task model. Coleman and colleagues showed that a small, cheap proxy model can do the selecting for a larger one with little loss, which helps with compute cost though not with the transfer problem in general.

Use production as the selector. The most reliable source of informative examples is failure in deployment: user corrections, low-confidence predictions on real inputs, disagreements between the model and a human reviewer. This is active learning in the sense that matters, and it selects from the distribution the model will keep facing.

The economic argument, stated properly

Active learning is usually sold on cost reduction: the same accuracy with fewer labels. Given the evidence, a more defensible claim is about allocation. A fixed budget buys more useful information when part of it is directed.

That reframing matters for how the technique interacts with the rest of an annotation operation. Selected examples are harder, so they take longer and produce more disagreement. An operation priced per item, staffed for average difficulty, will see throughput fall and error rates rise when active learning is switched on, and may conclude the method has failed when it is working as designed. The hard items are where expert annotators and adjudication, discussed in earlier articles, should be concentrated.

What this changes

FOR AI BUILDERS

Split the labelling budget between a model-selected stream and a random stream, and compare models trained on each.

Inspect the top of the uncertainty ranking by hand before sending it out. If it is full of garbage, add a filter.

Record how each example was selected. Future users of the dataset need to know it is not a random sample.

FOR ENTERPRISES

Be wary of quoted label savings from benchmark studies. Expect large gains where your important cases are rare, and modest or uncertain gains elsewhere.

Plan for selected data to be slower and costlier per item. The saving is in the total, not the unit price.

Make sure the evaluation set is collected independently of any model-guided selection.

THE INDIKA PERSPECTIVE

Deciding what to label is at least as valuable as labelling it well, and it is a joint decision between the model team and the data team. The data side contributes things the model cannot see: which uncertain items are unanswerable, which reflect a guideline gap, which need an expert. An operation that simply labels whatever a selection algorithm sends is leaving out the half of the loop where human judgement improves the selection.

RESEARCH & REFERENCES

- Settles, B. (2009). "Active Learning Literature Survey." Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Lewis, D. and Gale, W. (1994). "A Sequential Algorithm for Training Text Classifiers." ACM SIGIR.
- Sener, O. and Savarese, S. (2018). "Active Learning for Convolutional Neural Networks: A Core-Set Approach." ICLR.
- Lowell, D., Lipton, Z. and Wallace, B. (2019). "Practical Obstacles to Deploying Active Learning." EMNLP.
- Karamcheti, S. et al. (2021). "Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering." ACL.
- Munjal, P. et al. (2022). "Towards Robust and Reproducible Active Learning Using Neural Networks." IEEE/CVF CVPR.
- Coleman, C. et al. (2020). "Selection via Proxy: Efficient Data Selection for Deep Learning." ICLR.
- Siddhant, A. and Lipton, Z. (2018). "Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study." EMNLP.
- Attenberg, J. and Provost, F. (2011). "Inactive Learning? Difficulties Employing Active Learning in Practice." ACM SIGKDD Explorations.
- Kirillov, A. et al. (2023). "Segment Anything." IEEE/CVF ICCV.

What the Human in the Loop Is For

Adding a person to an AI system is assumed to make it safer and better. A meta-analysis of more than a hundred experiments found that, on average, it made things worse than the better of the two alone.

7 MINUTE READ

“Human in the loop” is one of the most reassuring phrases in applied AI. It appears in governance frameworks, procurement requirements and product pages. The assumption behind it is rarely stated because it seems too obvious to need stating: a person plus a model will do at least as well as either alone.

In 2024, three researchers at MIT tested that assumption against the published evidence. Vaccaro, Almaatouq and Malone conducted a preregistered meta-analysis of 106 experimental studies, covering 370 comparisons, in which the performance of humans alone, AI alone and humans working with AI had all been measured on the same task.

On average, the human-AI combination performed significantly worse than the better of the human or the AI working alone.

The effect was small, and it was an average over very different tasks. The pattern within it is what makes the study useful.

Two findings that explain the average

The direction depended on who was better to begin with. When humans alone outperformed the AI alone, adding the AI helped: the combination beat the human. When the AI alone outperformed humans, adding a human hurt: the combination was worse than the AI.

That asymmetry makes sense on reflection. A person who is better than the machine can use its output as one more input and overrule it when it is wrong. A person who is worse than the machine cannot reliably tell when it is wrong, and their interventions replace some correct machine answers with incorrect human ones.

The direction depended on the kind of task. Combinations showed losses on decision tasks, such as classifying, diagnosing or forecasting. They showed more favourable results on tasks that involved creating content.

FIGURE 12 · DATA VISUALISATION

Human plus AI, compared with the better of the two alone



Effect sizes (Hedges' g) for the human-AI combination relative to the better of human or AI alone. Source: Vaccaro, Almaatouq and Malone (2024). Where AI alone beat humans alone, the authors report losses for the combination.

The implication for anyone designing oversight is uncomfortable. The setting where a human reviewer is most commonly mandated, a person checking the decisions of a model that is on average more accurate than they are, is the setting where the evidence for benefit is weakest.

Why review fails

The mechanisms have been studied for decades, mostly in aviation and industrial control before AI.

Lisanne Bainbridge set out the core problem in 1983 in a paper called “Ironies of Automation”. The more reliable an automated system, the less its human supervisor has to do, and the less practised and attentive they become. The supervisor is then expected to step in precisely when the automation has failed, which is the hardest situation, with the least preparation.

Parasuraman and Manzey’s review of the subsequent research distinguishes two failures. **Complacency** is insufficient monitoring: the human stops checking. **Automation bias** is deference: the human sees the system’s recommendation and follows it even when other information contradicts it. Both occur in experts as well as novices, and neither is reliably prevented by training or by telling people to be vigilant.

The effect has been observed with current AI systems. In a field experiment with several hundred consultants at Boston Consulting Group, Dell’Acqua and colleagues gave participants access to a large language model. On tasks within the model’s competence, those with access were faster and produced higher-quality

work. On a task chosen to lie just outside its competence, those with access were 19 percentage points less likely to reach the correct answer than those without. They had trusted an output that looked as good as the others.

The authors called the boundary a jagged frontier. From the user's side, tasks the model does well and tasks it does badly are hard to tell apart, because the output is equally fluent in both.

A further finding, from Bansal and colleagues, closes off an obvious remedy. Giving people the model's explanation along with its answer increased their tendency to accept the answer, whether it was right or wrong. Explanations raised trust. They did not improve discrimination.

KEY TAKEAWAY

A reviewer adds value only on the cases where they would have decided differently from the model and been right. If you cannot say which cases those are, you have a signature, not a safeguard.

What humans contribute

None of this argues for removing people. It argues for being specific about what they are there to do. Across the research, human contribution is strongest in five roles.

Defining the task. What counts as a good answer, which errors are tolerable and which are not, what the categories mean. A model cannot supply its own objective. This is the role that the articles on annotation guidelines and preference data in this series return to repeatedly.

Supplying information the model does not have. Context from outside the input: the patient mentioned something in the corridor, the supplier changed last week, this customer has complained before. Where the human sees more than the model, the Vaccaro pattern predicts gains.

Handling novelty. Inputs unlike anything in the training data, where the model's confidence is least meaningful. This requires that the system can recognise such inputs and route them, which is a technical design problem in its own right.

Generating ground truth. Independent human judgements, made without seeing the model's output, are what make it possible to measure the model at all. This is a different activity from reviewing its decisions, and it is immune to automation bias by construction.

Carrying accountability. Some decisions require a person who can be answerable for them. That is a legitimate institutional requirement. It should not be mistaken for a quality control.

The regulatory expectation

Regulation has begun to make these distinctions. Article 14 of the European Union's AI Act requires that high-risk systems be designed so that people can oversee them effectively. It goes further than requiring presence. It says the people assigned must be able to understand the system's capacities and limitations, to remain aware of the tendency to over-rely on its output, which it names as automation bias, and to decide not to use the output or to override it.

That wording sets a functional test. A human who is nominally in the loop but lacks the time, information or authority to disagree does not satisfy it.

Designing a loop that works

The practical consequences follow from asking, for each point where a human is placed, what they know that the model does not.

Route, do not blanket. Reviewing every output spreads attention thinly and trains reviewers to approve. Reviewing the subset where the model is uncertain, the input is unusual or the stakes are high concentrates attention where it can change outcomes.

Decide before seeing. For high-stakes decisions, have the human form a judgement first and compare with the model's afterwards. This removes the anchor. It costs time, which is why it should be reserved for the routed subset.

Measure the reviewer. Insert cases where the model's output is known to be wrong and track whether reviewers catch them. Track override rates and whether overrides were correct. A reviewer whose overrides are wrong as often as right is adding noise.

Give reviewers something the model lacks. Additional context, more time, access to the source, the ability to ask a question. If the reviewer sees exactly what the model saw, and is less accurate than the model, the evidence predicts a loss.

Turn review into data. Every correct override is a training example and an evaluation case. A loop that discards them wastes its most valuable output. A loop that captures them is a continuous data pipeline.

Sample for audit separately. Independently of live review, have a random sample of model decisions assessed blind by qualified people. This measures the system. Live review does not, because it is contaminated by the model's suggestion.

The objection from necessity

A practitioner might reply that this is a counsel of perfection. Regulators and customers demand a human reviewer. Removing one because a meta-analysis says the average effect is negative would be reckless.

That is correct, and it is not what the evidence suggests doing. The studies show that undifferentiated human review of a stronger system does not reliably help. They also show clear benefit where humans are the stronger party. The work is to move human effort from the first situation to the second: fewer cases, better chosen, with more information and real authority.

There is a second reply. Model accuracy on average says little about accuracy on a given case. A model that is better than a clinician across ten thousand scans may be much worse on a particular rare presentation. Human review targeted at such cases can add value even when the model is better overall. That is an argument for precise routing, and it depends on knowing where the model is weak, which is an evaluation-data problem.

What this changes

FOR AI BUILDERS

For every human checkpoint, write down what the human knows or can do that the model cannot. If the answer is nothing, redesign the checkpoint.

Instrument review: seeded errors, override rates, override accuracy. Report them alongside model metrics.

Capture overrides as labelled data with the reviewer's reasoning. Feed them into both training and evaluation sets.

FOR ENTERPRISES

Treat "human in the loop" as a design specification to be tested, not a compliance statement to be asserted.

Staff review roles with people who have the expertise and time to disagree with the system, and the authority to act on it.

Separate live review from independent audit. Only the second tells you how well the system is performing.

THE INDIKA PERSPECTIVE

Human-in-the-loop services are often sold as coverage: a person looks at everything. The research points to a different offer. Identify where the model is weak, route those cases to people equipped to decide them, verify that those people are exercising independent judgement, and return every correction as data. That is a smaller loop with more engineering in it, and it is the version for which there is evidence.

RESEARCH & REFERENCES

Vaccaro, M., Almaatouq, A. and Malone, T. (2024). "When combinations of humans and AI are useful: A systematic review and meta-analysis." *Nature Human Behaviour*.

Bainbridge, L. (1983). "Ironies of Automation." *Automatica*.

Parasuraman, R. and Manzey, D. (2010). "Complacency and Bias in Human Use of Automation: An Attentional Integration." *Human Factors*.

Dell'Acqua, F. et al. (2023). "Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality." *Harvard Business School Working Paper 24-013*.

Bansal, G. et al. (2021). "Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance." *ACM CHI*.

Goddard, K., Roudsari, A. and Wyatt, J. (2012). "Automation bias: a systematic review of frequency, effect mediators, and mitigators." *Journal of the American Medical Informatics Association*.

Green, B. and Chen, Y. (2019). "Disparate Interactions: An Algorithm-in-the-Loop Analysis of Fairness in Risk Assessments." *ACM FAT**.

Amershi, S. et al. (2014). "Power to the People: The Role of Humans in Interactive Machine Learning." *AI Magazine*.

European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act), Article 14: Human oversight.

RLHF Is a Data Problem

The technique that turned language models into assistants is usually explained as an algorithm. Most of what determines its outcome happens before any algorithm runs.

7 MINUTE READ

The paper that set out the method behind ChatGPT is long, most of it appendices. The reinforcement-learning algorithm at its centre is described in a few paragraphs. Much of the rest is about people: how they were recruited, what they were told, how they were tested, how often they agreed, and what was done when they did not.

That allocation of pages is a fair guide to where the difficulty lies.

The pipeline, as a sequence of datasets

Reinforcement learning from human feedback, or RLHF, is normally drawn as three training steps. It is more revealing to describe it as three datasets, each made by people.

FIGURE 13 · DATA VISUALISATION

RLHF as three datasets

DATASET	MADE BY	USED TO	PROMPTS
Demonstrations	Annotators write the ideal response	Fine-tune the base model	about 13,000
Comparisons	Annotators rank several model responses	Train a reward model to predict human preference	about 33,000
Optimisation prompts	No human labels	Tune the model against the reward model	about 31,000

Source: Ouyang et al. (2022), "Training language models to follow instructions with human feedback". About 40 contractors produced the human-made data.

Demonstrations. Annotators are given prompts and write the response they think a good assistant should give. The pretrained model is fine-tuned on these. In the InstructGPT work, this dataset contained about 13,000 prompts.

Comparisons. The model generates several responses to a prompt. Annotators rank them from best to worst. A separate model, the reward model, is trained to predict which response a human would prefer. This dataset contained about 33,000 prompts.

Prompts for optimisation. The language model is then tuned to produce responses the reward model scores highly, using a further 31,000 prompts with no human labels.

The reinforcement learning happens in the last step, and it optimises against a learned imitation of human preference. Everything the final model knows about what people want came through the first two datasets. The idea of learning a reward function from human comparisons instead of specifying it by hand predates language models: Christiano and colleagues showed in 2017 that agents could learn complex behaviours from a modest number of human preference judgements.

The decisions that are not in the algorithm

Read as an annotation project, the InstructGPT paper is a list of design choices, each of which would be familiar to anyone who has run a labelling operation.

Who labels. The team hired about 40 contractors through freelance and data-labelling platforms. Candidates took a screening test measuring, among other things, how well they could identify sensitive content and how closely their rankings agreed with the researchers' own. Those who scored well were selected. The model's notion of a good answer is therefore, to a first approximation, the notion held by forty people chosen for agreeing with the researchers.

What they are told. The instructions asked annotators to prioritise helpfulness to the user, then truthfulness, then harmlessness, with guidance on handling ambiguity. Those instructions changed during the project. Each revision altered what the data meant.

How much they agree. The paper reports that annotators involved in producing training data agreed with one another about 73% of the time on comparisons. Roughly a quarter of the time, two selected, trained people looking at the same pair of outputs preferred different ones. The reward model is fitted through that disagreement as though it were noise around a single preference.

What they see. The prompts came largely from usage of an earlier product and from annotators' own inventions. The model was tuned for the distribution of requests those sources happened to contain.

The authors are candid about this. They write that the model's behaviour is determined in part by the contractors' judgements, their instructions and the researchers' own choices, and that these do not represent everyone who will use or be affected by the model.

The headline result shows what such data can do. Human evaluators preferred outputs from a 1.3-billion-parameter model trained this way to outputs from the 175-billion-parameter base model. A modest quantity of preference data did what two orders of magnitude of scale had not.

Three ways the data fails

Since the method optimises toward whatever the annotators rewarded, its failures are annotation failures.

Annotators reward what looks right. Hosking, Blunsom and Bartolo asked whether human preference scores capture the properties people assume they capture. They found that judgements of factual accuracy were swayed by how assertively a response was written. Confident wrong answers were less likely to be marked as containing errors. A model optimised on such ratings learns to be assertive.

Annotators reward agreement. Researchers at Anthropic studied sycophancy, the tendency of assistants to tell users what they appear to want to hear. Across several production models they found consistent patterns: changing a correct answer when the user pushed back, or tailoring feedback to the user's stated view. Analysing preference data, they found that responses matching a user's beliefs were more likely to be preferred. The behaviour was learned from the signal.

Annotators cannot evaluate what they cannot check. A rater comparing two explanations of a tax rule, or two code snippets, in a minute or two, without relevant expertise, will prefer the one that is clearer and better organised. Whether it is correct is often beyond what the task allows them to determine.

Casper and colleagues, in a survey of open problems with the method, organise these under three headings: problems with the human feedback, problems with the reward model, and problems with the policy. The first heading includes annotator error, limited attention, the difficulty of evaluating hard tasks, and the fact that a single reward function cannot represent the differing preferences of different people.

KEY TAKEAWAY

A reward model is a compressed copy of a group of annotators. Optimising against it produces a system that is very good at pleasing that group, under the conditions in which they were asked.

Is the data problem being engineered away?

There are three serious arguments that the importance of human preference data is declining.

The first is that newer methods need less of it. Direct preference optimisation, introduced by Rafailov and colleagues in 2023, removes the separate reward model and the reinforcement-learning loop, and trains on preference pairs directly. It is simpler and more stable. It still requires the pairs. The algorithm became easier; the data requirement stayed.

The second is that AI feedback can replace human feedback. Constitutional AI, developed at Anthropic, uses a model to critique and revise its own outputs against a written set of principles, and to generate the preference labels. Researchers at Google have reported that on tasks such as summarisation, feedback from a strong model performed comparably to human feedback. A later article in this series examines that evidence. The short version is that the approach works where a strong model's judgement is already reliable, and that the written principles, and the validation that the judge agrees with people, are themselves human-made data.

The third is that reinforcement learning with automatically checkable rewards, such as whether code passes its tests or a proof verifies, avoids human judgement altogether. Where such rewards exist, this is true, and it is where some of the fastest recent progress has come from. Most of what people want from an assistant has no automatic checker.

In each case, human data is concentrated, not eliminated. Less of it is needed. What is needed is harder to produce and has more influence per item.

What follows for practice

If RLHF is an annotation pipeline, the standards of good annotation apply, with higher stakes.

Specify the rater pool deliberately. The reward model will encode their preferences. For a medical assistant, raters need clinical knowledge. For a product used across India, they need to be drawn from across India. Convenience sampling produces a model aligned with whoever was convenient.

Give raters the means to check. Time, reference material, the ability to run the code. Where correctness matters, a preference collected without verification is a preference for plausibility.

Separate the dimensions. A single overall ranking blends accuracy, tone, completeness and safety into one number, and raters will weight them inconsistently. Asking about each dimension separately costs more and yields a signal that can be audited.

Measure and keep disagreement. As earlier articles argued, a quarter of comparisons being contested is information. Some of it reflects items with no better answer. Some reflects different legitimate preferences among users.

Audit for the known failure modes. Seed comparisons in which the more assertive answer is wrong, or the more agreeable answer is wrong, and measure how raters respond.

Version the guidelines with the data. A comparison is interpretable only alongside the instructions under which it was made.

What this changes

FOR AI BUILDERS

Treat the rater guideline as the specification of model behaviour, because functionally it is. Review it with the same care as a system prompt or a loss function.

Test raters on items with verifiable answers.
Preference without accuracy checking drifts toward fluent error.

Log which raters, guideline version and prompt source produced each comparison. Without that, a behavioural regression cannot be traced.

FOR ENTERPRISES

When fine-tuning with feedback, the raters should resemble your users or your experts, not a general crowd.

Ask any provider of preference data who the raters were, how they were selected and how they were instructed.

Budget for expert raters where correctness cannot be judged by a layperson. That is where preference data most often goes wrong.

THE INDIKA PERSPECTIVE

The quality of a preference dataset is set by decisions that look administrative: recruitment criteria, instructions, time allowed per judgement, what tools raters have. Those decisions are where a model's values and blind spots originate. A data partner working on post-training should expect to be questioned about each one, and should be able to show, with evidence, how its raters behave on the cases designed to mislead them.

RESEARCH & REFERENCES

Ouyang, L. et al. (2022). "Training language models to follow instructions with human feedback." NeurIPS.

Christiano, P. et al. (2017). "Deep Reinforcement Learning from Human Preferences." NeurIPS.

Stiennon, N. et al. (2020). "Learning to summarize from human feedback." NeurIPS.

Bai, Y. et al. (2022). "Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback." Anthropic, arXiv preprint.

Hosking, T., Blunsom, P. and Bartolo, M. (2024). "Human Feedback is not Gold Standard." ICLR.

Sharma, M. et al. (2024). "Towards Understanding Sycophancy in Language Models." ICLR.

Casper, S. et al. (2023). "Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback." Transactions on Machine Learning Research.

Rafailov, R. et al. (2023). "Direct Preference Optimization: Your Language Model is Secretly a Reward Model." NeurIPS.

Bai, Y. et al. (2022). "Constitutional AI: Harmlessness from AI Feedback." Anthropic, arXiv preprint.

Lee, H. et al. (2024). "RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback." ICML.

Whose Preferences?

Preference data has replaced labels as the scarce input for generative AI. Unlike a label, a preference always belongs to someone.

7 MINUTE READ

Ask ten people whether a photograph contains a dog and, barring odd cases, you will get one answer. Ask ten people which of two essays is better and you will get a vote.

That difference is easy to overlook, because the two kinds of data are stored the same way: an item, an identifier, a choice. They are collected through similar interfaces by similar workforces. But they are different kinds of object. The first is an attempt to measure a fact. The second is a record of what a particular person, in a particular situation, liked.

Generative AI runs increasingly on the second kind. As the previous article described, post-training methods learn from comparisons between outputs. Public leaderboards rank models by which one users preferred. Product teams tune on thumbs-up and thumbs-down. The question of whose preferences these are has moved from a footnote to a design decision.

What a preference is evidence of

A classification label has an implied referent outside the annotator. If two annotators disagree about whether an image shows a dog, one of them is probably wrong.

A preference has no such referent. If one rater prefers a concise answer and another prefers a thorough one, neither is mistaken. They want different things. Aggregating them produces an answer to a different question: what does this group, weighted this way, prefer on average?

FIGURE 14 · COMPARISON TABLE

A label and a preference are different objects

	CLASSIFICATION LABEL	PREFERENCE
What it claims	Something about the item	Something about the person judging
When two people differ	One is probably wrong	Both may be right
Aggregation yields	An estimate of the truth	The average taste of whoever was asked
Essential metadata	Guideline version, annotator skill	Who the raters were, and in what context they judged
Main quality risk	Error	An unrepresentative panel; plausibility mistaken for accuracy
Closest discipline	Measurement	Survey research

Synthesis by the editors.

Three properties follow, and each has consequences for how the data should be collected.

Preferences depend on the person. The PRISM dataset, built by Kirk and colleagues, collected the preferences of 1,500 participants from 75 countries across more than 8,000 live conversations with 21 language models, and linked each rating to the participant's background. The authors found that which models were rated best depended on who was in the sample. Different subgroups, defined by demographics or geography, produced different rankings. There was no single ordering that a different choice of raters would have left unchanged.

Preferences depend on the context. The same person wants brevity when checking a fact and depth when learning a subject. A preference stripped of the task it was expressed in is hard to interpret.

Preferences can be mistaken about their own object. A rater may prefer the answer that sounds authoritative and be wrong about which answer is accurate. As the previous article noted, work by Hosking and colleagues found that assertiveness shifted human judgements of factual accuracy. This is where preference and fact become entangled: a rater's preference for answer A is partly a belief that A is correct, and that belief can be false.

Where the preferences have come from so far

It is instructive to ask who has supplied the preference data that current systems learned from.

For early post-training work, the answer was small groups of contractors. InstructGPT used about forty, selected in part for agreement with the research team.

For public leaderboards, it is whoever turns up. Chatbot Arena, the most influential of these, lets visitors compare two anonymous models and vote. Its founding paper reported more than 240,000 votes. This is real usage by real people and far more varied than a contractor pool. It is also a self-selected population of people who visit model-comparison websites, asking the kinds of questions such people ask. A 2025 critique titled “The Leaderboard Illusion” raised further concerns, including that some developers were able to test many private model variants and publish only the best, and that access to arena data was unevenly distributed, which could let models be tuned to the arena’s particular distribution.

For product feedback, it is existing users, with all the selection that implies. People who dislike a product leave and stop providing feedback.

None of these is illegitimate. Each is a sample from some population, and none is a sample from “humanity” or from any particular enterprise’s customers.

Research on model outputs is consistent with this. Santurkar and colleagues compared the opinions expressed by language models with survey responses from different demographic groups in the United States and found that models aligned more closely with some groups than others, with the pattern shifting between model generations as training procedures changed. Durmus and colleagues at Anthropic did the same across countries using global survey questions and found that default responses were closer to the opinions of some populations, largely Western ones, than others.

KEY TAKEAWAY

There is no view from nowhere in a preference dataset. The choice is between specifying whose view it represents and leaving that to whoever was easiest to recruit.

The case for an average

The objection to all this is practical. A product ships one default behaviour. It has to be tuned to something. A broad average of reasonable people’s preferences is a sensible target, and worrying about whose preferences are represented is a luxury.

There is something to this. For many properties, such as coherence, relevance and following the instructions given, preferences converge across almost everyone. A response that ignores the question is bad for all raters.

But for a large class of properties, averaging produces a result nobody wanted. If half of users want formal language and half want casual, the mean satisfies neither. If preferences about directness differ between cultures, a model tuned to the pooled average is slightly wrong everywhere. And averages silently weight by headcount: a minority of users with distinct needs, whether speakers of a less common language or practitioners in a specialised field, contribute little to the mean and are served badly by it.

A second practical objection is that preference data is expensive, and specifying the rater population makes it more so. That is true. The alternative is to pay less for a signal whose meaning is unknown.

Three ways to handle plurality

Researchers have begun to describe what it would mean for a model to accommodate differing preferences, a programme sometimes called pluralistic alignment. Sorensen and colleagues distinguish several targets, which map onto practical choices.

Represent the spread. Where views differ on a contested question, the model presents the range instead of selecting one. This suits public-facing systems dealing with questions of value.

Make it steerable. The model can be directed toward a given style or standard through instructions or configuration. The preference data then needs to cover the range of styles, with each example tagged by which standard it reflects.

Match a defined population. The model's default is tuned so that its behaviour matches the distribution of preferences in a specified group. This is what an enterprise usually wants: the model should prefer what our customers, or our underwriters, or our clinicians prefer.

There have also been experiments in sourcing the principles themselves from a deliberately chosen public. Anthropic and the Collective Intelligence Project ran a process in which roughly a thousand Americans contributed to and voted on principles for model behaviour, and a model was then trained against the resulting document. The exercise was small and exploratory, and it made the sampling question explicit.

Collecting preference data properly

For a team commissioning or building a preference dataset, the implications are concrete.

Define the population first. State whose preferences the model should reflect. Recruit raters to match, and record the composition of the pool.

Keep rater identity attached. Store a persistent, anonymised rater identifier and relevant attributes with every judgement. This allows analysis by subgroup, detection of individual raters with unusual patterns, and re-weighting later.

Separate fact from taste. Ask about correctness separately from style, and where correctness matters, have it assessed by someone equipped to check. A single “which is better” question lets a preference for confident prose masquerade as a judgement of accuracy.

Collect reasons. A short justification with each preference costs time and repays it. It shows what raters were attending to, exposes misreadings of the task, and can be used directly in training.

Report the disagreement. For each comparison, how split were the raters? High-agreement comparisons are safe to train on. Split ones need a decision about what the model should do when people differ.

Check the prompt distribution. Preferences are expressed on particular prompts. If the prompts do not resemble what the deployed model will be asked, the preferences will not transfer.

What this changes

FOR AI BUILDERS

Analyse preference data by rater subgroup before training on the aggregate. Find out where the average conceals a split.

Use separate rating dimensions for accuracy, completeness, tone and safety, with qualified raters on the accuracy dimension.

Treat public leaderboard rankings as one population's preferences on one prompt distribution. Validate on your own.

FOR ENTERPRISES

Decide whose judgement defines “good” for your application, and source preference data from those people.

Do not assume a model that ranks highly on general preference leaderboards matches your users' preferences. Test with them.

In multilingual or multi-regional deployments, collect preferences in each market. Preferences about tone and directness do not translate.

THE INDIKA PERSPECTIVE

Building preference data is closer to survey research than to labelling. It needs a sampling frame, a recruited panel that matches it, instruments that separate distinct questions, and reporting that shows the spread as well as the mean. For a country with India's variety of languages, regions and professional contexts, that discipline decides whether a model reflects its users or reflects whoever happened to be rating that week.

RESEARCH & REFERENCES

- Kirk, H. R. et al. (2024). "The PRISM Alignment Dataset: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models." NeurIPS Datasets and Benchmarks.
- Chiang, W.-L. et al. (2024). "Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference." ICML.
- Singh, S. et al. (2025). "The Leaderboard Illusion." arXiv preprint.
- Ouyang, L. et al. (2022). "Training language models to follow instructions with human feedback." NeurIPS.
- Hosking, T., Blunsom, P. and Bartolo, M. (2024). "Human Feedback is not Gold Standard." ICLR.
- Santurkar, S. et al. (2023). "Whose Opinions Do Language Models Reflect?" ICML.
- Durmus, E. et al. (2023). "Towards Measuring the Representation of Subjective Global Opinions in Language Models." Anthropic, arXiv preprint.
- Sorensen, T. et al. (2024). "A Roadmap to Pluralistic Alignment." ICML.
- Kirk, H. R. et al. (2024). "The benefits, risks and bounds of personalizing the alignment of large language models to individuals." Nature Machine Intelligence.
- Anthropic and the Collective Intelligence Project (2023). "Collective Constitutional AI: Aligning a Language Model with Public Input." Research report.
- Rafailov, R. et al. (2023). "Direct Preference Optimization: Your Language Model is Secretly a Reward Model." NeurIPS.

When AI Grades AI

Language models now generate much of the feedback used to train and evaluate other language models. The evidence on when that works is more specific than either its advocates or its critics suggest.

7 MINUTE READ

A team building a customer-support assistant needs to know whether this week's version is better than last week's. There are four thousand test conversations. Having people read each response and score it would take days and cost more than the team's evaluation budget for the quarter.

So they do what almost every team now does. They write a prompt describing what a good response looks like, give it to a strong language model along with each conversation, and ask it for a score. The whole run finishes before lunch.

The practice is called LLM-as-a-judge, and it has become the default way of evaluating generative systems. Its close relative, reinforcement learning from AI feedback or RLAI, uses the same idea in training: a model supplies the preference labels that human raters used to supply.

Both raise an obvious question. If a model is doing the grading, what is the grade worth?

The evidence that it works

The affirmative case rests on several careful studies.

Zheng and colleagues, in the paper that introduced the MT-Bench benchmark, compared GPT-4's preferences between pairs of chatbot responses with those of human experts. Agreement between the model judge and humans exceeded 80%, which was the same level at which the humans agreed with each other. By that measure the model was as good a judge as another person.

On the training side, researchers at Google compared reinforcement learning from AI feedback with the human version on summarisation and dialogue tasks. Policies trained on preferences generated by a language model were rated by humans as comparable to policies trained on human preferences. They reported a similar result even when the judge model was the same size as the model being trained.

Anthropic’s Constitutional AI took the idea furthest. A written list of principles, the constitution, was given to a model, which then critiqued and revised its own responses and chose between alternatives according to those principles. The resulting preferences were used for training with no human labels on harmfulness. The model produced was both less harmful and, by the authors’ measures, no less helpful than one trained with human feedback.

These are substantial results. For a meaningful set of tasks, a strong model’s judgement tracks human judgement well enough to train and evaluate on.

The evidence on how it fails

The same literature documents consistent ways in which model judges go wrong, and they are not random.

FIGURE 15 · FRAMEWORK

Documented biases of model judges, and controls

BIAS	WHAT HAPPENS	CONTROL
Position	The response shown first is favoured	Judge each pair in both orders; treat flips as ties
Verbosity	Longer answers score higher regardless of content	Score against specific criteria; cap or normalise length
Self-preference	A model rates its own or related outputs more highly	Use a judge from a different model family, or a panel
Limited competence	The judge cannot grade what it cannot solve	Supply the reference answer or source; route to experts
Rubric gaps	Criteria are vague or omit a failure type	Have people read outputs regularly and revise the rubric
Task dependence	Agreement with humans varies widely by task	Calibrate against human judgements for each task

After Zheng et al. (2023), Wang et al. (2024), Panickssery et al. (2024), Shankar et al. (2024) and Bavaresco et al. (2025).

Position. Zheng and colleagues found that judges tended to favour whichever response was presented first. Wang and colleagues showed the effect was large enough to be exploited: by changing the order in which two responses appeared, they could make either one win a majority of comparisons.

Verbosity. Longer responses were preferred even when the additional length added nothing.

Self-preference. Judges rated outputs from themselves, or from closely related models, more favourably. Panickssery and colleagues found that language models could recognise their own outputs at better than chance, and that the strength of that recognition correlated with the strength of the bias. An evaluation in which a model family judges itself against competitors is not neutral.

Limited competence. A judge cannot reliably grade what it cannot do. Zheng's study found judges struggled on mathematics and reasoning questions when they could not solve the problem themselves, and were sometimes misled by a wrong answer placed in front of them.

Uneven reliability across tasks. A large study by Bavaresco and colleagues tested model judges against human judgements across twenty language-evaluation datasets. Agreement varied widely by task and by the property being judged. Their conclusion was that model judges should be validated against human judgements for each task before being relied on.

There is also a structural problem with using AI feedback for training. If the judge shares systematic blind spots with the model being trained, which is likely when they come from the same family or the same training data, those blind spots are reinforced. Human raters, whatever their limits, fail in different ways from the model.

KEY TAKEAWAY

An AI judge is a measuring instrument. Like any instrument, it needs calibrating against a trusted standard, and for judgements about quality the trusted standard is qualified human assessment.

Criteria do not write themselves

A less discussed difficulty concerns the judge's instructions.

Shankar and colleagues studied how practitioners build model-based evaluators and observed a phenomenon they called criteria drift. People did not know their evaluation criteria in advance. They discovered and revised them while looking at outputs. Seeing a response they disliked for a reason they had not anticipated, they added a criterion. Seeing the judge apply a criterion too literally, they reworded it.

That means the judge's prompt is a living document, written by people, encoding their evolving sense of quality. It plays the role an annotation guideline plays for human raters, and it has the same failure modes: ambiguity, omitted cases and unstated assumptions. A judge is only as good as its rubric, and writing the rubric is the human-in-the-loop work that automation does not remove.

Constitutional AI makes the same point at a larger scale. The model generates its own feedback, but from a constitution that people wrote. The human contribution was compressed from tens of thousands of comparisons into a page of principles. It was not eliminated, and each word on that page has more influence than any individual comparison had.

A division of labour

The honest reading of the evidence is that neither “AI can replace human feedback” nor “AI cannot grade AI” is right. A more useful statement has three parts.

Model judges are good at scale and consistency. They apply a rubric to ten thousand items without fatigue, at low cost, with reproducible results. For regression testing, where the question is whether a change made things better or worse on a fixed set, that is most of what is needed.

Humans are needed for calibration. A sample of the judge’s decisions has to be compared with decisions by qualified people, per task and per criterion. If agreement is high, the judge can be trusted on that task. If not, the rubric or the judge needs work. This calibration set is a dataset in its own right, to be built with the care this series has argued for throughout: appropriate raters, measured agreement, recorded ambiguity.

Humans are needed where the judge is out of its depth. Specialised domains, novel failure types, questions where correctness requires checking a source. Here the judge’s score should be treated as a filter that selects cases for expert review, not as a verdict.

The practical design is a pyramid. A model judge scores everything. People audit a stratified sample, weighted toward low-confidence and high-stakes cases. Experts adjudicate the disagreements. Results from the upper levels are used to revise the rubric and recalibrate the judge.

Making a judge trustworthy

Several inexpensive controls address the documented biases.

Randomise order and average. Present each pair in both orders. If the verdict flips, record a tie.

Use a different model family as judge. Do not have a model grade its own outputs or those of its siblings when comparing against others.

Ask for a rubric-based score, not an overall impression. Specific criteria with defined levels are applied more consistently than “rate from one to ten”.

Require reasoning before the score. Having the judge explain its assessment first often improves agreement with humans and gives auditors something to read.

Give the judge the reference. For factual tasks, supply the correct answer or the source document. A judge asked to verify against evidence is doing an easier and more reliable task than one asked to know the answer.

Use a panel. Several judges from different families, with disagreements escalated to people, reduce the effect of any single model's quirks.

Re-calibrate when anything changes. A new judge model, a revised rubric, or a shift in the kind of outputs being judged each invalidate the previous calibration.

What automated feedback cannot do

Two limits deserve stating plainly.

A model judge cannot tell you what you should want. Whether a support assistant should be brief or thorough, cautious or direct, is a decision about the product and its users. As the previous article argued, that is a question about whose preferences count.

And a model judge cannot detect a failure nobody thought to describe. Rubrics list known problems. New ones are found by people reading outputs with open attention, which is slow, unscalable and the origin of every criterion a judge is later asked to apply.

What this changes

FOR AI BUILDERS

Before relying on an automated judge, measure its agreement with qualified human raters on a few hundred items from your task, by criterion.

Control for position, length and self-preference explicitly. They are well documented and cheap to mitigate.

Keep reading outputs by hand. Scheduled, unstructured review is how new failure modes enter the rubric.

FOR ENTERPRISES

Ask how any reported evaluation score was produced. If by a model judge, ask how the judge was validated and against whom.

Treat the evaluation rubric as a controlled document owned by the business, since it defines what "good" means.

Fund a standing human calibration set. It is small relative to total evaluation volume and it underwrites everything else.

THE INDIKA PERSPECTIVE

Automated judging does not reduce the need for expert human evaluation. It changes its purpose. People no longer score every output; they produce the reference judgements that tell you whether the automated scores can be believed. That calibration set is small, has to be right, and has to be refreshed whenever the judge, the rubric or the product changes. It is among the highest-value datasets an AI team owns.

RESEARCH & REFERENCES

- Zheng, L. et al. (2023). "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena." NeurIPS Datasets and Benchmarks.
- Lee, H. et al. (2024). "RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback." ICML.
- Bai, Y. et al. (2022). "Constitutional AI: Harmlessness from AI Feedback." Anthropic, arXiv preprint.
- Wang, P. et al. (2024). "Large Language Models are not Fair Evaluators." ACL.
- Panickssery, A., Bowman, S. and Feng, S. (2024). "LLM Evaluators Recognize and Favor Their Own Generations." NeurIPS.
- Bavaresco, A. et al. (2025). "LLMs instead of Human Judges? A Large Scale Empirical Study across 20 NLP Evaluation Tasks." ACL.
- Shankar, S. et al. (2024). "Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences." ACM UIST.
- Liu, Y. et al. (2023). "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment." EMNLP.
- Koo, R. et al. (2024). "Benchmarking Cognitive Biases in Large Language Models as Evaluators." ACL Findings.

Evaluation Has Become Data Generation

Benchmarks used to be built once and used for a decade. They now leak, saturate and mislead within months, which makes evaluating a model a continuous exercise in making new data.

7 MINUTE READ

SWE-bench is one of the most watched benchmarks in AI. It tests whether a model can resolve real issues taken from open-source software projects, and it is graded automatically: the model’s code change either makes the project’s hidden tests pass or it does not. There are no subjective judgements and no human raters. It appears about as objective as an evaluation can be.

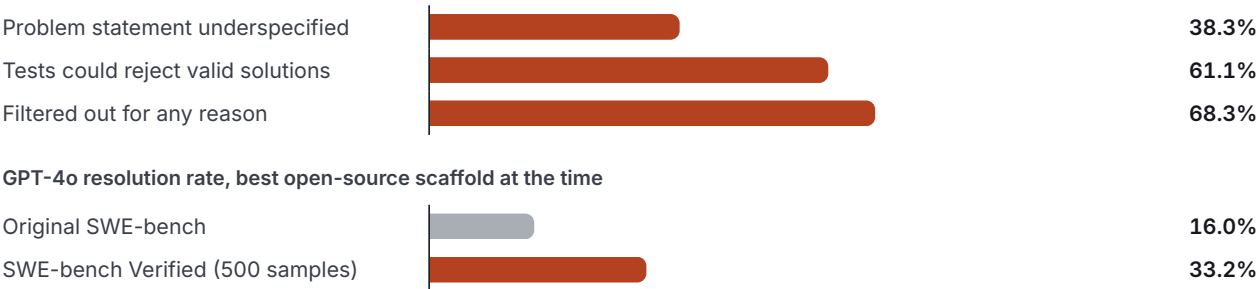
In 2024, OpenAI hired 93 professional software developers to read the benchmark’s problems. They reviewed 1,699 of them. For 38.3%, they judged the problem statement too underspecified for anyone to know what a correct solution would be. For 61.1%, they judged that the tests could reject valid solutions, for instance by requiring a function name the problem never mentioned. In total, 68.3% of the reviewed samples were filtered out.

From what remained, the team assembled a subset of 500 problems that people had confirmed were solvable and fairly graded. On the original benchmark, the best configuration of GPT-4o at the time scored 16%. On the verified subset, it scored 33.2%.

FIGURE 16 · DATA VISUALISATION

What human review found in an automatically graded benchmark

Of 1,699 SWE-bench samples reviewed by 93 software developers



Source: OpenAI (2024), “Introducing SWE-bench Verified”. Categories overlap, so the first two rows do not sum to the third.

The model had not changed. Its measured ability doubled because the measurement was repaired. A benchmark with no humans in it turned out to need ninety-three of them.

That episode contains most of what has changed about evaluation.

Three ways benchmarks stop working

They contain errors. An earlier article in this series described label errors in classic test sets. SWE-bench shows the same problem in a modern agentic benchmark: tasks that are ambiguous, graders that are wrong. An automatic grader is a labelled dataset. Somebody wrote the tests, and they can be mistaken.

They leak. Language models are trained on very large crawls of the public web. Benchmarks are published on the public web. A model may have seen the test questions, and their answers, during training.

The evidence that this affects scores is direct. Researchers at Scale AI wrote more than a thousand new arithmetic word problems matched in style and difficulty to GSM8K, a popular benchmark, and compared model performance on old and new. Several model families scored measurably worse on the fresh problems, with drops of up to 8%, and the size of the drop correlated with signs that a model had memorised the original set. The strongest models showed little decline, which is itself informative: contamination inflates some scores and not others, so it scrambles rankings.

A study from Apple took a different approach, generating variants of the same benchmark's problems with only names and numbers changed. Performance varied across variants of what was logically the same question, and fell when an irrelevant clause was added. Oren and colleagues developed a statistical test that can detect whether a model has memorised a benchmark, by checking whether it assigns higher probability to the test examples in their published order than in shuffled order.

They saturate. When top models all score above 90%, the remaining differences are within the noise of the benchmark's own errors, and it can no longer rank them. The useful lifetime of a new benchmark has shortened as models have improved faster.

Under all three sits a problem of validity. Raji and colleagues, in a paper titled "AI and the Everything in the Whole Wide World Benchmark", argued that general-purpose benchmarks are taken as measures of broad capability that they were never designed to measure. A high score certifies performance on that test, under those conditions.

The gap between benchmark and deployment

For an enterprise, the more pressing issue is that even a perfect public benchmark measures the wrong thing.

A model's score on general coding tasks says little about how it handles one company's codebase, with its conventions and its legacy systems. A model's score on medical licensing questions says little about how it summarises discharge notes in a particular hospital's format. Recht and colleagues' finding that accuracy fell by 11 to 14 points when ImageNet models met a carefully reconstructed version of their own test set shows how sensitive results are to small changes in the data, and deployment involves much larger ones.

Agentic systems widen the gap further. Kapoor and colleagues, in a study titled "AI Agents That Matter", found that agent benchmarks often lacked proper held-out sets, that reported results were difficult to reproduce, and that leaderboards ignored cost, so a system could top the table by making many expensive attempts. Evaluating an agent means evaluating a sequence of actions in an environment, and each test case requires someone to build the environment, define success and confirm that success is achievable.

Anthropic's own account of the difficulties is direct. In a 2023 essay on the challenges of evaluating AI systems, its researchers described how multiple-choice benchmarks were sensitive to trivial formatting changes, how third-party frameworks required substantial effort to run faithfully, and how human evaluations were slow and hard to make consistent. The conclusion was that robust evaluation is much harder than it looks from outside.

KEY TAKEAWAY

A benchmark score answers the question "how did this model do on that data?" Whether the answer transfers to your use is a separate question, and only your own data can address it.

Evaluation as a production process

If static benchmarks decay, evaluation becomes something that has to be manufactured continuously. Several approaches from the research literature describe what that looks like.

Dynamic benchmarks. Dynabench, from Kiela and colleagues, replaced the fixed test set with a loop. People write examples intended to fool the current best model. Examples that succeed, and that other people verify, go into the next round's dataset. The benchmark moves as models improve.

Behavioural testing. CheckList, from Ribeiro and colleagues, borrowed from software testing. Instead of one aggregate score, a model is tested against a matrix of specific capabilities, with test cases written or templated by people: does it handle negation, does its answer survive a change of name, does it stay consistent under paraphrase. The authors found failures in commercial systems that had passed standard benchmarks.

Private and refreshed test sets. The simplest defence against contamination is not to publish. A test set that has never been on the internet cannot be in the training data. This requires that the set be created originally, kept under access control, and replaced periodically, because every query to a hosted model is also a potential leak. Jacovi and colleagues proposed practical measures for public benchmarks as well, including encrypting test data and refusing to send it unprotected to model APIs.

Human verification of automatic graders. The SWE-bench exercise is the template. Before trusting an automated metric, have qualified people confirm that the tasks are well posed and the grading is fair.

Each of these is a data-production activity. Each requires people with relevant skill writing, checking and curating examples to a specification. Evaluation data has become a product with a manufacturing process, a shelf life and a security requirement.

Why not rely on public benchmarks and model judges?

There is a reasonable case for a lighter approach. Public benchmarks are free and comparable. Automated judges, discussed in the previous article, are fast. A team could select a model from leaderboard results and monitor it with model-graded checks, without building any evaluation data of its own.

For low-stakes uses, that may be sufficient. Its limits are those already described. Leaderboard position reflects performance on tasks that may be contaminated and are not yours. Model judges need calibrating against human judgement on your task. And neither can tell you about failure modes specific to your data, which are the ones that will cost you.

A second objection is that custom evaluation is expensive. In absolute terms it is modest. A few hundred carefully built, expert-verified test cases is a small dataset. Compared with the cost of selecting the wrong model, or shipping a regression nobody measured, it is among the better investments available.

What an evaluation programme contains

A serious evaluation capability has five parts.

A private core set. Several hundred to a few thousand cases drawn from real or realistic inputs, with answers or rubrics established by qualified people. Never published. Access logged.

A slice structure. Cases tagged by the dimensions that matter: task type, difficulty, language, customer segment, risk level. Results are reported by slice.

A refresh cycle. A portion of the set retired and replaced on a schedule, with new cases drawn from recent production data and newly observed failures.

A failure pipeline. Every significant production error becomes a test case. This is how the evaluation set learns.

A verification step. People periodically confirm that test cases are still well posed and that automated graders still agree with expert judgement.

What this changes

FOR AI BUILDERS

Build a private evaluation set before selecting a model, and keep it out of prompts sent to third-party APIs wherever possible.

When a benchmark number matters to a decision, check for contamination: test on freshly written variants and compare.

Have qualified people audit your automatic graders. An unfair test case is a labelling error with a different name.

FOR ENTERPRISES

Do not select models on public leaderboard position alone. Require results on data that resembles your own.

Treat evaluation data as a long-lived, access-controlled asset with an owner and a refresh budget.

Ask vendors whether their reported results could be affected by training-data contamination, and how they know.

THE INDIKA PERSPECTIVE

Evaluation is where the argument of this series is clearest. The data is small, so volume is irrelevant. It must be correct, so expertise and adjudication matter. It must be new, so it cannot be scraped. It must be private, so it needs secure handling. And it has to be renewed indefinitely. An organisation that can produce such data to order has something more durable than a model.

RESEARCH & REFERENCES

Jimenez, C. et al. (2024). "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" ICLR.

OpenAI (2024). "Introducing SWE-bench Verified." Research publication, August.

Zhang, H. et al. (2024). "A Careful Examination of Large Language Model Performance on Grade School Arithmetic." NeurIPS Datasets and Benchmarks.

Mirzadeh, I. et al. (2025). "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models." ICLR.

Oren, Y. et al. (2024). "Proving Test Set Contamination in Black-Box Language Models." ICLR.

Jacovi, A. et al. (2023). "Stop Uploading Test Data in Plain Text: Practical Strategies for Mitigating Data Contamination by Evaluation Benchmarks." EMNLP.

Raji, I. D. et al. (2021). "AI and the Everything in the Whole Wide World Benchmark." NeurIPS Datasets and Benchmarks.

Recht, B. et al. (2019). "Do ImageNet Classifiers Generalize to ImageNet?" ICML.

Kapoor, S. et al. (2024). "AI Agents That Matter." Transactions on Machine Learning Research.

Kiela, D. et al. (2021). "Dynabench: Rethinking Benchmarking in NLP." NAACL.

Ribeiro, M. T. et al. (2020). "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList." ACL.

Anthropic (2023). "Challenges in evaluating AI systems." Research essay.

Liang, P. et al. (2023). "Holistic Evaluation of Language Models." Transactions on Machine Learning Research.

Red Teaming Is Dataset Construction

Adversarial testing of AI systems is usually described as a security exercise. Its lasting output is a dataset, and it inherits every question this collection has asked about datasets.

7 MINUTE READ

In 2023, researchers at Brown University took a set of requests that GPT-4 reliably refused, such as instructions for causing harm, and ran them through a translation service into languages with little presence on the internet, including Zulu and Scots Gaelic. They submitted the translated requests and translated the responses back into English.

The model's safeguards, which had held in English, largely gave way. Across the low-resource languages tested, the combined attack succeeded 79% of the time on a standard benchmark of harmful requests. In high-resource languages the success rate was far lower.

No sophisticated technique was involved. The weakness existed because the data used to teach the model what to refuse was overwhelmingly in English and a few other major languages. The safety training had a coverage gap, and the gap had the same shape as the data.

That is a finding about red teaming as much as about the model. Somebody had to think of trying it. Whether a vulnerability is found depends on who is looking and what they know to look for.

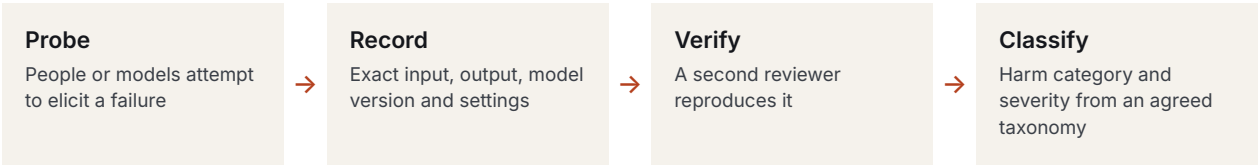
What red teaming produces

The term comes from military and security practice, where a designated team plays the adversary. Applied to AI, it means deliberately trying to make a system behave badly: produce harmful content, leak private information, follow malicious instructions, give dangerous advice.

The activity gets the attention. The output is what lasts. A red-team exercise produces a collection of inputs that caused failures, the outputs they caused, and a classification of what went wrong. That is a dataset. It has three downstream uses, all of them data uses.

FIGURE 17 · PROCESS MAP

From a finding to three datasets



THEN USED AS	PURPOSE	CONDITION FOR IT TO WORK
Evaluation data	A regression suite re-run on every model or prompt change	Held out from training
Training data	Attack paired with the desired response, to teach safer behaviour	Kept separate from the evaluation split
Classifier data	Examples for input and output filters in production	Labels consistent enough to learn from

Synthesis by the editors, after Ganguli et al. (2022) and Feffer et al. (2024).

- As **evaluation data**. The successful attacks become a test suite. Every future version of the model can be checked against them.
- As **training data**. The attacks, paired with the responses the model should have given, are used to teach refusal or safer behaviour.
- As **classifier data**. Examples of harmful inputs and outputs train the separate filters that screen traffic in production.

Anthropic’s early work on the subject made this explicit. Ganguli and colleagues ran a large human red-teaming effort against language models of several sizes and released the result as a dataset of 38,961 attacks, with analysis of what kinds of harmful output were elicited. The paper is as much about how to collect and label such data as about the vulnerabilities found. Among its observations was that models trained with human feedback became harder to red team as they grew larger, while other configurations did not.

The annotation problems, again

Once red teaming is seen as data production, the questions from the earlier parts of this series apply directly.

Coverage. A red-team dataset represents the attacks its creators thought of. The translation attack shows what follows when the team’s languages are narrower than the users’. Deng and colleagues documented the same pattern across a wider set of languages: unsafe responses became more frequent as the language’s resource level fell. Coverage gaps in safety data map onto gaps in who was recruited to probe.

Whose judgement. Whether an output is harmful is frequently contested. A frank answer about medication doses is a safety failure in one context and appropriate in another. Weidinger and colleagues at Google DeepMind argue that safety evaluation has to consider not only what a model can output but who is interacting with it and in what setting. Red-team labels are preference data with the stakes raised, and the problems of rater selection discussed in previous articles recur.

Agreement. Two red-teamers may disagree about whether an attack succeeded, how severe the result was, and which category it belongs to. Without a taxonomy and measured agreement, counts of “successful attacks” are not comparable between exercises.

Documentation. A finding that is not written down in reproducible form, with the exact input, the model version, the settings and the output, cannot be re-tested and cannot be used.

Feffer and colleagues surveyed published red-teaming exercises and found wide variation on all of these: who took part, what they were asked to do, what resources they had, what counted as a finding and what was reported afterwards. Their title asked whether red teaming, as practised, was a silver bullet or security theatre. Their answer was that without agreed structure it could not serve as the assurance mechanism policy discussions were treating it as.

Scale and structure

There are broadly three ways to produce adversarial data, and they complement each other.

Expert red teaming uses small groups with deep knowledge of a domain, such as biosecurity, child safety, fraud or a particular culture and language. It finds the failures that require knowledge to imagine. It is slow, expensive and does not provide broad coverage.

Crowd red teaming uses many people with varied backgrounds. The Generative Red Team event at the DEF CON security conference in 2023 brought more than two thousand participants to test models from several developers. Breadth of perspective is the benefit. Consistency of method and labelling is the cost.

Automated red teaming uses models and algorithms to generate attacks. Perez and colleagues showed that language models could write evaluation datasets testing for a wide range of behaviours, with human reviewers confirming the quality of most generated examples. Zou and colleagues showed that an automated search could discover strings of characters that, appended to a request, caused several different models to comply with harmful instructions, and that these strings transferred between models. Benchmarks such as HarmBench standardise the comparison of automated methods.

Automation gives volume and repeatability. It explores variations on what it was pointed at. The translation attack, and others like it, came from a person having an idea about how the system might be mismatched to the world. The usual pattern is that humans discover a new class of failure and automation then fills in its many instances.

KEY TAKEAWAY

Red teaming finds what its participants can imagine. A model tested only by English-speaking engineers has been tested against English-speaking engineers.

What regulation now expects

Adversarial testing has moved from voluntary practice toward obligation. The European Union's AI Act requires providers of general-purpose models with systemic risk to perform and document adversarial testing. The United States National Institute of Standards and Technology published a profile of its AI Risk Management Framework for generative AI in 2024, which lists red teaming among its recommended measures.

These frameworks ask for documentation: what was tested, by whom, with what result. That is a requirement for data with provenance.

The limits of the exercise

Three cautions are in order.

Red teaming demonstrates the presence of failures and cannot demonstrate their absence. An exercise that finds nothing may mean the system is robust or that the team lacked imagination, time or the right languages.

Training on red-team data can produce a model that resists the specific attacks it was shown and little else. If the same findings are used for both training and evaluation, the evaluation will overstate robustness. Held-out adversarial data, kept from training, is necessary for an honest measurement, for the same reasons the previous article gave for any test set.

And the work has a human cost. Red-teamers and the annotators who label harmful content are exposed to disturbing material as a condition of the job. Research on content moderators has documented psychological harm, and reporting on data workers has raised the same concern for those labelling toxic content for AI systems. An operation that produces safety data has a duty of care to the people producing it, including limits on exposure, support and fair pay.

Running it as a data programme

Treated as dataset construction, a red-team programme has a specification.

A taxonomy of harms, tailored to the application, with definitions and severity levels. Without it, findings cannot be counted or compared.

A recruitment plan that deliberately covers the languages, regions, domains and user groups the system will meet. Diversity here is a coverage requirement.

A recording standard: the exact input, the model version and configuration, the full output, the tester's classification, and a second person's verification.

A split between findings used for training and findings held back for evaluation.

A refresh schedule. Models change, attackers adapt and yesterday's suite goes stale.

A welfare policy for the people doing the work.

What this changes

FOR AI BUILDERS

Store every finding in reproducible form with a harm category and severity, verified by a second reviewer.

Hold back a portion of adversarial data from training and use it only for evaluation.

Test in every language and dialect your users write in, including mixed-language and transliterated input. Safety training does not transfer evenly across languages.

FOR ENTERPRISES

Ask what adversarial testing a system has had, by whom, in which languages and against what taxonomy. "It was red-teamed" is not an answer.

Include domain experts and people from your user populations among testers. Generic red teams find generic problems.

Treat red-team results as a living test suite to be re-run on every model or prompt change.

THE INDIKA PERSPECTIVE

Safety data is where multilingual and cultural coverage stops being a matter of inclusion and becomes a matter of security. A model deployed in India will be probed in Hindi, Tamil, Bengali, Hinglish and transliterated text, by people who know which local references and framings a filter trained on English will miss. Finding those failures first requires testers who share that knowledge, and a process that turns what they find into verified, reusable data.

RESEARCH & REFERENCES

Yong, Z.-X., Menghini, C. and Bach, S. (2023). "Low-Resource Languages Jailbreak GPT-4." NeurIPS Workshop on Socially Responsible Language Modelling Research.

Ganguli, D. et al. (2022). "Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned." Anthropic, arXiv preprint.

Deng, Y. et al. (2024). "Multilingual Jailbreak Challenges in Large Language Models." ICLR.

- Feffer, M. et al. (2024). "Red-Teaming for Generative AI: Silver Bullet or Security Theater?" AAAI/ACM Conference on AI, Ethics, and Society.
- Weidinger, L. et al. (2023). "Sociotechnical Safety Evaluation of Generative AI Systems." Google DeepMind, arXiv preprint.
- Perez, E. et al. (2023). "Discovering Language Model Behaviors with Model-Written Evaluations." ACL Findings.
- Zou, A. et al. (2023). "Universal and Transferable Adversarial Attacks on Aligned Language Models." arXiv preprint.
- Mazeika, M. et al. (2024). "HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal." ICML.
- NIST (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1).
- European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act), Article 55.
- Steiger, M. et al. (2021). "The Psychological Well-Being of Content Moderators: The Emotional Labor of Commercial Moderation and Avenues for Improving Support." ACM CHI.
- Perrigo, B. (2023). "Exclusive: OpenAI Used Kenyan Workers on Less Than \$2 Per Hour to Make ChatGPT Less Toxic." TIME, January.

The Synthetic Data Paradox

Generated data works best in the domains we already understand well enough to simulate. It works worst where real data is hardest to get, which is where it is most often proposed.

7 MINUTE READ

In 2024, a system from Google DeepMind solved geometry problems from the International Mathematical Olympiad at a level approaching that of a gold medallist. It had been trained on one hundred million examples of geometric theorems and proofs. Not one of them was written by a person. All were generated by a program that built random geometric figures and derived what could be proved about them.

Two years earlier, a group of researchers had examined a different use of synthetic data. Organisations wanted to share sensitive records, such as medical data, without exposing individuals. The proposal was to train a generative model on the real records and release its synthetic output instead. Stadler and colleagues tested whether this protected privacy. Their conclusion, published at a leading security conference, was that it offered no better trade-off than older anonymisation techniques. Synthetic data that preserved enough detail to be useful also preserved enough to leak information about real individuals, especially unusual ones. Synthetic data that was safe was of little use.

One result is a triumph of synthetic data. The other is a warning about it. The difference between them is the whole subject.

What a generator knows

Every synthetic dataset is produced by something: a physics simulator, a rendering engine, a rule-based program or a trained generative model. The data it emits can contain no more information about the world than the generator has.

That gives a simple way to sort the uses.

FIGURE 18 · FRAMEWORK

What a synthetic dataset can contain depends on its generator

GENERATOR'S SOURCE OF TRUTH	EXAMPLES	WHAT GENERATION ADDS	MAIN LIMIT
Known rules	Geometry, arithmetic, executable code, games	Unlimited, perfectly labelled examples	Only as broad as the rule system
A physical simulator	Rendered images, robot and driving simulation	Exact labels; rare or dangerous scenes on demand	The gap between simulation and reality
A model trained on real data	Generated text, images, tabular records	Volume, rebalancing, variation of what is already known	Thin tails, no new domains, memorised records, consistent errors

Synthesis by the editors, after Trinh et al. (2024), Tobin et al. (2017), He et al. (2023) and Stadler et al. (2022).

When the generator encodes rules that are known to be true, synthetic data is close to ideal. Geometry has axioms. A program that applies them correctly produces valid theorems, unlimited in number and perfectly labelled. The same holds for other formal domains: arithmetic, code that can be executed and tested, games with fixed rules. The generator knows the domain better than any dataset could.

When the generator is a faithful simulator of physical processes, synthetic data is useful to the degree the simulation is accurate. Renderers can produce images with exact labels for depth, object boundaries and position, which are expensive or impossible to obtain from photographs. A Microsoft team trained face-analysis models entirely on rendered synthetic faces and achieved accuracy competitive with models trained on real photographs, having invested heavily in the realism of the rendering. Robotics researchers train policies in simulation and transfer them to hardware, using a technique called domain randomisation, which varies textures, lighting and physics so widely that the real world looks like one more variation. The limiting factor is the remaining difference between simulation and reality, which practitioners call the sim-to-real gap.

When the generator is a model trained on real data, the synthetic data is a resampling of what that model learned. It can be valuable. It cannot contain anything the model did not absorb, and it will under-represent what the model absorbed weakly: the rare cases.

That third category is where most enterprise proposals fall, and where the paradox applies.

Where learned generators help

The evidence that model-generated data can improve models is solid within limits.

Azizi and colleagues at Google fine-tuned an image-generation model and used it to produce additional training images for ImageNet classes. Adding these to the real training set improved classification accuracy. The synthetic images were an augmentation of the real data, not a substitute for it.

Language models are routinely used to produce training data for smaller models. Microsoft's phi models were trained in part on synthetic "textbook-like" text and exercises generated by a larger model, and reached strong performance for their size. This is a form of distillation: knowledge held by a large model is transferred to a small one through generated examples.

Other dependable uses include rebalancing, by generating more examples of an under-represented class the generator does know about; creating variations for robustness, such as paraphrases or changes of format; and producing test cases to a specification.

What these have in common is that the generator already possesses the relevant knowledge. The synthetic data moves it, reshapes it or multiplies it.

Where they do not

The long tail. A learned generator reproduces the distribution it was trained on, and tends to do so with reduced variety. Rare events, unusual presentations and minority subgroups are thin in the training data and thinner still in the output. A study by He and colleagues on using synthetic images for recognition found benefits concentrated in settings with little or no real data, with gains diminishing as real data became available, and noted the gap in diversity between generated and real images. If the reason for wanting synthetic data is that the real data lacks edge cases, a generator trained on that real data lacks them too.

New domains. An enterprise that wants a model for its own documents, in its own terminology, cannot get there by asking a general model to imagine such documents. The general model will produce plausible text that resembles what it has seen elsewhere. The specifics that make the enterprise's data distinctive are exactly what it does not know.

Privacy. As Stadler's work showed, synthetic data is not automatically anonymous. Generative models can memorise and reproduce training examples. Carlini and colleagues extracted recognisable copies of training images from diffusion models, with duplicated and unusual images most at risk. A synthetic dataset derived from personal records needs the same privacy analysis as the records.

Errors at scale. A human annotator who misunderstands a guideline mislabels some items. A generator with a systematic flaw embeds it in every example, with perfect consistency. Because synthetic data is cheap, the flawed dataset can be very large.

KEY TAKEAWAY

Synthetic data transfers and multiplies knowledge. It does not create it. Before generating, ask where the generator's knowledge of your problem came from.

The verification problem

There is a second half to the paradox. To know whether a synthetic dataset is good, you have to compare it with reality. That requires real data.

The Royal Society and the Alan Turing Institute, in a 2022 overview of the field, made the point that the utility and privacy of synthetic data cannot be assumed from the method that produced it. They have to be evaluated empirically for each use. Evaluating utility means training a model on the synthetic data and testing it on real data. Evaluating fidelity means comparing distributions against a real sample.

So synthetic data does not remove the need for real, well-labelled data. It changes the quantity needed and what it is used for. A small, trusted real dataset becomes the reference against which large volumes of generated data are validated. If that reference is poor or absent, nobody can tell whether the synthetic data is helping.

A sensible forecast for most applied settings is a ratio that shifts, not a replacement. More of the training volume becomes synthetic. The real data that remains is concentrated in evaluation, in seeding the generator, and in the rare cases the generator cannot produce.

The case for optimism

The sceptical reading should not be overdone. There are three reasons to expect synthetic data to matter more.

Generators are improving. Each increase in the fidelity of simulators and generative models extends the range of problems where their output is good enough.

The public data supply is tightening, as the first article in this series described. Where real data is legally restricted, exhausted or prohibitively expensive, an imperfect synthetic alternative may be the only option.

And the most striking recent progress has come from systems that generate their own training signal in domains with a verifier: programs that can be run, proofs that can be checked, games that can be scored. Where an automatic check exists, a model can generate attempts, keep the ones that pass, and learn from them, improving beyond its starting point. This is the regime of the geometry result.

The critical word is verifier. In these cases new information does enter the system, from the checker. Without an external check, a model learning from its own output has no source of correction, a problem the next article examines.

A decision procedure

For a team considering synthetic data, four questions settle most cases.

What is the source of truth in the generator? Known rules, a validated simulator, or a model trained on data. The answer bounds what the synthetic data can contain.

What is missing from the real data, and does the generator have it? If the gap is volume of well-understood cases, generation helps. If the gap is rare or novel cases, it probably does not.

How will it be validated? Name the real dataset that will serve as the test. If none exists, build that first.

What is the mix? Plan to combine synthetic with real data, and measure how performance changes as the proportion varies.

What this changes

FOR AI BUILDERS

Always evaluate on real data. A model trained and tested on synthetic data has been measured against the generator's view of the world.

Use generation to rebalance and augment known categories. Collect real examples for the rare and the new.

Check synthetic outputs for near-copies of training records before treating them as privacy-safe.

FOR ENTERPRISES

Treat claims that synthetic data removes the need for real data with caution. Ask what the generator was trained on and how the output was validated.

Do not assume synthetic versions of personal data are exempt from privacy obligations. Seek a specific assessment.

Invest in a small, high-quality real reference dataset. It determines whether any synthetic data programme can be trusted.

THE INDIKA PERSPECTIVE

Synthetic data and human-produced data are complements. Generation handles volume where the pattern is known. People supply what generators lack: the rare case, the new domain, the expert judgement, and the real reference set against which everything generated is checked. The practical skill is in designing the combination and knowing, for a given problem, which part each should carry.

RESEARCH & REFERENCES

- Trinh, T. et al. (2024). "Solving olympiad geometry without human demonstrations." *Nature*.
- Stadler, T., Oprisanu, B. and Troncoso, C. (2022). "Synthetic Data – Anonymisation Groundhog Day." *USENIX Security Symposium*.
- Wood, E. et al. (2021). "Fake It Till You Make It: Face analysis in the wild using synthetic data alone." *IEEE/CVF ICCV*.
- Tobin, J. et al. (2017). "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World." *IEEE/RSJ IROS*.
- Tremblay, J. et al. (2018). "Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization." *IEEE/CVF CVPR Workshops*.
- Azizi, S. et al. (2023). "Synthetic Data from Diffusion Models Improves ImageNet Classification." *Transactions on Machine Learning Research*.
- Gunasekar, S. et al. (2023). "Textbooks Are All You Need." *Microsoft Research, arXiv preprint*.
- He, R. et al. (2023). "Is Synthetic Data from Generative Models Ready for Image Recognition?" *ICLR*.
- Carlini, N. et al. (2023). "Extracting Training Data from Diffusion Models." *USENIX Security Symposium*.
- Jordon, J. et al. (2022). "Synthetic Data – what, why and how?" *The Royal Society and The Alan Turing Institute*.

Model Collapse, Read Carefully

A 2024 paper in Nature showed that models trained on their own output degrade over generations. What the result implies depends on an assumption that follow-up work has examined closely.

7 MINUTE READ

The experiment is simple to describe. Train a language model on human-written text. Use it to generate a new corpus. Train a second model on that corpus alone. Use the second model to generate a third corpus, and repeat.

In 2024, Shumailov and colleagues reported in Nature what happens. Over successive generations, the models degrade. First they lose the tails of the distribution: rare words, unusual constructions, minority patterns. The authors call this early collapse. Later, the output converges on a narrow band with little resemblance to the original data, which they call late collapse. In one demonstration, a model prompted with text about medieval architecture was, several generations on, producing repetitive text about jackrabbits.

The paper gave the phenomenon a memorable name, model collapse, and it was widely reported as a prediction that AI systems would poison themselves as the internet filled with machine-written text.

That reading goes beyond what the experiment shows. The follow-up research is more precise, and more useful to anyone making decisions about data.

Why it happens

The mechanism is statistical and not specific to any architecture.

A model trained on a finite sample cannot represent rare events as well as common ones. When it generates, the rare events are slightly under-produced. The next model, trained on that output, sees even fewer, and represents them less well again. Each generation samples from an approximation of the previous one, and the errors compound in one direction: toward the centre of the distribution.

Other groups observed the same effect in images. Alemohammad and colleagues trained generative image models in what they called an autophagous, or self-consuming, loop and found that without enough fresh real data in each generation, quality or diversity progressively declined. They named the condition Model Autophagy Disorder.

The qualifier in that sentence matters: without enough fresh real data.

The assumption that decides the outcome

The original experiment discards the previous data at each step. Generation three is trained only on the output of generation two. The human-written starting corpus is gone.

Gerstgrasser and colleagues asked whether that matches reality. On the actual internet, old data does not vanish when new data is published. Human-written text from 2015 is still there. Machine-generated text is added alongside it. The better model of the real situation is accumulation, not replacement.

They re-ran the experiments both ways, across language models, image models and other generative systems.

FIGURE 19 · COMPARISON TABLE

Two regimes, two outcomes

	REPLACE	ACCUMULATE
Each generation trains on	Only the previous generation's output	The original real data plus all synthetic data so far
Real data over time	Discarded after the first generation	Kept, though its share shrinks
Reported result	Error grows with each generation; tails vanish first	Error stays bounded across generations
Closest real-world analogue	A pipeline that overwrites its datasets with generated ones	The web, where old text persists as new text is added
Remaining risk	Collapse	Slower loss of diversity if the real share becomes small

After Shumailov et al. (2024), Gerstgrasser et al. (2024) and Dohmatob et al. (2025).

Under replacement, they reproduced the collapse. Under accumulation, where each generation trained on the original real data together with all synthetic data produced so far, test error did not grow without limit. It stayed bounded. Their theoretical analysis of a simplified model confirmed the same result: with accumulation, error has a finite ceiling regardless of the number of generations.

That finding substantially changes the practical message. Collapse is not an inevitable consequence of synthetic data existing in the world. It is a consequence of losing the real data.

How reassuring is that?

Not completely, for three reasons.

Bounded is not the same as unharmed. A model whose error stays below a ceiling may still be worse than it would have been with clean data. Dohmatob and colleagues studied this under the heading of strong model collapse. In their theoretical setting, even a very small proportion of synthetic data in the training mix, on the order of one per cent, could prevent performance from improving as the dataset grew. Adding more data did not wash the effect out. The result is for simplified models, and how far it carries to large systems is an open question, but it argues against complacency.

The proportion of real data falls over time. Accumulation keeps the old human-written data. It does not keep it at a constant share. If machine-generated content is added faster than human-written content, the real fraction shrinks, and the results on fresh data suggest that fraction matters.

Diversity erodes before quality does. Early collapse is the loss of tails. A model can remain fluent and accurate on average while becoming less varied and worse on unusual inputs. Studies comparing machine-generated text with human writing have found reduced lexical and syntactic variety. Standard quality metrics, which measure the average case, will not show this.

The fair summary is that catastrophic collapse requires conditions unlikely to hold at internet scale, while a slower narrowing remains a credible risk that is hard to see with ordinary metrics.

Is the web filling with synthetic text?

The risk depends on how much machine-generated content is entering the data that models train on. Direct measurement is difficult, but there are indications.

A study by Thompson and colleagues at Amazon examined multi-way parallel text on the web, meaning content that appears in many languages. They found that a large share of such content showed signs of machine translation, and that it was disproportionately low in quality and concentrated in lower-resource languages. For a language with little original web content, a substantial fraction of what exists online may already be machine output.

In a narrower domain, Liang and colleagues estimated the proportion of text in peer reviews at major AI conferences that had been substantially modified or produced by language models after ChatGPT's release. Their estimates ranged from 6.5% to 16.9%.

These are specific measurements, not a census of the web. They show that the contamination is uneven. It is likely to be heaviest precisely where human-written data is scarcest, which is where a model has the least real signal to anchor it.

KEY TAKEAWAY

The lesson of model collapse is about bookkeeping. Models do not fail because synthetic data exists. They are at risk when nobody can tell which data is synthetic and the real data is no longer kept apart.

From a modelling problem to a provenance problem

If the key variable is the proportion and identity of real data in the mix, then the central capability is knowing where each piece of data came from.

For data an organisation generates itself, this is easy in principle and often neglected in practice. Synthetic examples should be labelled as such at creation, with the generating model and date recorded, and that label should travel with them. Many pipelines mix generated and collected data into one training file with no marker.

For data from outside, it is harder. Reliable detection of machine-generated text remains unsolved, and detectors perform worse on text by non-native writers and on edited output. Timestamps help: data from before late 2022 is very unlikely to be the output of a modern language model, which gives older archives a particular value. Source matters too. Data obtained directly from a known human process, such as a company's own records, a commissioned annotation project or a licensed archive, has a provenance that scraped text does not.

This is one reason first-party and commissioned data is appreciating. Its value is no longer only that it is relevant. It is that its origin is known.

Should we worry less?

A sceptic could reasonably argue that the concern is overstated. Frontier developers filter their training data heavily. Much of the best recent progress involves deliberately training on synthetic data, including model-generated reasoning traces and distilled outputs, with good results. If synthetic data were as harmful as the collapse narrative implies, that would not work.

The observation is correct and consistent with the research. In those successful cases the synthetic data is curated: generated to a purpose, filtered by a verifier or a quality model, and combined with real data. The previous article described the

conditions. Kazdan and colleagues, examining different training regimes, likewise found that outcomes depended heavily on how synthetic and real data were combined across generations.

The risk identified by the collapse literature is the uncured case: synthetic data of unknown origin and quality entering training without anyone having chosen it. The remedy is curation and provenance, both of which are data-management disciplines.

What to do

Label synthetic data at source. Record that it is generated, by what, when, and from which inputs. Keep the label attached through every transformation.

Preserve the real data. Never overwrite or discard verified human-origin datasets in favour of generated ones. Version them and store them separately.

Track the ratio. For each training run, know the proportion of real to synthetic data, overall and by category.

Monitor the tails. Evaluate on rare cases and measure output diversity directly, since average quality metrics lag.

Prefer known sources. Where it matters, use data whose human origin can be established: first-party records, commissioned collection, dated archives, licensed corpora.

Take extra care in low-resource languages. The available web text is more likely to be machine-translated, and there is less human-written data to offset it.

What this changes

FOR AI BUILDERS

Add a provenance field to every training record, at minimum distinguishing human-origin, machine-generated and unknown.

When using synthetic data over several rounds, always retain and include the original real data.

Include diversity and rare-case metrics in evaluation. Collapse begins in the tails.

FOR ENTERPRISES

Regard verified human-origin data, including your own historical records, as an asset whose value is rising.

Ask data suppliers how they establish that data is human-produced, and how they prevent undisclosed use of generative tools by annotators.

Be especially cautious with scraped data in languages with little original web content.

THE INDIKA PERSPECTIVE

Human-origin data used to be the default and needed no label. It now needs to be certified. That places a new requirement on any organisation that produces data with people: to show that people produced it. Controls on annotator tool use, audit trails, timestamps and process evidence stop being administrative overhead and become part of what makes the data worth buying.

RESEARCH & REFERENCES

Shumailov, I. et al. (2024). "AI models collapse when trained on recursively generated data." *Nature*.

Alemohammad, S. et al. (2024). "Self-Consuming Generative Models Go MAD." *ICLR*.

Gerstgrasser, M. et al. (2024). "Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data." *arXiv preprint*.

Dohmatob, E. et al. (2025). "Strong Model Collapse." *ICLR*.

Kazdan, J. et al. (2024). "Collapse or Thrive? Perils and Promises of Synthetic Data in a Self-Generating World." *arXiv preprint*.

Guo, Y. et al. (2023). "The Curious Decline of Linguistic Diversity: Training Language Models on Synthetic Text." *arXiv preprint*.

Thompson, B. et al. (2024). "A Shocking Amount of the Web is Machine Translated: Insights from Multi-Way Parallelism." *ACL Findings*.

Liang, W. et al. (2024). "Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews." *ICML*.

Liang, W. et al. (2023). "GPT detectors are biased against non-native English writers." *Patterns*.

Bohacek, M. and Farid, H. (2023). "Nepotistically Trained Generative-AI Models Collapse." *arXiv preprint*.

Time Is the Hardest Label

Video is often treated as a stack of images to be annotated one after another. The difficult part of video is everything that happens between the frames.

7 MINUTE READ

One hour of video recorded at thirty frames per second contains 108,000 images.

That arithmetic is the usual starting point for explaining why video annotation is expensive, and it is misleading. If video were only a large number of images, the problem would be one of throughput, and throughput problems yield to money and tooling. Adjacent frames are nearly identical, so most of those 108,000 images carry almost no new information. Label one frame in ten and interpolate, and the volume problem largely goes away.

What remains is harder. A photograph can be labelled by answering “what is here?” A video poses questions that a photograph cannot: is this the same object as before, when did this begin, when did it end, what caused it, what is about to happen. None of these is about a frame. Each is about the relationship between frames, and each turns out to be far more ambiguous than drawing a box.

Four things that do not exist in an image

FIGURE 20 · COMPARISON TABLE

Four questions a still image never asks

QUESTION	WHY ANNOTATORS DIFFER	WHAT A GUIDELINE HAS TO FIX
Identity Is this the same object as before?	Occlusion, crossing paths, exit and re-entry	Rules for when a track continues, ends or restarts
Temporal boundary When does it start and end?	Actions have no sharp edges	An observable anchor event for each boundary
Granularity At what level do we describe it?	“Cooking” and “cracking an egg” are both true	One level of description, with boundary examples
Intent and cause What is about to happen, and why?	Requires inference about what is not visible	The context window annotators see, and who is qualified to judge

Synthesis by the editors, after Sigurdsson et al. (2017), Moltisanti et al. (2017) and Gu et al. (2018).

Identity. In a still image, each object is labelled once. In video, an object must keep the same identifier across every frame in which it appears. That is simple while it stays in view. It becomes a judgement when the object passes behind something and re-emerges, when two similar objects cross paths, or when it leaves the frame and something like it enters a minute later. Annotators have to decide whether this is the same car, and the answer has consequences for any model that learns to track.

Boundaries in time. An action has a start and an end. Where exactly does “picking up a cup” begin: when the hand starts moving, when it touches the cup, or when the cup leaves the table? Studies of temporal annotation have found that different annotators place these boundaries differently, and that models are sensitive to the inconsistency. Sigurdsson and colleagues, analysing action recognition datasets, found considerable ambiguity among people about the temporal extent of everyday activities. Moltisanti and colleagues showed that for interactions with objects in first-person video, labelled start and end times varied enough between annotators to change measured model accuracy, and proposed anchoring boundaries to an explicit, defined event to reduce the variation.

Granularity. The same thirty seconds can be described as “cooking”, “making an omelette”, “cracking an egg” or “moving the right hand toward the bowl”. All are correct. A dataset must choose a level, and annotators must hold to it. The AVA dataset from Google took one approach, defining eighty atomic visual actions, such as standing, holding or talking to, and labelling them for each person at one-second intervals, so that complex activities could be composed from simple parts. Other datasets label whole clips with a single high-level category. These are different tasks producing different models.

Intent and causation. Many valuable video tasks require inferring what cannot be seen. Is this pedestrian about to cross? Is this person shoplifting or putting their own phone in their pocket? Did the worker follow the safety procedure? The label depends on events before and after, on context, and on an inference about a mental state. Two attentive annotators can watch the same clip and disagree in good faith.

What large video datasets reveal

The scale of effort behind research video datasets gives a sense of the difficulty.

Kinetics, one of the standard action-recognition benchmarks, sidestepped temporal annotation by design. Each clip is about ten seconds long and carries one label for the whole clip. This made large-scale collection feasible and limited what could be learned: the label says an action occurs somewhere in the clip, not when.

Something-Something took a different route. Crowd workers were given a template, such as “pushing something from left to right”, and asked to film themselves performing it. The dataset was built to force models to attend to

motion: the same objects appear in different actions, so the action cannot be guessed from a single frame. Here the annotation was moved upstream, into the instructions for recording.

Ego4D is the most ambitious example. A consortium of universities and Meta collected 3,670 hours of first-person video from 931 camera wearers in 74 locations across nine countries. The annotation covered narrations of what the wearer was doing, the moments at which objects changed state, who was speaking and to whom, and forecasts of what would happen next. Each layer needed its own guidelines, its own training and its own quality process. The paper describes an annotation effort on a scale few projects have matched.

The differences between these datasets are differences of annotation design. What a video model can do is determined in large part by which temporal questions someone decided to ask and how carefully they defined the answers.

Tools change the cost, not the judgement

Video annotation tooling has improved a great deal, and the gains are real.

The basic technique is keyframe interpolation: an annotator marks an object on selected frames and software fills in the frames between. Vondrick and colleagues studied this in early work on crowdsourced video annotation and reported a finding that still holds. The trade-off between human effort and automation was not simple. Fixed-interval keyframes were inefficient, and the best strategy depended on how the object moved. They also found that getting good results from non-expert annotators took considerable investment in interface and instructions.

More recent tools use learned models to track objects. Meta's Segment Anything Model 2 extended promptable segmentation to video. In building its dataset, the authors report that annotation with the model in the loop was several times faster than a frame-by-frame approach, and that the model needed roughly a third as many human interactions as earlier methods to segment an object through a video.

These tools address the volume problem and the mechanical part of the identity problem. An annotator clicks once and the object is followed. When the tracker loses the object, drifts onto a neighbour or fails to re-identify it after an occlusion, a person has to notice and correct it. As in the general case discussed in the article on AI-assisted annotation, the human work shifts from drawing to supervising, and what is left is the hard part.

Tools do nothing for the other three problems. No tracker decides where “picking up” begins, which level of description to use, or whether a pedestrian intends to cross. Those remain matters of definition and judgement.

KEY TAKEAWAY

Automation has made the spatial part of video annotation cheap. The temporal part, deciding when things start, whether they are the same and what they mean, is still set by guidelines and people.

Is detailed temporal labelling still needed?

There is a serious argument that it matters less than it did. Large video-language models are trained on video paired with text, such as narration, subtitles or captions, with no frame-level labels. Self-supervised methods learn from the video itself by predicting masked or future content. If models can learn from raw video and loosely aligned text, perhaps precise temporal annotation is a legacy of an older approach.

For pretraining, that is largely right. General visual representations are increasingly learned without dense labels.

Three needs persist. Evaluation requires ground truth: to know whether a model correctly localised an event in time, someone must have marked where the event was. Fine-tuning for specific applications, such as detecting a defect on a production line, a fall in a care home or a safety violation on a site, requires examples of that specific event, precisely bounded. And the narrations and captions that weakly supervised models learn from are themselves annotations, whose quality and temporal alignment affect what is learned. Ego4D's narrations were written by people.

So the dense labels are needed in smaller quantity and for more specific purposes, which is the pattern seen across this series.

Designing a video annotation task

Several practices follow from treating time as the central difficulty.

Define events by observable anchors. “The action starts when the hand first contacts the object” can be applied consistently. “The action starts when the person begins” cannot. Each temporal label needs an operational definition.

Fix the granularity and give examples. State the level of description and show boundary cases: what counts as one action, and what as two.

Measure temporal agreement. Agreement on video should be assessed in time, using overlap between annotators' intervals, as well as on category. Low overlap on boundaries signals an underspecified definition.

Write identity rules. Specify what happens on occlusion, on exit and re-entry, and on merging and splitting. Without rules, each annotator invents their own.

Annotate in passes. Separating the tasks, objects first, then tracks, then events, then attributes, lets each pass have focused guidelines and suitable annotators.

Give annotators the context. Judgements about intent need the preceding and following seconds, sometimes audio, sometimes other camera views. An annotator shown an isolated clip is guessing.

Sample by event, not by time. Uniform sampling of footage yields mostly nothing happening. Use models or metadata to find candidate events and concentrate human effort there.

What this changes

FOR AI BUILDERS

Report temporal agreement between annotators, not only class agreement. Boundary noise limits how well a model can be measured.

Tolerate boundary uncertainty explicitly in metrics, or define anchors tight enough to remove it.

Audit tracker-assisted labels specifically at occlusions and crossings, where automated tracks fail.

FOR ENTERPRISES

When scoping a video AI project, specify the events to be detected in operational terms before any labelling. Vague event definitions are the main source of cost overrun.

Expect domain knowledge to matter. Recognising a procedural violation or a clinical event in video requires someone who knows the procedure.

Budget for the hard minutes, not the total hours. Most footage needs little annotation; a small share needs a lot.

THE INDIKA PERSPECTIVE

Video projects fail more often on definitions than on volume. The questions that decide the outcome are asked before labelling starts: what exactly is the event, where does it begin, what makes two detections the same object. Working those out with a client's domain experts, and testing them on a pilot until annotators agree in time as well as in category, is slower than starting to label and saves most of the rework.

RESEARCH & REFERENCES

Grauman, K. et al. (2022). "Ego4D: Around the World in 3,000 Hours of Egocentric Video." IEEE/CVF CVPR.

Kay, W. et al. (2017). "The Kinetics Human Action Video Dataset." arXiv preprint, DeepMind.

Goyal, R. et al. (2017). "The 'Something Something' Video Database for Learning and Evaluating Visual Common Sense." IEEE ICCV.

Gu, C. et al. (2018). "AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions." IEEE/CVF CVPR.

Sigurdsson, G., Russakovsky, O. and Gupta, A. (2017). "What Actions are Needed for Understanding Human Actions in Videos?" IEEE ICCV.

Moltisanti, D. et al. (2017). "Trespassing the Boundaries: Labeling Temporal Bounds for Object Interactions in Egocentric Video." IEEE ICCV.

Vondrick, C., Patterson, D. and Ramanan, D. (2013). "Efficiently Scaling up Crowdsourced Video Annotation." International Journal of Computer Vision.

Ravi, N. et al. (2025). "SAM 2: Segment Anything in Images and Videos." ICLR.

Alwassel, H. et al. (2018). "Diagnosing Error in Temporal Action Detectors." ECCV.

Machines That Move

Language models learned from text that already existed. Robots and autonomous vehicles have no such inheritance. Every example they learn from has to be made.

7 MINUTE READ

When researchers at Google set out to train a general-purpose robot controller, they needed examples of robots doing things. Nothing suitable existed. So they built the dataset: 13 robots, operated by people over 17 months in office kitchens, producing about 130,000 recorded episodes covering more than 700 tasks.

Compare that with the origins of a language model. Its training text was written over decades by billions of people for their own reasons, and was sitting on the internet waiting to be collected. The language-model developer's problem was gathering and filtering. The robot developer's problem is manufacturing.

That asymmetry explains much of why AI that writes has advanced faster than AI that acts.

There is no internet of actions

Text, images and video are by-products of human life, published in enormous quantities. The data a robot needs is different in kind. It needs records of what a body sensed, what it did in response, and what happened next: camera views from the robot's own position, joint angles, forces, and the commands that produced them.

Nobody generates that as a by-product. It has to be recorded deliberately, on physical hardware, usually by a person operating the robot remotely to demonstrate the task.

The published figures give a sense of the cost.

FIGURE 21 · DATA VISUALISATION

What it took to build datasets for machines that move

DATASET	CONTENTS	COLLECTION EFFORT REPORTED
RT-1 robot manipulation	About 130,000 episodes, more than 700 tasks	13 robots over 17 months
DROID robot manipulation	76,000 trajectories, about 350 hours, 564 scenes, 84 tasks	50 data collectors over 12 months, three continents
Open X-Embodiment pooled robot data	22 robot types, 527 skills	Datasets pooled from 21 institutions
nuScenes driving	1,000 scenes of 20 seconds; 1.4 million images; 1.4 million 3D boxes, 23 classes	6 cameras, 5 radars and 1 lidar per vehicle
Waymo Open Dataset driving	1,150 scenes of 20 seconds; about 12 million 3D boxes	Multi-sensor fleet recordings

Sources: Brohan et al. (2023), Khazatsky et al. (2024), Open X-Embodiment Collaboration (2024), Caesar et al. (2020), Sun et al. (2020).

DROID, a 2024 dataset built by a consortium of university and industry labs, took 50 data collectors working across three continents for 12 months to gather 76,000 demonstration trajectories, about 350 hours of robot interaction, across 564 scenes and 84 tasks. That is among the larger and more varied robot datasets in existence. It is 350 hours.

Open X-Embodiment, a collaboration of 21 institutions, pooled existing datasets from 22 different robot types into a common format, covering 527 skills. Its purpose was to test whether learning from many robots at once could help each of them, as learning from many languages helps a language model. The results suggested that it could. The exercise also showed how small and fragmented robot data is. A field-wide pooling of datasets from 21 institutions yielded a collection that would be unremarkable by the standards of text.

A language model may train on trillions of words. The leading robot datasets contain demonstrations numbered in the tens or hundreds of thousands, each of which took a person and a machine several minutes to produce.

Driving: plenty of data, not enough of the right kind

Autonomous vehicles look like the exception. Test fleets record continuously, and the leading developers have logged large distances. Raw sensor data is abundant.

The bottleneck there is of a different kind, and there are two parts to it.

Annotation is intensive. The public research datasets show what a labelled driving scene involves. nuScenes consists of 1,000 scenes of twenty seconds each, recorded with six cameras, five radars and a lidar unit. It contains 1.4 million

camera images and 1.4 million three-dimensional bounding boxes across 23 object classes. The Waymo Open Dataset contains 1,150 scenes, also twenty seconds each, with around 12 million three-dimensional boxes.

Each box is a cuboid placed in a lidar point cloud, consistent with what the cameras show, carrying a class, a size, an orientation and an identity maintained across time. The annotator is working in three dimensions, across several sensors that have to be reconciled, tracking every object through the scene. An earlier dataset, Cityscapes, reported that annotating a single street image with fine pixel-level labels took more than an hour and a half on average. Annotating a multi-sensor sequence is a larger task.

The events that matter are rare. Almost all driving is uneventful. A child running into the road, debris falling from a truck, an unusual vehicle, a traffic officer overriding a signal: these are the cases a system must handle and the ones a fleet almost never records. The distribution of driving situations has a very long tail, and safety depends on performance in the tail.

So the industry's practical problem is not obtaining data. It is finding, within vast recordings, the small fraction that is informative, and labelling it to a standard that supports safety claims. This is the active-learning setting described in an earlier article, at its most consequential.

What makes physical-world labels hard

Annotation for physical AI has features that do not arise in text or ordinary images.

Three dimensions. Labels describe position, extent and orientation in space. Errors of a few centimetres or degrees matter for a system that must not hit things.

Multiple sensors. Cameras, lidar and radar see differently and are mounted in different places. A label must be consistent across all of them, which depends on precise calibration and on annotators who understand what each sensor shows.

Time. Everything in the previous article about video applies. Objects must keep their identity. Their motion must be physically plausible.

Ambiguity of intent. Predicting what other road users will do, or whether a manipulation succeeded, requires judgement.

Success is not always visible. Did the robot place the object properly? Is the grasp stable? The answer may depend on forces and outcomes that a camera view does not settle.

Safety-critical evaluation. Labels used to validate a system's safety need a higher standard of accuracy and auditability than labels used for training, for the reasons given in the article on test sets, with greater consequences.

The simulation argument

The natural response to scarce real data is to generate it. Simulators can produce unlimited driving scenes and robot trajectories with perfect labels, including dangerous situations that could never be staged.

Simulation is indispensable, and widely used for testing as well as training. Domain randomisation, described in the article on synthetic data, has allowed policies trained in simulation to work on real hardware for a growing range of tasks.

Its limit is the one set out in that article. A simulator contains what its builders put in. It reproduces the situations they thought to model, with physics as accurate as they made it. The unanticipated situation, which is the definition of the long tail, is absent from the simulator for the same reason it is absent from the engineers' plans. Contact-rich manipulation, deformable objects, liquids and the behaviour of other people remain hard to simulate faithfully.

In practice, simulation and real data are used together. Real data grounds the simulator and reveals what it is missing. Simulation multiplies and varies what is known. Both need labelled real-world data as the reference.

The foundation-model argument

A second response is that large pretrained models will reduce the need for robot-specific data. Vision-language models trained on internet images and text already understand objects, scenes and instructions. Attach such a model to a robot and it may need only a little embodied data to learn to act.

There is evidence for this. Models that combine a pretrained vision-language backbone with robot data have shown better generalisation to new objects and instructions than models trained on robot data alone. Internet-scale pretraining does supply semantic knowledge: what a cup is, what “put it on the shelf” means.

It does not supply the mapping from that understanding to motor commands for a particular body, or the physical feel of contact, friction and weight. That part still has to come from embodied experience. The required amount may fall. The requirement does not go away, and as elsewhere in this series, less data that has to be more carefully chosen is a harder data problem, not an easier one.

KEY TAKEAWAY

For language, the data existed and the breakthrough was learning from it. For robots, learning methods have run ahead of the data. Progress depends on who can produce embodied data in quantity and variety.

What the data operation looks like

Producing data for physical AI is an industrial activity with several distinct parts.

Collection. Operating vehicles or tele-operating robots, with trained operators, in varied environments. Variety of scenes, objects and conditions matters more than hours in one setting, as the DROID authors emphasise.

Curation. Selecting the informative fraction from large recordings, using model uncertainty, disagreement between sensors or models, and triggers for rare events.

Annotation. Three-dimensional, multi-sensor, temporally consistent labelling, increasingly model-assisted, with people verifying and correcting.

Quality assurance. Geometric consistency checks, cross-sensor validation, and independent audit of safety-relevant labels.

Demonstration quality. For robot learning, the quality of the human demonstration is itself a property of the data. Hesitant or inconsistent demonstrations teach hesitant or inconsistent policies. Operator training and filtering of poor episodes are part of the pipeline.

Scenario description. Tagging recordings with structured descriptions of what happened, so that specific situations can be retrieved for training and testing.

What this changes

FOR AI BUILDERS

Prioritise diversity of environments and conditions over volume in one setting. Generalisation follows variety.

Build the mining pipeline early. The ability to find rare events in recorded data is worth more than the ability to record more.

Hold safety-validation labels to a separate, higher and independently audited standard.

FOR ENTERPRISES

In physical AI projects, expect data collection and labelling to dominate cost and schedule. Plan for it as the main workflow.

Site-specific data matters. A model trained elsewhere will need examples from your facility, equipment and conditions.

Ask how a supplier's system was tested on rare events, and where that test data came from.

THE INDIKA PERSPECTIVE

Physical AI extends the data layer from annotation to production. Someone has to operate the machines, design what they record, select what is worth keeping and label it in three dimensions and through time. For manufacturing, logistics, agriculture and mobility in India, the environments, vehicles, road behaviour and working practices differ enough from those in existing datasets that much of this data will have to be made locally.

RESEARCH & REFERENCES

- Brohan, A. et al. (2023). "RT-1: Robotics Transformer for Real-World Control at Scale." *Robotics: Science and Systems*.
- Khazatsky, A. et al. (2024). "DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset." *Robotics: Science and Systems*.
- Open X-Embodiment Collaboration (2024). "Open X-Embodiment: Robotic Learning Datasets and RT-X Models." *IEEE ICRA*.
- Caesar, H. et al. (2020). "nuScenes: A Multimodal Dataset for Autonomous Driving." *IEEE/CVF CVPR*.
- Sun, P. et al. (2020). "Scalability in Perception for Autonomous Driving: Waymo Open Dataset." *IEEE/CVF CVPR*.
- Cordts, M. et al. (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding." *IEEE CVPR*.
- Geiger, A., Lenz, P. and Urtasun, R. (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite." *IEEE CVPR*.
- Tobin, J. et al. (2017). "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World." *IEEE/RSJ IROS*.
- Brohan, A. et al. (2023). "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control." *Conference on Robot Learning*.

Cost per Label Is the Wrong Unit

Annotation is bought by the item. Its value is realised by the model. Almost every bad purchasing decision in AI data comes from confusing the two.

7 MINUTE READ

An AI team needs 100,000 documents classified. Two suppliers respond. The first quotes eight cents per document. The second quotes twenty.

On a spreadsheet, the decision is made: 8,000 dollars against 20,000. In most procurement processes, a difference of that size would need a strong justification to overcome, and “quality” is hard to put a number on. The cheaper supplier wins.

The team finds out what it bought about three months later, when the model trained on those labels underperforms in testing and somebody starts looking at the data.

The spreadsheet compared prices. The quantity that should have been compared is the cost of obtaining a correct, useful training example, and that depends on things the per-item price does not show.

What a label costs

Start with a definition. The effective cost of one usable label is the total spent, divided by the number of labels that turn out to be correct and useful.

Written out, the numerator contains more than the unit price. It includes the redundancy needed to reach acceptable confidence, the quality assurance needed to find the bad labels, the internal engineering time spent specifying, checking and cleaning, and the cost of rework. The denominator is not the number of items delivered. It is the number that are right.

FIGURE 22 · ANNOTATED EXAMPLE

Price per label and cost per correct label can rank suppliers differently

	SUPPLIER A	SUPPLIER B
Quoted price per item	\$0.08	\$0.20
Accuracy of a single label on this task	80%	96%
Labels bought per item	3, majority vote	1
Labelling spend, 100,000 items	\$24,000	\$20,000
Internal audit time at \$75 per hour	80 hours: \$6,000	20 hours: \$1,500
Total spend	\$30,000	\$21,500
Correct labels obtained	89,600	96,000
Cost per correct label	\$0.33	\$0.22

Hypothetical figures chosen to show the mechanism; not market data. Majority-vote accuracy assumes independent errors, which flatters Supplier A.

The table works through an illustration. The figures are hypothetical and chosen to show the mechanism. Supplier A's labels are right 80% of the time on this task, so the buyer triple-labels, takes a majority and spends engineering time auditing the result. Supplier B's are right 96% of the time with single labelling and a light audit.

On these assumptions the cheaper supplier costs more per correct label. The specific outcome depends on the numbers. The point is that the ranking can reverse, and the unit price contains no information about whether it will.

Sheng, Provost and Ipeirotis formalised part of this in 2008. They showed that when labels are noisy, a fixed budget can be better spent re-labelling existing items than acquiring new ones. The implication for pricing is direct: a noisy label is not a cheaper version of a good label. It is a fraction of one, and several must be bought to make a whole.

The cost that arrives later

Redundancy and rework are the visible part. The larger costs come after the labels have been accepted.

The model is worse than it should be. Label noise, particularly systematic noise, reduces accuracy. The team's response is usually to try more model work: a larger architecture, more tuning, more compute. Engineering time is spent compensating for a data problem that has not been diagnosed as one.

The evaluation is wrong. If test labels come from the same source, the model's measured performance is unreliable, as the article on label errors described. Decisions about what to ship rest on a faulty instrument.

The failure appears in production. The cascades research documented how data problems surface late and compound. By the time one is traced to its origin, the cost includes re-collection, re-labelling, re-training, re-validation and whatever harm occurred in between. In Sambasivan's study, practitioners described abandoning entire datasets.

The engineering team becomes the quality department. Sculley and colleagues' analysis of technical debt in machine-learning systems noted that the model code is a small part of a real system, surrounded by data collection, verification and monitoring. When purchased data is unreliable, expensive engineers end up doing data cleaning, the most costly way to do it.

None of these appears on the invoice. All of them are caused by what was on it.

Why low prices are low

It helps to ask how a supplier arrives at a very low unit price. Annotation cost is mostly labour. There are a limited number of ways to reduce it.

Pay people less. Research on crowd work has documented the result. An analysis of millions of tasks on Amazon Mechanical Turk by Hara and colleagues estimated median hourly earnings of about two dollars, with unpaid time spent searching for tasks and on rejected work accounting for much of the shortfall. Ethnographic studies, including Gray and Suri's book-length account, describe the instability this creates. Low pay and high turnover mean annotators do not accumulate experience on a task, and experience is where quality comes from.

Give them less time. A per-item price implies a time per item. If it is too short for the judgement the task requires, annotators will take the shortcut the task allows.

Reduce oversight. Quality assurance is overhead. So are training, guideline development, pilot rounds and feedback to annotators.

Use automation quietly. Pre-labelling with a model and light human confirmation lowers cost considerably. As discussed earlier in this series, it also changes the error profile from random to systematic unless verified carefully.

Each of these is a reasonable way to reach a price. Each reduces yield on tasks that require care. The buyer who selects on price is selecting for them.

KEY TAKEAWAY

A per-label price tells you what you will pay. Only yield, redundancy and rework tell you what a correct label will cost, and those are properties of the task and the supplier together.

When cheap is correct

The argument should not be read as “always pay more”. For a large class of tasks the low-price option is the right one.

Where a task is simple, well specified and tolerant of random error, non-expert labels at low cost with modest redundancy perform well. Snow and colleagues’ result, that four inexpensive labels matched an expert for certain language tasks, has been replicated many times. Paying a premium for such work buys nothing.

Where volume matters more than precision, as in pretraining or weak supervision, noisy labels at scale can be the better use of a budget. The article on data volume set out the conditions.

And where a model can do the first pass reliably, verified automation is cheaper than either supplier.

The error is to apply one purchasing logic to all tasks. The useful question is what the task requires, and it is answered by measurement.

How active learning and tiering change the sum

Two techniques from earlier articles shift the economics in ways per-label pricing obscures.

Active learning reduces the number of labels needed by directing effort to informative examples. Those examples are harder, so each costs more. A supplier quoting a flat rate will lose money on an actively selected set, or will cut corners. The total cost falls while the unit cost rises, and a buyer focused on unit cost will see only the second.

Tiering directs easy items to generalists and hard items to experts. The blended price is higher than an all-generalist rate and far lower than an all-expert one. Its value depends on the routing being accurate, which is a capability to pay for.

In both cases the efficient design has a higher price per item and a lower cost per unit of model improvement.

A better way to buy

Several changes to procurement practice follow.

Run a paid pilot on your own task. Give each candidate supplier the same few hundred items, with a subset for which you hold expert-verified answers. Measure accuracy against them, by category. This costs little and produces the one number the price list omits.

Compute effective cost. With measured accuracy, estimate the redundancy and audit effort needed to reach your quality target with each supplier. Compare totals.

Price by difficulty. Agree different rates for different item classes. A flat rate rewards speed on hard items, which is the opposite of what you want.

Contract on measured quality. Specify accuracy against an agreed reference set, with defined sampling, in place of or alongside a price. Include what happens when it is missed.

Ask how the price is achieved. Annotator pay, training, tenure, time allowed per item, the extent of automation and how it is verified. These are the inputs to yield.

Account for internal time. Hours your engineers spend writing guidelines, answering questions and cleaning output are part of the cost of the data. A supplier who needs less of them is cheaper than the invoice shows.

Value evaluation data separately. A test set is small and its errors are costly. Apply a different standard and a different budget.

What this changes

FOR AI BUILDERS

Track model performance per labelling dollar, not labels per dollar. The first is what the budget is for.

Keep a verified reference set for every labelling task and score incoming batches against it.

Record the time your own team spends on data quality. It is usually the largest hidden cost.

FOR ENTERPRISES

Do not let procurement compare annotation suppliers on unit price alone. Require a pilot with measured accuracy on your task.

Expect to pay different rates for different difficulty levels, and treat a single flat rate for complex work as a warning sign.

Ask suppliers about workforce pay, tenure and training. They are leading indicators of quality.

THE INDIKA PERSPECTIVE

Selling annotation by the item invites comparison on a figure that says little about value. The more honest commercial conversation starts from the outcome: what accuracy the model needs, on which cases, verified how. From there the mix of generalist, expert and automated work can be designed, and the price follows from the design. That conversation is harder to have than quoting a rate, and it is the one that aligns what the buyer pays for with what the buyer needs.

RESEARCH & REFERENCES

- Sheng, V., Provost, F. and Ipeirotis, P. (2008). "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers." ACM KDD.
- Sambasivan, N. et al. (2021). "Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI." ACM CHI.
- Sculley, D. et al. (2015). "Hidden Technical Debt in Machine Learning Systems." NeurIPS.
- Northcutt, C., Athalye, A. and Mueller, J. (2021). "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks." NeurIPS Datasets and Benchmarks.
- Hara, K. et al. (2018). "A Data-Driven Analysis of Workers' Earnings on Amazon Mechanical Turk." ACM CHI.
- Gray, M. and Suri, S. (2019). Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass. Houghton Mifflin Harcourt.
- Snow, R. et al. (2008). "Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks." EMNLP.
- Settles, B. (2009). "Active Learning Literature Survey." Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Build, Buy or Partner

Whether to run data operations in-house or outside is usually argued on cost. A fifty-year-old theory of the firm gives a better answer, and it depends on the kind of data.

8 MINUTE READ

In June 2025, Meta agreed to invest about 14.3 billion dollars for a 49% stake in Scale AI, a company whose business was supplying labelled data and human feedback to AI developers. Scale's chief executive joined Meta. Within days, according to Reuters and other outlets, several of Scale's largest customers, themselves competitors of Meta, began moving work elsewhere.

Their concern was easy to understand. A data supplier sees what a model developer is working on. It learns which capabilities the developer is trying to improve, which failures it is trying to fix, how it defines quality and what its evaluation looks like. That information is strategically sensitive, and it had just come under the partial ownership of a rival.

The episode was unusual in scale. The underlying question is one every organisation building with AI has to answer. Which data work should be done inside the company, which bought from a supplier, and on what terms?

A theory that fits

The standard economic account of why firms make some things and buy others was developed by Oliver Williamson, building on Ronald Coase, and earned Williamson a Nobel prize. Its central idea is that using a market has costs beyond the price: the cost of specifying what you want, of checking that you received it, and of protecting yourself if the other party behaves opportunistically.

Williamson identified the conditions under which those costs become high enough that it is better to bring the activity inside. Two matter most here.

Asset specificity. How far does the work depend on knowledge, tools or relationships particular to this buyer? A highly specific investment has little value in any other use, which leaves each side dependent on the other.

Measurement difficulty. How easily can the buyer verify the quality of what was delivered? Where quality is hard to observe, a supplier has room to cut corners, and a contract cannot fully prevent it.

When both are low, the market works well. When both are high, firms tend to integrate, or to form long-term relationships that sit between a market and a hierarchy.

AI data spans the whole range, and that explains why one answer does not fit.

FIGURE 23 · FRAMEWORK

Sourcing data work by specificity and verifiability

	QUALITY IS EASY TO VERIFY	QUALITY IS HARD TO VERIFY
GENERIC any buyer could use it	Buy competitively Boxes, clear transcription. A commodity; automate where possible.	Buy the process General preference ranking and quality rating. Select on method, reputation and audit rights.
SPECIFIC depends on your knowledge	Buy with a specification Extraction from your own formats. Invest in transfer; guard against lock-in.	Keep control, partner for capacity Domain evaluation, expert feedback, product red teaming. Your experts define quality.

Framework by the editors, applying Williamson (1979) to AI data work.

Four kinds of data work

Generic and easy to verify. Bounding boxes around cars. Transcription of clear audio in a major language. The task is the same for every buyer, guidelines are standard and correctness can be checked by sampling. This is a commodity. Buy it, competitively, and expect the price to keep falling as automation absorbs it.

Generic and hard to verify. Preference ranking of general-purpose assistant responses. Open-ended quality ratings. Any buyer could use the output, but whether it is good is hard to tell from the output alone. Here the supplier's process, its recruitment, training and quality control, is what is being bought, and reputation and audit rights matter more than price.

Specific and easy to verify. Extracting fields from a company's own invoice formats. The task is particular to the buyer, but a correct answer is unambiguous. This can be bought, with an investment in transferring the specification. Lock-in is the risk: the supplier accumulates knowledge of the formats that the buyer would have to rebuild elsewhere.

Specific and hard to verify. Judging whether an underwriting assistant's recommendation is sound by the insurer's own standards. Evaluating a clinical summarisation model against one hospital's practice. Red-teaming a product for the ways its particular users might be harmed. The knowledge needed is inside the organisation, and so is the ability to tell good work from bad.

The fourth quadrant is where arm's-length purchasing breaks down, and it is where most of the value in applied AI now lies.

What the research says about the boundary

Studies of annotation as a working practice support the theory.

Miceli, Schuessler and Yang conducted fieldwork inside annotation companies and found that the meaning of the labels was largely set by the client. Annotators applied categories they had been given, often without the context to question them, and client instructions were treated as ground truth even when annotators could see they fitted the data poorly. The authors' title describes what they observed: interpretation was imposed downward. Knowledge that annotators had about the data did not flow back up to those who designed the scheme.

Wang, Prabhakaran and Díaz interviewed annotation workers in India and reached a related conclusion. Annotators were treated as interchangeable executors of a specification, while often holding knowledge that would have improved it.

Sambasivan's data-cascades study listed poor documentation across organisational boundaries among its four main triggers. Assumptions made on one side of a handover were invisible on the other.

These findings point in the same direction. The organisational boundary between whoever understands the purpose of the data and whoever produces it is where information is lost. A pure outsourcing arrangement maximises that boundary.

The case for building

The argument for doing data work internally follows directly.

Knowledge stays close to the work. The people defining quality are near the people labelling. Guidelines change as understanding improves, without contract variations.

Confidentiality. Sensitive data, and the sensitive knowledge of what the organisation is trying to improve, never leaves. The 2023 reports of Samsung employees pasting proprietary code into a public chatbot were a reminder of how easily information escapes through tools, and a supplier's workforce is a larger surface.

Accumulated expertise. Annotators who stay on a domain become skilled in it. That is an asset, and internally it belongs to the organisation.

Feedback. Problems found in labelling inform the product. Production failures inform the labelling.

The case against building

The costs are equally real.

It is a different business. Recruiting, training, scheduling and managing an annotation workforce, building tooling and running quality systems is an operational discipline. Most AI teams are not set up for it, and those that try tend to underinvest and produce the low-status, under-resourced data function that Sambasivan's interviewees described.

Demand is uneven. Data needs arrive in bursts, around a new model, a new product, a new language. A permanent team sized for peaks is idle between them. One sized for the average is overwhelmed.

Breadth is hard to hold. A project may need radiologists this quarter, Tamil speakers next quarter and security researchers after that. No single employer can retain all of them.

Tooling and method move quickly. Specialists who work across many clients see more problems and more ways of solving them.

A team that sends its domain experts to label data full-time has also made an expensive choice about how to use its scarcest people.

KEY TAKEAWAY

The question is not who labels the data. It is where the knowledge of what the data should be resides, and whether the arrangement keeps that knowledge in contact with the people doing the work.

The intermediate form

Williamson's theory predicts that when specificity and measurement difficulty are high but full integration is impractical, firms form long-term, relationship-based arrangements with safeguards. That is a fair description of what works for data.

In such a partnership the client keeps certain things. It owns the definition of quality: the taxonomy, the rubric, the evaluation set. It supplies domain experts for the decisions only they can make, such as guideline design, adjudication of hard cases and the gold standard. It retains the data and the rights to it.

The partner brings what is difficult to build internally. A trained, managed and stable workforce, with access to specialists as needed. Tooling and quality systems. Method: how to pilot a guideline, measure agreement, route by difficulty and counter automation bias. Capacity that scales up and down.

What distinguishes this from outsourcing is the interface. Annotators' questions and disagreements reach the client's experts and change the guidelines. The partner's team stays on the account long enough to learn the domain. Quality is measured jointly against a reference both sides trust.

Safeguards worth insisting on

The Scale episode suggests several protections that have become more important.

Independence. Who owns the supplier, and who else does it serve? A supplier controlled by a competitor, or by a model developer with its own interests, sees information it may have reason to use.

Data rights. The contract should state that the client owns the data, the labels and the guidelines, that none of it is used to train the supplier's or anyone else's models, and that it is deleted or returned at the end.

Workforce stability and separation. Named, vetted teams, not an anonymous crowd. Separation from teams serving competitors. Controls on the use of external AI tools.

Security. Access controls, secure environments for sensitive data, and audit logs. For evaluation data, protection against leakage into any training set.

Portability. Guidelines, gold sets and annotator performance data should be the client's and in a form that could be taken elsewhere. This is the defence against lock-in.

Transparency of process. The right to see who the annotators are in aggregate, how they were trained, how agreement and accuracy are measured, and how much of the work is automated.

Deciding for a given workload

A short set of questions places most data work.

How specific is the knowledge required? If any competent annotator could do it from a written guideline, it is generic. If it needs someone who knows your products, customers or standards, it is specific.

How would you know if it were done badly? If sampling and checking would reveal it, verification is easy. If it would surface only in model behaviour months later, it is hard.

How sensitive is it? Consider what the data, and the task itself, reveal about your strategy.

How steady is the demand? Continuous work can justify a standing team. Bursts favour external capacity.

Generic, verifiable, non-sensitive work should be bought on price and automated where possible. Specific, hard-to-verify, sensitive work should be kept close, with internal experts in control and a partner providing capacity and method under strong safeguards. Most organisations will have some of each and should manage them differently.

What this changes

FOR AI BUILDERS

Keep ownership of guidelines, rubrics and evaluation sets inside your team regardless of who does the labelling.

Build a channel through which annotator questions and disagreements reach whoever can change the specification.

Avoid sending evaluation data through the same pipeline, people or tools as training data.

FOR ENTERPRISES

Segment data work by specificity and verifiability before deciding how to source it. A single sourcing policy will be wrong for part of it.

Examine a data supplier's ownership, client base and data-use terms as carefully as you would a cloud provider's.

Budget your own experts' time for guideline design and adjudication. No external arrangement works without it.

THE INDIKA PERSPECTIVE

The role a data partner should aim for is the intermediate one: close enough to the client's domain to carry its standards, independent enough to be trusted with its data and plans. That means stable teams that learn an account, a working interface with the client's experts, clear terms that the data and the specifications belong to the client, and no conflict of interest in how the work could be used. Independence, in particular, has become something buyers ask about first.

RESEARCH & REFERENCES

Williamson, O. (1979). "Transaction-Cost Economics: The Governance of Contractual Relations." *Journal of Law and Economics*.

Coase, R. (1937). "The Nature of the Firm." *Economica*.

Miceli, M., Schuessler, M. and Yang, T. (2020). "Between Subjectivity and Imposition: Power Dynamics in Data Annotation for Computer Vision." *Proceedings of the ACM on Human-Computer Interaction (CSCW)*.

Wang, D., Prabhakaran, V. and Díaz, M. (2022). "Whose AI Dream? In search of the aspiration in data annotation." *ACM CHI*.

Sambasivan, N. et al. (2021). "Everyone wants to do the model work, not the data work: Data Cascades in High-Stakes AI." *ACM CHI*.

Miceli, M., Posada, J. and Yang, T. (2022). "Studying Up Machine Learning Data: Why Talk About Bias When We Mean Power?" *Proceedings of the ACM on Human-Computer Interaction (GROUP)*.

Reuters (2025). Reporting on Meta's investment in Scale AI and subsequent customer responses, June.

Bloomberg (2023). Reporting on Samsung's restriction of generative AI tools following an internal data leak, May.

Sculley, D. et al. (2015). "Hidden Technical Debt in Machine Learning Systems." NeurIPS.

Where Did This Data Come From?

For most datasets used to train AI, nobody can fully answer that question. Courts, regulators and the mathematics of model training are all beginning to require an answer.

8 MINUTE READ

A machine-learning engineer downloads a dataset from a popular hosting platform. The page says it is released under a permissive licence that allows commercial use. She fine-tunes a model on it and the model goes into a product.

How confident should she be in that licence?

In 2024, a group of researchers calling themselves the Data Provenance Initiative published an audit of more than 1,800 text datasets widely used for fine-tuning. They traced each one back to its original source and compared the licence its creators had specified with the licence shown on the hosting platforms where people find it.

On those platforms, licence information was missing for more than 70% of datasets. Where a licence was shown, it was miscategorised more than half the time, and the errors usually ran in one direction: the platform listed terms more permissive than the creators intended.

The engineer's confidence should be low. Her situation is ordinary. Datasets are assembled from other datasets, repackaged, filtered, translated and re-uploaded, and at each step the record of where the content came from, and on what terms, tends to fall away.

What provenance means

Provenance is the documented history of a piece of data. For AI training data, it covers at least five things.

FIGURE 24 · FRAMEWORK

What a provenance record has to answer

ELEMENT	THE QUESTION	WHY IT IS ASKED
Source	Who created the content, when and for what purpose?	Relevance, bias and the date of consent
Rights and consent	Under what licence or terms? Is personal data involved, and on what basis?	Copyright and data-protection exposure
Collection method	Collected directly, licensed, scraped, purchased or generated?	Determines what else must be checked
Processing and labelling	What was filtered or altered? Who annotated it, how, with what agreement and what AI assistance?	What is missing; how far labels can be trusted
Lineage	What was it derived from, and which datasets and models derive from it?	Tracing a fault, a withdrawal or a claim

Synthesis by the editors, after Gebru et al. (2021), Bender and Friedman (2018) and Longpre et al. (2024).

For most of the history of machine learning, hardly any of this was recorded. Geiger and colleagues reviewed a sample of machine-learning papers that used human-labelled data and found that many did not report basic facts such as who the annotators were, whether they were trained, or how agreement was measured. A widely cited proposal, datasheets for datasets, was written in response to this gap. It took its inspiration from the electronics industry, where every component ships with a document describing its characteristics and limits.

The absence of provenance has not been a theoretical concern.

Three datasets that came apart

LAION-5B is a collection of nearly six billion image–text pairs scraped from the web, used to train widely deployed image generators. In late 2023, an investigation by the Stanford Internet Observatory found that it contained links to more than a thousand instances of child sexual abuse material. The dataset was taken offline and later re-released after cleaning. Earlier academic work by Birhane and colleagues had already documented other harmful content in its predecessor. Nobody had reviewed the contents, because at that scale nobody could.

Retracted datasets continue to circulate. Peng and colleagues studied what happened to several face and person-recognition datasets after their creators withdrew them over consent and privacy concerns. The datasets, and derivatives of them, kept being used and cited in research for years. Withdrawal by the original source did not reach the copies.

The open web is closing. A second study from the Data Provenance Initiative audited 14,000 web domains behind major training corpora and documented a sharp rise in restrictions on AI crawlers through robots.txt files and terms of service. Content that was freely collected a few years ago is now explicitly withheld. Data scraped before the restriction and data scraped after it carry different positions on consent, and a dataset without dates cannot say which it contains.

Three pressures converging

Courts. Litigation over the use of copyrighted material in training has moved from filing to rulings. In the United States in 2025, a federal court in the case of Thomson Reuters against Ross Intelligence rejected a fair-use defence for the use of legal headnotes to train a competing legal research tool. In a case brought by authors against Anthropic, a federal judge held that training on lawfully acquired books was fair use, and drew a distinction for copies obtained from pirate libraries; the case later settled for a reported 1.5 billion dollars. In a case against Meta, another judge found for the company on the record before him while expressing doubt about the general proposition. The United States Copyright Office published a lengthy report on training and copyright in the same year.

The law is unsettled and differs between jurisdictions, and nothing here is legal advice. One practical point is common to the rulings. The outcome turned on facts about the data: how it was obtained, from where, and what was done with it. An organisation that cannot establish those facts is poorly placed in any dispute.

Regulators. The European Union's AI Act imposes data governance obligations on high-risk systems under Article 10, including requirements about the origin of data, how it was collected and prepared, and examination for bias. For general-purpose models, providers must publish a summary of the content used for training, following a template the Commission issued in 2025. In India, the Digital Personal Data Protection Act of 2023 sets conditions on processing personal data, with implementing rules notified in 2025. The United States NIST AI Risk Management Framework, though voluntary, treats documentation of data provenance as part of responsible practice.

Each of these asks organisations to state where training data came from. That cannot be reconstructed after the fact.

The technical need. As the article on model collapse argued, the effect of synthetic data on a model depends on the proportion of real data in the mix. Managing that proportion requires knowing which records are human-made. Detection after the fact is unreliable. The only dependable method is to record origin when the data is created or acquired.

Provenance is also what makes debugging possible. When a model misbehaves, the question is which data taught it that. Without lineage, the question has no answer.

KEY TAKEAWAY

Provenance cannot be added later. A dataset's history is known at the moment it is created or acquired and is progressively lost after that. Recording it is cheap at the time and often impossible afterwards.

The cost of caring

There is a serious argument that strict provenance requirements do more harm than good.

Much of the progress in AI came from large, loosely documented, scraped datasets. Requiring full provenance would exclude most of them. It would favour large companies that can afford licensing and compliance over researchers and start-ups. And it would exclude data from communities and languages that are under-represented, where formal licensing is rare, making models less inclusive.

These are real costs. The Data Provenance Initiative's own authors note that restrictions on crawling fall unevenly and may skew what models learn.

Two responses are available. First, provenance is not the same as permission. Recording where data came from does not by itself prevent its use. It enables an informed decision about risk. An organisation may choose to use data of uncertain status for research and not for a commercial product, and it can only make that distinction if it knows which data is which.

Second, the balance of risk depends on the use. For a research prototype, the consequences of poor provenance are small. For a product deployed to customers in a regulated industry, they include injunctions, forced retraining and regulatory penalties. The appropriate standard varies accordingly.

What a provenance record contains

Several documentation standards exist: datasheets for datasets, data statements for language data, data cards, and the dataset nutrition label. They differ in format and agree on substance. A usable record answers the following.

Origin. Where did each part of the data come from? Who created it, when, and for what purpose?

Acquisition. How was it obtained: collected directly, licensed, scraped, purchased, generated?

Rights. Under what licence or terms? What uses are permitted? Does it contain personal data, and on what basis was it collected?

Composition. What is in it? Which populations, languages and time periods are represented? What is known to be missing?

Processing. What filtering, cleaning, deduplication, translation or augmentation was applied?

Labelling. Who annotated it, with what qualifications, under what instructions, with what agreement and quality control? Was AI assistance used?

Synthetic content. Which records, if any, were machine-generated, by which model?

Lineage. Which other datasets were derived from this one, and which models were trained on it?

For commissioned data, all of this is available at the time of creation if somebody thinks to write it down. For acquired data, it has to be requested, and the supplier's ability to provide it is a good indicator of the supplier.

Building it into the pipeline

Provenance documentation written once as a compliance exercise goes stale. Durable practice treats it as metadata carried with the data.

Record at ingestion. Every dataset entering the organisation gets a record of source, date, terms and contact before it is used.

Tag at the record level where it matters. Source, licence, date and human-or-synthetic origin, attached to individual records and kept through transformations.

Version datasets. A training run should reference a specific, immutable version, so that a model's training data can be stated exactly.

Keep the link from model to data. For each model, which datasets and versions went into it. This is what a regulator, a litigant or an internal investigation will ask for.

Plan for removal. If a source must be withdrawn, because consent is revoked or a licence disputed, can you identify the affected records and the models trained on them?

Require it from suppliers. Make provenance documentation a contractual deliverable for any purchased or commissioned data.

What this changes

FOR AI BUILDERS

Verify licences at the original source. Hosting-platform metadata is wrong often enough that it should not be relied on.

Carry source, date, licence and synthetic-origin fields on every training record through every pipeline stage.

Maintain a model-to-dataset lineage map that can be produced on request.

FOR ENTERPRISES

Ask of every AI system in use: what was it trained on, and can the supplier document it? Treat the absence of an answer as a risk.

Give commissioned, fully documented data a value above its relevance alone. Clear provenance removes a class of legal and regulatory exposure.

Align data documentation with the disclosure rules that apply to you, including the EU AI Act where relevant and India's data protection law.

THE INDIKA PERSPECTIVE

Commissioned data has an advantage that is easy to overlook: its whole history can be known. Who produced it, under what consent, with what instructions and tools, on which dates. A data partner can deliver that record with the data, as a matter of course. For a buyer facing regulators, auditors or opposing counsel, the documentation may be worth as much as the dataset.

RESEARCH & REFERENCES

Longpre, S. et al. (2024). "A large-scale audit of dataset licensing and attribution in AI." Nature Machine Intelligence.

Longpre, S. et al. (2024). "Consent in Crisis: The Rapid Decline of the AI Data Commons." NeurIPS Datasets and Benchmarks.

Geiger, R. S. et al. (2020). "Garbage In, Garbage Out? Do Machine Learning Application Papers in Social Computing Report Where Human-Labeled Training Data Comes From?" ACM FAT*.

Gebru, T. et al. (2021). "Datasheets for Datasets." Communications of the ACM.

Bender, E. and Friedman, B. (2018). "Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science." Transactions of the Association for Computational Linguistics.

Thiel, D. (2023). "Identifying and Eliminating CSAM in Generative ML Training Data and Models." Stanford Internet Observatory.

Birhane, A., Prabhu, V. and Kahembwe, E. (2021). "Multimodal datasets: misogyny, pornography, and malignant stereotypes." arXiv preprint.

Peng, K., Mathur, A. and Narayanan, A. (2021). "Mitigating Dataset Harms Requires Stewardship: Lessons from 1000 Papers." NeurIPS Datasets and Benchmarks.

European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act), Articles 10 and 53.

United States Copyright Office (2025). Copyright and Artificial Intelligence, Part 3: Generative AI Training. Pre-publication version.

Thomson Reuters Enterprise Centre GmbH v. Ross Intelligence Inc. (D. Del. 2025); Bartz v. Anthropic PBC (N.D. Cal. 2025); Kadrey v. Meta Platforms, Inc. (N.D. Cal. 2025).

Government of India (2023). The Digital Personal Data Protection Act, 2023.

NIST (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0).

Not a Translation Problem

The obvious way to build AI for India's languages is to translate English datasets. The evidence on what that produces explains why Indian-language AI is a data-creation problem.

7 MINUTE READ

When a team at Google set out to build a question-answering benchmark covering eleven typologically diverse languages, they made an unusual decision. They would not translate anything.

The standard method for creating a multilingual dataset was, and is, to take an English one and translate it. It is fast and cheap and gives parallel data across languages. The authors of the benchmark, called TyDi QA, rejected it on principle. Translated text, they argued, is not the same as text written in a language. It carries the structure and concerns of the source. A question translated from English is a question an English speaker wanted answered, phrased the way English phrases it. So they had native speakers write questions from scratch in each language about things those speakers wanted to know.

The phenomenon they were avoiding has a name, translationese, and its effects on models are measurable. Artetxe and colleagues showed that translation leaves artefacts that models pick up on, and that evaluating on translated test sets can give a distorted picture of how well a model handles the language as written by its speakers.

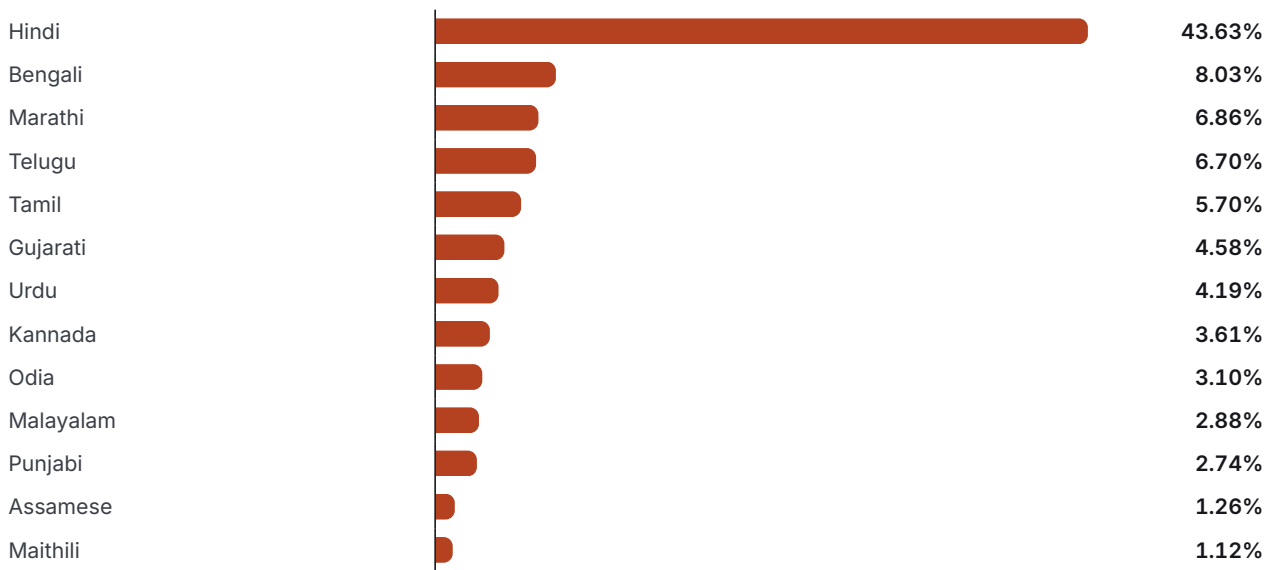
For India, the point carries particular weight, because of how far the starting position is from English.

The shape of the problem

The 2011 Census of India recorded 121 languages spoken by ten thousand or more people. Twenty-two are listed in the Eighth Schedule of the Constitution, and together they are the mother tongue of 96.71% of the population. Behind those figures the census counted 270 identifiable mother tongues with ten thousand or more speakers, grouped from more than 19,500 raw responses.

FIGURE 25 · DATA VISUALISATION

Mother tongue of India's population, by scheduled language



Share of total population, thirteen largest of the 22 scheduled languages. Source: Census of India 2011, Language (Paper 1 of 2018). The census groups many mother tongues under "Hindi".

Hindi, by the census's definition, is the mother tongue of 43.63% of the population. Six further languages each account for more than 4%. The remainder is a long tail of languages with tens of millions of speakers each, more than most European languages have.

Speaker numbers bear little relation to digital resources. Joshi and colleagues at Microsoft Research classified the world's languages into six tiers by the amount of labelled and unlabelled data available. English and a handful of others sat in the top tier. Hindi was placed a tier below, and most other Indian languages lower still, despite speaker populations in the tens of millions. A language can be among the twenty most spoken in the world and still be low-resource for AI.

What translation cannot supply

Content. A translated English corpus is about what English text is about. It will not contain the names of Indian places, institutions, foods, festivals, government schemes, kinship terms or legal concepts in the proportions in which they occur in real use. A model trained on it will be fluent on translated encyclopaedia content and thin on what users ask about.

Script. India's scheduled languages are written in more than ten different scripts. Some languages use more than one. And a great deal of everyday digital text is not in the native script at all: it is typed in Latin characters, with spelling improvised by each writer. A user may write the same Hindi word three ways in one

conversation. The AI4Bharat group built Aksharantar, a large transliteration resource, specifically because romanised input is how many people type. Translation of English produces clean native-script text and none of this.

Code-mixing. Multilingual speakers switch languages within a sentence, as a normal way of talking. Hindi and English, Tamil and English, Bengali and English, combined in a single utterance. Benchmarks built for code-mixed text, such as GLUECoS, found that multilingual models handled code-mixed text less well than text in a single language, and improved when exposed to code-mixed data. No monolingual source corpus contains it, so translating one cannot produce it.

Register and dialect. The formal standard of a language, as found in news and government documents, can differ considerably from how it is spoken in a given district. Translation systems output the standard. Users do not always write in it.

Context. Bhatt and colleagues examined what fairness evaluation would need to cover in India and showed that the relevant social categories differ from those in benchmarks developed in the West. Caste, religion and region carry a weight that an imported benchmark does not test. A dataset translated from English inherits the source's assumptions about which distinctions matter.

Translation of the web by the web

There is a further difficulty. The alternative to translating datasets, collecting text from the web, is itself partly contaminated by translation.

As noted in the article on model collapse, Thompson and colleagues found that a large share of multi-way parallel content on the web appears to be machine-translated, and that this is concentrated in lower-resource languages. For a language with limited original web content, a scraped corpus may consist largely of machine output of uncertain quality. Training on it means learning from an earlier model's errors.

Scraped data in major languages can be assumed to be mostly human-written, at least if it predates recent language models. For many Indian languages, that assumption is weaker.

What has been built

The response from Indian research institutions has been to create data, and the record over the past several years is substantial.

AI4Bharat, based at IIT Madras, has released a series of open resources. Samanantar assembled 49.7 million sentence pairs between English and eleven Indian languages. IndicTrans2 extended translation to all 22 scheduled languages, supported by a parallel corpus of 230 million sentence pairs, and the team created

644,000 manually translated sentence pairs as part of that work, along with what they describe as the first benchmark with parallel text across all 22. IndicCorp and related efforts gathered monolingual text.

The Government of India's Bhashini mission, launched in 2022 under the National Language Translation Mission, funds language technology and data creation for Indian languages and runs a crowdsourcing initiative, Bhasha Daan, through which the public can contribute text and speech. The IndiaAI Mission, approved in 2024 with an outlay of a little over ten thousand crore rupees, includes a datasets platform among its components.

Two features of this work are instructive. Even translation models needed human translators: the 644,000 manually produced pairs were made because mined and machine-aligned data was not sufficient for training and was unsuitable for evaluation. And evaluation sets were built separately and carefully, because, as IndicGenBench and similar efforts recognise, without human-created test data in each language there is no way to know how well anything works.

KEY TAKEAWAY

Translation moves English content into Indian scripts. It does not produce the content, spellings, mixtures and registers that speakers of Indian languages use, and those are what a model needs to have seen.

The argument for translating anyway

Translation should not be dismissed. It has real uses and the pragmatic case is strong.

It is far cheaper than original collection. For instruction-following and reasoning data, where the skill being taught is largely independent of language, translated data transfers reasonably well. Large multilingual models show cross-lingual transfer: abilities learned in English partly carry over to other languages without any data in them. And machine translation for major Indian languages has improved to the point where its output is usable for many purposes.

A reasonable position is that translation is a good way to obtain volume for general capabilities and a poor way to obtain coverage of what is specific to a language and its speakers. The two are complementary.

The mistake is to treat a translated dataset as having solved the problem, and especially to evaluate on translated data. A model tested on translated English questions will be scored on its handling of translated English questions.

What original data collection involves

Creating data in a language, as distinct from into it, has practical requirements.

Native speakers as authors. Prompts, questions and documents written by people who use the language daily, about subjects they choose.

Regional and social spread. Hindi in Bihar is different from Hindi in Rajasthan. A language's data should be collected across the places and communities where it is spoken.

Real input forms. Native script, romanised text and code-mixed text, in proportions that reflect how people type.

Domain coverage. Agriculture, health, finance, law and government services, in the vocabulary used locally.

Native-speaker evaluation. Test sets written and verified by speakers, never translated, with the same care this series has argued for throughout.

Annotators who share the context. Judging whether a response is appropriate, offensive or accurate in Marathi requires a Marathi speaker who knows the setting.

The last point connects India's linguistic situation to its workforce. The country has large numbers of educated native speakers of every scheduled language, many of them multilingual. That is the input that cannot be imported or synthesised.

What this changes

FOR AI BUILDERS

Never evaluate an Indian-language system only on translated test data. Build or obtain a native test set first.

Include romanised and code-mixed inputs in training and evaluation in realistic proportions.

Check scraped corpora in lower-resource languages for machine-translated content before training on them.

FOR ENTERPRISES

Do not assume a model's English performance predicts its performance in Hindi, and do not assume Hindi predicts Odia or Assamese. Test each language you serve.

Budget for original data creation in your domain and languages. Open datasets cover general language; your products, terms and customers are not in them.

Involve native speakers from your actual customer regions in defining quality.

THE INDIKA PERSPECTIVE

India's language diversity is usually described as an obstacle to AI. For data, it is also a capability: there are few places with native speakers of so many major languages, across so many professions, in one labour market. Turning that into datasets is organisational work, recruiting by language and region, writing guidelines that respect how each language is used, and building native evaluation sets. It is the layer that open models and translated corpora leave unfinished.

RESEARCH & REFERENCES

- Clark, J. et al. (2020). "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages." Transactions of the Association for Computational Linguistics.
- Artetxe, M., Labaka, G. and Agirre, E. (2020). "Translation Artifacts in Cross-lingual Transfer Learning." EMNLP.
- Office of the Registrar General and Census Commissioner, India (2018). Census of India 2011: Language. Paper 1 of 2018.
- Joshi, P. et al. (2020). "The State and Fate of Linguistic Diversity and Inclusion in the NLP World." ACL.
- Madhani, Y. et al. (2023). "Aksharantar: Open Indic-language Transliteration datasets and models for the Next Billion Users." EMNLP Findings.
- Khanuja, S. et al. (2020). "GLUECoS: An Evaluation Benchmark for Code-Switched NLP." ACL.
- Bhatt, S. et al. (2022). "Re-contextualizing Fairness in NLP: The Case of India." AACL-IJCNLP.
- Thompson, B. et al. (2024). "A Shocking Amount of the Web is Machine Translated: Insights from Multi-Way Parallelism." ACL Findings.
- Ramesh, G. et al. (2022). "Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages." Transactions of the Association for Computational Linguistics.
- Gala, J. et al. (2023). "IndicTrans2: Towards High-Quality and Accessible Machine Translation Models for all 22 Scheduled Indian Languages." Transactions on Machine Learning Research.
- Singh, H. et al. (2024). "IndicGenBench: A Multilingual Benchmark to Evaluate Generation Capabilities of LLMs on Indic Languages." ACL.
- Ministry of Electronics and Information Technology, Government of India. National Language Translation Mission (Bhashini), 2022; IndiaAI Mission, Cabinet approval, March 2024.

The Voice Gap

Speech is how most people in India will reach AI. It is also the modality in which the distance between available data and real use is greatest.

7 MINUTE READ

In 2020, researchers at Stanford tested the speech-recognition systems of five large technology companies on recordings of interviews with Black and white speakers in the United States. The systems had been trained on very large quantities of audio and were in commercial use.

For white speakers, the systems misrecognised on average 19 words in every hundred. For Black speakers, 35.

The cause was not the language, which was English in both cases. Analysis pointed to the acoustic models: the systems had been trained on speech that sounded more like one group than the other. The difference in accuracy was a difference in training data, expressed as a difference in service.

That study concerns one country and two groups. Extend the same logic to a country with twenty-two scheduled languages, hundreds of dialects and a very large population for whom speaking to a device is more natural than typing, and the scale of the data question becomes clear.

Accuracy follows hours

The relationship between training audio and recognition accuracy is among the better-established facts in the field.

OpenAI's Whisper model was trained on 680,000 hours of audio collected from the web. About 117,000 of those hours covered 96 languages other than English. The paper reports a strong relationship between the amount of training data for a language and the model's error rate in it: languages with more hours were recognised better, across the range. The authors note that many of the languages with poor performance simply had very little data.

Divide 117,000 hours among 96 languages, unevenly, and many Indian languages are left with a small share. And hours alone are not the whole story. Audio collected from the web is mostly prepared speech: broadcasts, audiobooks, lectures, subtitled video. It under-represents ordinary people talking in ordinary settings.

What Indian speech data needs to cover

Language. Each of the scheduled languages, and the many widely spoken languages outside the schedule.

Region and accent. Speech varies by district. The Svarah benchmark was created to test English recognition on Indian accents, with speakers from across the country, because models accurate on American and British English were noticeably less so on Indian English.

Speaker. Age, gender, education and rural or urban background all affect speech. A dataset recorded from university students in two cities describes those students.

Style. Reading a prepared sentence is acoustically and linguistically different from speaking spontaneously. Conversation, with interruptions, hesitations and overlap, is different again.

Language-mixing. Speakers switch between languages mid-sentence. Sitaram and colleagues' survey of code-switched speech and language processing describes how this breaks the assumption, built into most systems, that an utterance is in one language. A shared task on multilingual and code-switched recognition for Indian languages was organised at a leading speech conference in 2021 because the problem was recognised as distinct.

Conditions. Telephone audio, low-cost handsets, background noise from streets, markets and households.

Vocabulary. Names of people, places, crops, medicines, schemes and products.

Each of these is a dimension along which a dataset can have coverage or lack it, in the sense set out in the article on data quality. A model fails along whichever dimension its data did not span.

How the datasets have evolved

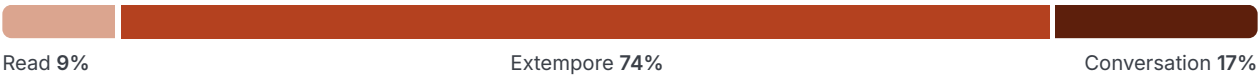
The sequence of open speech datasets from Indian research groups shows a clear progression in what was understood to matter.

FIGURE 26 · DATA VISUALISATION

Three open Indian speech datasets, three methods

DATASET	HOURS	LANGUAGES	SPEAKERS	DISTRICTS	KIND OF SPEECH
Kathbath	1,684	12	1,218	203	Read sentences
Shrutilipi	about 6,400	12	Radio broadcasts	n/a	News bulletins mined from archives
IndicVoices	7,348	22	16,237	145	Mostly spontaneous; see below

IndicVoices by speaking style



Sources: Javed et al. (2023), Bhogale et al. (2023), Javed et al. (2024).

Kathbath provided 1,684 hours of labelled speech across 12 languages from 1,218 contributors in 203 districts. It was read speech: contributors read sentences shown to them. This gives clean, well-transcribed audio and is relatively efficient to collect.

Shrutilipi took a different route, mining about 6,400 hours across 12 languages from the archives of All India Radio by aligning broadcast audio with its transcripts. This yielded volume at low cost, in the formal register of news bulletins.

IndicVoices changed the method. It collected 7,348 hours from 16,237 speakers across 145 districts and all 22 scheduled languages. Only 9% is read speech. Seventy-four per cent is extempore, where speakers responded in their own words to prompts about their lives and interests, and 17% is conversation. The authors describe a collection process involving district-level coordinators, recruitment designed to cover demographic groups, questions and scenarios tailored to regions and domains, and a multi-stage transcription and quality-control pipeline.

The shift from Kathbath to IndicVoices is a shift from recording a language to recording how its speakers talk. It is also a large increase in operational complexity. Read speech needs a script and a microphone. Spontaneous speech from sixteen thousand people in 145 districts needs field teams, local trust, consent procedures, and transcribers who understand each dialect well enough to write down what was said, including the hesitations and the switches into other languages.

Transcribing spontaneous speech is an annotation task with every difficulty this series has discussed. Transcribers disagree on how to render dialect words, on where one language ends and another begins, and on how to spell words that have no standard written form. Guidelines have to settle these questions, and the answers shape what the model learns.

The self-supervision argument

A fair objection is that modern speech models need far less labelled data than this account implies.

Self-supervised methods learn representations of speech from unlabelled audio and then need only a modest amount of transcribed speech to reach good accuracy. An AI4Bharat study pretrained a model on about 17,000 hours of raw speech across 40 Indian languages and showed substantial gains. Meta's Massively Multilingual Speech project extended recognition to more than a thousand languages, using recordings of religious texts as a source of aligned audio where nothing else existed.

These methods help. They reduce the quantity of transcription required. They do not remove the coverage requirement. Unlabelled audio still has to include the accents, conditions and speaking styles the model will meet. And a model trained largely on religious readings by a small number of speakers, as the MMS authors acknowledge, reflects that domain and those voices.

Evaluation is unaffected by any of this. To know the error rate for elderly women speaking Bhojpuri on a basic handset, somebody must have recorded and accurately transcribed elderly women speaking Bhojpuri on a basic handset.

KEY TAKEAWAY

A speech model serves the people whose voices were in its training data. Coverage is determined by whom you recorded, where, doing what, and it cannot be inferred from total hours.

Why this is a sovereignty question

Speech data has characteristics that make its governance more sensitive than text.

A voice recording is biometric. It can identify the speaker. Spontaneous speech about a person's life contains personal information. India's Digital Personal Data Protection Act of 2023 sets conditions for processing such data, centred on consent and purpose.

Speech data for Indian languages also has to be collected in India, from Indian speakers. It cannot be obtained elsewhere. That makes decisions about who collects it, under what terms, and who may use the result matters of public policy as well as commerce. The open release of datasets such as IndicVoices under permissive licences is one answer: publicly funded data as shared infrastructure. Government programmes including Bhashini and the IndiaAI Mission's datasets platform follow the same reasoning.

Open datasets provide a foundation of general coverage. They do not cover an insurer's claims vocabulary, a hospital's consultations or a bank's customer calls in six states. That domain-specific layer is for each organisation to build, and it carries the same obligations of consent and care.

Collecting speech data well

Several practices follow from the research.

Design the sample. Specify target counts by language, district, age band, gender and background before collection, and track them.

Prefer natural speech. Prompt speakers with questions and scenarios, not scripts, for a large share of the data.

Record in realistic conditions. Use the devices and environments in which the product will be used.

Obtain informed consent. Explain in the speaker's language what the recording is for, who will use it and how to withdraw.

Pay contributors fairly. They are providing something that cannot be obtained otherwise.

Transcribe with native speakers of the dialect. Set and document conventions for dialect words, mixed-language segments, numerals and disfluencies.

Measure transcription agreement. Have a sample transcribed twice and compare. Disagreements reveal where conventions are unclear.

Evaluate by slice. Report error rates by language, region, gender, age and recording condition, since the average conceals the groups served worst.

What this changes

FOR AI BUILDERS

Report word error rate by speaker group and condition. A single figure per language conceals most of what matters.

Include spontaneous and conversational speech in both training and test sets. Read-speech results overstate real-world accuracy.

Write explicit transcription conventions for code-mixed and dialect speech before transcription begins.

FOR ENTERPRISES

Test any voice product on recordings of your own customers, in your regions, on their devices, before launch.

Treat voice recordings as sensitive personal data, with consent, retention and access controls to match.

Expect to build domain-specific speech data. Open datasets will not contain your terminology or your callers.

THE INDIKA PERSPECTIVE

Speech is the clearest case in this series of data that has to be made by people, in place. It requires reaching speakers in their districts, recording them with consent in natural conditions, and transcribing with people who know the dialect. That is field operations, linguistics and annotation combined. For AI to work for those who will speak to it instead of typing, this is the work that has to be done, language by language.

RESEARCH & REFERENCES

Koenecke, A. et al. (2020). "Racial disparities in automated speech recognition." Proceedings of the National Academy of Sciences.

Radford, A. et al. (2023). "Robust Speech Recognition via Large-Scale Weak Supervision." ICML.

Javed, T. et al. (2023). "IndicSUPERB: A Speech Processing Universal Performance Benchmark for Indian Languages." AAAI.

Bhogale, K. et al. (2023). "Effectiveness of Mining Audio and Text Pairs from Public Data for Improving ASR Systems for Low-Resource Languages." IEEE ICASSP.

Javed, T. et al. (2024). "IndicVoices: Towards building an Inclusive Multilingual Speech Dataset for Indian Languages." ACL Findings.

Javed, T. et al. (2023). "Svarah: Evaluating English ASR Systems on Indian Accents." Interspeech.

Javed, T. et al. (2022). "Towards Building ASR Systems for the Next Billion Users." AAAI.

Sitaram, S. et al. (2019). "A Survey of Code-switched Speech and Language Processing." arXiv preprint.

Diwan, A. et al. (2021). "MUCS 2021: Multilingual and Code-Switching ASR Challenges for Low Resource Indian Languages." Interspeech.

Pratap, V. et al. (2024). "Scaling Speech Technology to 1,000+ Languages." Journal of Machine Learning Research.

Government of India (2023). The Digital Personal Data Protection Act, 2023.

Ministry of Electronics and Information Technology, Government of India. National Language Translation Mission (Bhashini), 2022; IndiaAI Mission, Cabinet approval, March 2024.