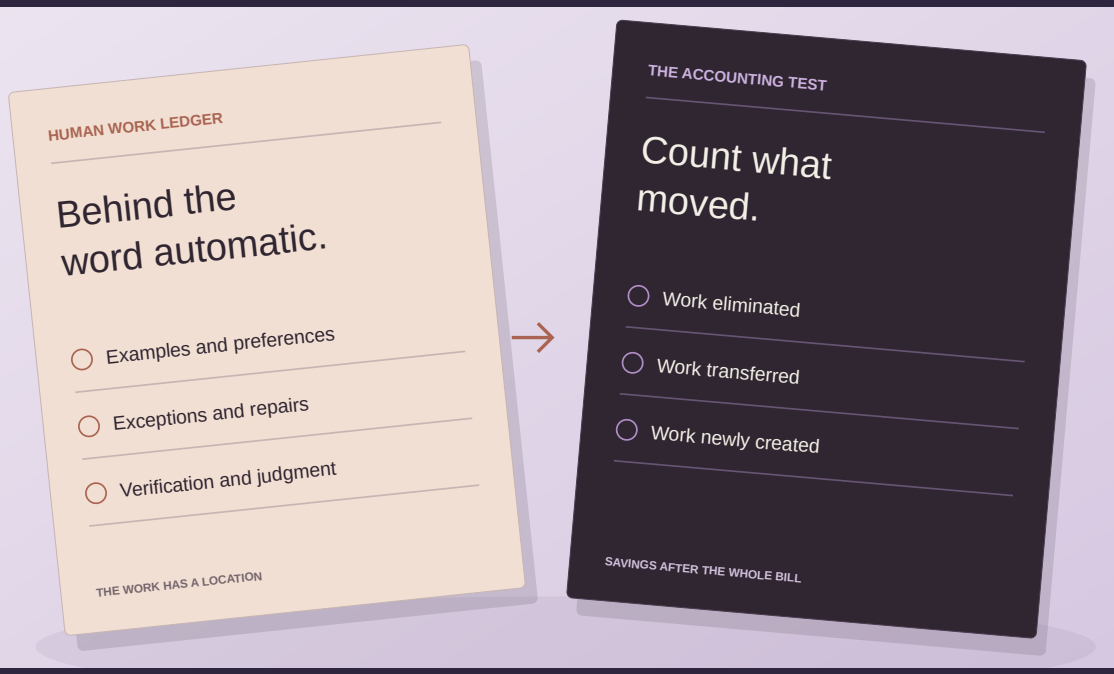
kymata labs

RESEARCH PAPER 06 / AI & WORK

The Work Behind the Word Automatic

AI can remove a task while leaving its judgments—and its consequences—in somebody else's hands.



Kymata Labs Research
2026-11-16 · Edition 1.0

Independent analysis
kymatalabs.com

Small studio. Open worlds.

THE LONGER VIEW

Executive summary.

Automation can eliminate work, reuse earlier judgment, or transfer unfinished obligations to someone outside the demonstration. This paper distinguishes upstream examples, preference annotation, live exception handling and verification using inspected first-party research artifacts and engineering guidance. Human-feedback records and synthetic-annotation methods are treated separately: model-generated ratings can deliver real savings while concentrating responsibility in the evaluator and rubric. A hypothetical expense workflow exposes the difference between completing a visible interaction and resolving the underlying case. The paper proposes an account of reusable setup, case work and maintenance, with explicit owners, observation boundaries and unresolved outcomes. It makes no claim about hidden industry headcount, wages, or individual commercial staffing. Its purpose is to distinguish a genuine reduction in effort from a saving recorded before somebody else finishes the work.

> Qualitative analysis of six inspected first-party repository documents, preserved with exact revisions and SHA-256 receipts. The linked full papers, external datasets and commercial staffing arrangements were not audited. Historical dataset methods are not treated as current industry measurements. The four-part work account, expense-service example and local evaluation design are original analytical proposals, not observed productivity results. Source inspection: November 16, 2026 UTC.

01 / The empty chair in the demonstration
02 / Four kinds of work hiding inside one word
03 / A preference is somebody's decision
04 / When the annotator is another model
05 / The exception desk is part of the product
06 / Checking is work, but it need not be all the work
07 / An account that can admit a saving
08 / A procurement question with an operational answer
09 / What these sources do not tell us
Q&A / Questions worth asking
Sources / References

© 2026 Kymata Labs LLC. All rights reserved.
Read the web edition: kymatalabs.com/research/the-work-behind-the-word-automatic

The empty chair in the demonstration

A demonstration of an autonomous product has a compelling cast: a person asks, a machine acts, a task disappears. The brevity is the point. Nobody wants a demonstration of every upstream decision, every discarded answer, or the colleague who will have to resolve an unusual case next Tuesday.

But a demonstration is also a frame. Work outside it has not necessarily vanished. It may have happened earlier, preparing examples and defining acceptable behavior. It may happen afterward, when a customer disputes the result. It may move to another department, become a supplier's responsibility, or return to the user as the instruction to check everything carefully.

The consequential question is not whether a machine did something real. It often did. The question is whose work the word “automatic” allows an organization to stop counting.

This paper argues that autonomy needs an account of judgment as well as an account of execution. That account must distinguish work genuinely eliminated from work moved, concentrated, or reused. It must also identify who can change a decision when the automated path becomes inappropriate. A system can execute without intervention while still depending on people to decide what execution ought to mean.

The evidence here comes from inspectable research and engineering artifacts. Human-feedback datasets show specific judgments that helped train or evaluate systems. Synthetic-feedback projects show that machines can perform some annotation work themselves. Code-review guidance makes a downstream acceptance obligation explicit. These sources do not expose the staffing or working conditions behind every commercial assistant. They support a narrower, useful argument about what organizations should measure and make accountable.

There is no worker testimony in this paper, no estimate of hidden industry headcount, and no inference that an automated service secretly employs a person for every answer. Those would be reporting claims requiring different evidence. The problem examined here can exist even when every visible step is performed by software.

Four kinds of work hiding inside one word

Begin before the model answers. Someone must decide what examples belong in a training or evaluation collection, what makes a response acceptable, and which failures deserve attention. Some of that activity can be automated. Its existence is nevertheless different from the act of generating a response for a user.

OpenAssistant's repository makes this distinction unusually visible. Its README describes contributing by submitting, ranking, and labeling prompts and responses. Its historical project plan separates human-generated instruction examples from ranking sampled completions, and describes collecting independent votes before using rankings to train a reward model. The plan is evidence of the intended process, not a verified count of completed annotations. The repository now marks the project completed. [1]

That account contains two distinct forms of work. Writing or selecting an example creates material from which a system can learn. Ranking candidate answers supplies a preference about how it should behave. Calling both “data” is convenient for storage and imprecise for responsibility. A well-written answer does not tell us which competing answer a user should prefer; a preference label does not establish the answer's factual truth.

Anthropic's HH-RLHF repository documents another distinction. Its human-preference records contain a chosen and a rejected text. Its red-teaming records separately describe conversations with human adversaries, their descriptions of attempted attacks, and their ratings of success. A preference between two answers and a deliberate attempt to expose a failure are different activities, even when both eventually become training material. The repository is now deprecated in favor of its hosted dataset. [2]

After deployment, two further categories become important: exception handling and verification. Exception handling decides what happens when the ordinary route cannot appropriately finish a case. Verification establishes whether an output or action meets the applicable standard, sometimes before release and sometimes through later sampling. These categories can overlap in one person's job. They answer different questions.

A reviewer can verify that an answer follows a policy and still lack authority to grant an exception to it. A support specialist can authorize an exception without having inspected the model's earlier training data. An evaluator can detect systematic errors without being empowered to repair an individual customer's outcome. “Human in the loop” conceals these differences if it names a person without specifying their function.

FIGURE 01 ANALYTICAL WORK MAP

Four judgments, four different responsibilities

The person supplying an example is not necessarily the person who can resolve a disputed outcome. Separate the function before assigning a reviewer.

01

Material

Upstream examples

Create or select examples from which a system can learn. Decide what belongs in the collection.

02

Judgment

Preference annotation

Compare or rate candidate responses under a stated rubric. Human and model annotations must be distinguished.

03

Authority

Exception handling

Resolve a present case that the ordinary path cannot finish. Preserve evidence and an empowered destination.

04

Acceptance

Verification

Check a result against the applicable standard. Disclose what was examined and what remains outside scope.

A functional distinction, not an exhaustive labor taxonomy, headcount estimate or claim that every deployment uses all four functions.

Kymata Labs analytical classification, informed by first-party dataset and review documentation. [1] [2] [3] [6]

The four-part account used here is therefore operational, not a claim that the entire AI supply chain fits four boxes. Upstream examples establish material; preference annotation supplies behavioral judgments; exception handling resolves departures from routine; verification tests whether a result deserves acceptance. Management, infrastructure, research, maintenance, and the creation of source works also matter. The narrower classification isolates judgments most easily lost when a product promises to finish work by itself.

A preference is somebody's decision

The word “feedback” can make annotation sound like a residue produced after the important activity. In the inspected artifacts, it is a representation of a decision that the training or evaluation process can use.

OpenAI's archived summarization repository describes a released human-feedback dataset and shows its record structure. A comparison includes two summaries, a worker identifier, a selected answer, and optional notes. Separate evaluation records contain ratings along named axes. The README distinguishes data used to select models during training from labels reserved for final evaluation. We inspected this documentation, not the external dataset or the full research paper. [3]

These fields are worth reading as an editorial record. The optional notes preserve a trace of interpretation. The comparison records which candidate was preferred. The evaluation axes permit different qualities to be considered separately. None turns a subjective judgment into a universal truth

merely by giving it a column in a file.

Consider an illustrative assistant answering an employee's question about a travel expense. One answer may be concise and useful while quietly assuming the employee is eligible. Another may correctly insist on checking eligibility while frustrating someone who wants a quick answer. Which is better depends partly on the task, policy, available evidence, and consequences of error. That choice can be expressed as a preference, but the conditions still matter after the label is recorded.

The same issue applies when a task has a definite answer. A preferred response can contain a mistake that neither candidate makes obvious. Comparing candidates can improve behavior without certifying every proposition in the preferred text. This is a logical distinction, not an empirical claim that human-feedback datasets are generally unreliable.

An organization buying a model will seldom see every such decision. That is an information boundary, not proof of wrongdoing. It should resist translating “trained with human feedback” into “independently verified for our use.” A feedback process and a local acceptance test have different subjects, and the people making those judgments may have different information.

When the annotator is another model

The obvious reply is that this labor account describes yesterday's AI. If a model can judge responses, does the argument survive synthetic feedback?

It does, but it must change. Counting every preference label as a human action would be wrong. AlpacaFarm's repository explicitly describes simulating pairwise preferences with language models and conducting automatic evaluation. It presents the framework as a way to make research on learning from feedback more accessible. The inspected documentation does not establish the savings for a particular deployed business. It does establish that annotation need not involve a fresh human comparison for each example. [4]

UltraFeedback makes the distinction equally concrete. Its documentation describes sampling instructions from multiple datasets, obtaining model responses, and asking GPT-4 to annotate them under instruction-following, truthfulness, honesty, and helpfulness criteria. The rubric is part of the method; the generated feedback is another part. These are model annotations, not a collection of direct human verdicts on every response. [5]

That is a genuine kind of automation. A reusable rubric plus a model may replace many repeated rating operations. It can make an evaluation affordable enough to run more often or across material that would otherwise go unchecked. An account that calls this mere displacement, without measuring what actually remains, would be as misleading as the sales claim it criticizes.

However, the method also changes where judgment concentrates. Someone chooses the evaluator, the instructions, the material, and the conditions under which the resulting scores will be trusted. Those choices affect many annotations at once. A smaller amount of work may acquire greater reach. Human involvement can decline while the consequences of a particular human decision increase.

The repositories preserve two cautions about that concentration. AlpacaFarm's README says its original automatic annotator pool could no longer be used after an underlying model's deprecation. It documents a switch and warns readers that results should be compared with the corresponding AlpacaEval configuration, rather than the original AlpacaFarm results. The same label for a task does not guarantee the same evaluator over time. [4]

UltraFeedback's README documents an overall-score correction announced in December 2023. It describes rules using fine-grained scores and re-annotation for remaining cases. It also explicitly acknowledges that GPT-4 can make mistakes. This paper does not reproduce the correction, inspect the linked issue, or attribute hours of human work to it. The documented episode illustrates maintenance of an automated judgment process, not a measured labor bill. [5]

FIGURE 02 — DOCUMENTED METHODS · ANALYTICAL IMPLICATION

Automating the rating changes where judgment sits

Synthetic annotation can replace repeated human comparisons. Its rubric and evaluator still determine what the resulting scores mean.

01

A person rates

Human comparison

A recorded choice or rating represents a human judgment about particular candidate outputs.

02

A model rates

Synthetic comparison

An evaluator model applies instructions to candidate outputs. Each resulting label is not a fresh human verdict.

03

Someone decides

Process ownership

Select the rubric and evaluator; check their suitability when the process or model changes.

No measured labor saving or superiority of one method is claimed. The sources document methods, configuration changes and limitations.

Human-feedback documentation from Anthropic and OpenAI; synthetic-feedback documentation from AlpacaFarm and UltraFeedback; Kymata Labs analysis. [2] [3] [4] [5]

The important distinction is between performing each rating and being responsible for the rating system. Automation may reduce the former. Whether it reduces the latter, or makes that responsibility easier to discharge, requires evidence. Invoking an infinite chain of people behind every tool would tell us nothing. Naming the decisions that materially affect the current workflow tells us where to look.

The exception desk is part of the product

Upstream annotation and live exception handling should never be collapsed into a single image of a hidden worker. One helps shape a model or its evaluation. The other resolves a present case. Their costs, timing, and authority differ.

Return to the hypothetical travel assistant. It can read an expense policy, identify a stated limit, and assemble a reimbursement request. These may be useful automated operations. Now imagine that the employee's trip was extended because an event organizer changed the schedule. The employee has a written authorization, but the standard policy document does not describe this circumstance.

A plausible answer is no longer the entire job. Someone must establish whether the authorization is valid, whether it covers the claimed expense, and who may approve an exception. The assistant may gather the relevant documents and route the case. That is progress. A final decision may still require a person with the appropriate authority, or a separately authorized rule that already covers the situation.

Calling the task complete when the request leaves the chat window would hide this boundary. Calling the whole system useless because an exception remains would hide the useful work. The product is the path through the ordinary case and the exceptional one, including the point at which responsibility changes hands.

An effective handoff therefore needs more than a button marked “contact support.” It needs the original request, the evidence already gathered, the action already attempted, the reason the automated path stopped, and a destination able to decide. These are proposed design requirements. No claim is made that the cited datasets implement such a service workflow.

Without this continuity, work can return to the user as repeated explanation. In the example, the employee may have to reconstruct the conversation, locate the authorization again, and explain what was already tried. That would be transferred effort, even if the automated interaction was brief. To establish that it happens in a real service would require observing that service; the hypothetical identifies what an evaluation should record.

This also explains why a declining escalation count is ambiguous. It might mean that more cases are resolved correctly. It might mean the route to help has become harder to find, or that people abandon the task. A count of escalations, by itself, cannot distinguish these explanations. Pair it with dispositions and a way to examine unresolved outcomes before announcing that the work disappeared.

Checking is work, but it need not be all the work

Verification is sometimes dismissed as a temporary inconvenience that better models will erase. It is also sometimes exaggerated into a demand that a person repeat every automated operation. Both positions make the economic question harder to answer.

Google's code-review guidance supplies a useful, bounded example. It asks reviewers to consider system fit, functionality, complexity, and whether tests would fail when behavior is broken. It also asks reviewers covering only part of a change to make that scope explicit. This is engineering guidance, not a

study of AI labor or a rule that every product needs line-by-line human inspection. [6]

The relevance is the shape of acceptance. A result can pass a mechanical check while leaving a question about whether that check tests the right thing. A second model can provide useful criticism while leaving a question about whether its rubric matches the intended decision. Review has a subject and a scope; the presence of a reviewer cannot substitute for either.

Proportion matters. A reversible formatting suggestion does not justify the same inspection as a consequential action involving money or access. A mature deterministic check may make routine human review unnecessary. A constrained workflow with a narrow permission boundary may safely automate more than an unrestricted assistant. These are design possibilities, not measured performance claims.

The goal should be to reduce the cost of reaching a defensible outcome. Better evidence presentation, smaller action scope, reliable checks, and clearer exception rules can all help. Demanding that a person recreate the task from scratch would erase some of the very benefit being evaluated. Allowing a person to approve a result without enough information would turn oversight into a signature without substance.

There is a fairness issue here as well as an efficiency issue. If a named reviewer lacks time, evidence, or authority to intervene, assigning responsibility to them does not solve the underlying design problem. This is a governance judgment, not a finding about any named employer. It follows from the mismatch between what the person is expected to guarantee and what the workflow allows them to do.

An account that can admit a saving

A useful labor account must permit the answer that automation worked. Otherwise it is a way to redistribute suspicion, not a way to understand a product.

Start with one bounded outcome: for example, a reimbursement decision accepted under a specified policy, including a route for a disputed decision. State the service quality required, the observation period, and the cases eligible for comparison. Count a rejected or abandoned attempt as an outcome with costs, not as missing data conveniently outside the denominator.

Next separate three kinds of cost: reusable setup, work triggered by individual cases, and continued maintenance. A training rubric used across many requests should not be charged in full to every request. Neither should it disappear because it was written before the measurement window. Record the allocation rule and vary it if the expected volume is uncertain.

Within each category, record who performs the activity. Employee, contractor, supplier, user, and model are different entries. A model invocation is not person-time. A vendor invoice does not disclose the vendor's full labor supply chain. Customer effort may be outside the buyer's payroll and still affect whether the service is better. State what the account includes, what it cannot observe, and why.

Decision boundary	What to record	The interpretation it permits
Setup and calibration	Rubric, owner, effort, expected reuse	Whether reusable work is fairly allocated
Routine processing	Accepted outcome and active person-time	Whether ordinary cases need less human work
Exceptions and disputes	Reason, destination, resolution, user effort	Whether apparent savings reappear elsewhere
Verification and maintenance	Scope, checks, revisions, review effort	Whether acceptance and continued use remain supportable

These records should support two separate comparisons. Does total observed effort decline for an equivalent outcome? And how is the remaining effort distributed? A workflow can reduce total effort while concentrating difficult cases in one team. That result is compatible with a real productivity gain and a need to redesign staffing or authority. Aggregate savings do not settle the distribution question.

Conversely, a workflow may move effort from employees to customers while improving a department's headline metric. Whether that trade is acceptable depends on the service and user experience, but it should be described as a transfer where observed. An organization cannot determine the answer from its payroll system alone.

FIGURE 03 PROPOSED ACCOUNTING TEST

Where does the saving survive?

Follow an equivalent outcome past the visible response. Keep effort reduction and effort distribution as separate findings.

01

Outcome

Define completion

Specify acceptance, quality, eligible cases and the observation window before comparing workflows.

02

Effort

Retain the work

Include allocated setup, case handling, review and maintenance. Preserve rejected and unresolved attempts.

03

Distribution

Name the owner

Separate employees, suppliers, users and model execution. Unobserved work stays unknown.

04

Comparison

Test the gain

Report total observed effort and who bears it. A real saving can coexist with concentrated difficult work.

This method has not been validated as an intervention. It is designed to recognize genuine savings as well as transfers and unresolved obligations.

Kymata Labs proposed evaluation method, motivated by the distinctions in the inspected feedback and review artifacts. [1] [2] [3] [4] [5] [6]

A procurement question with an operational answer

The next step is to turn that account into a small evaluation, not an indefinitely expanding documentation project. Choose a representative class of work and define completion before testing. Keep the comparison standard stable. Give both automated and existing processes the benefit of any policy clarification introduced during the trial, so that better instructions are not automatically credited to the new technology.

Retain the case record through resolution, including escalation, abandonment, correction, and appeal where relevant. Measure active human effort separately from elapsed time: a request waiting overnight has delay but not necessarily continuous labor. Record assistance actually used. If some work is unobservable, mark it unknown instead of entering zero.

The evaluation should also test whether the proposed reviewer can review. Show the reviewer a case with a known, safely constructed mismatch between the answer and its evidence. Can they locate the problem, stop the action, and send it to an appropriate owner? This is a proposed local acceptance check, not an experiment run for this paper. Passing one such check would demonstrate that path under those conditions, not certify all future oversight.

Ask the vendor for a clear boundary statement: what the service performs, what it expects the buyer to verify, what happens when it cannot finish, and which changes to models or evaluators may affect the process. The buyer can then decide which obligations fit its actual staffing and competence. No access to private training data is required to discover that a promised verification task has no owner.

For workers, the useful disclosure is similarly practical. Which judgments are being delegated? Which remain? What evidence will be available, and who can reverse a decision? A promise to keep a human involved is weak if it does not answer those questions. Consultation should include people who resolve exceptions, not only those who approve the purchase.

The result may justify expansion. Routine work may become cheaper, checks may improve, and difficult cases may receive more attention. It may justify a narrower launch with better routing. It may show that the apparent gain depended on asking customers or an understaffed review team to absorb the cost. Each is a meaningful finding if the measurement boundary is visible.

What these sources do not tell us

The research artifacts establish specific, documented methods. OpenAssistant describes contributed examples and rankings. Anthropic documents human-preference and red-teaming record types. OpenAI documents human summary comparisons and evaluation fields. AlpacaFarm and UltraFeedback document model-generated judgments and changes to those processes. Google supplies explicit review criteria. We inspected the captured repository documents, not the associated full papers, private operations, or external datasets. [1][2][3][4][5][6]

They are historical and methodological evidence, not a representative sample of today's products. OpenAssistant is marked completed; the OpenAI repository is archived; Anthropic's repository is deprecated. Their documentation remains useful for distinguishing kinds of judgment, but none should be presented as a current commercial staffing disclosure.

No industry labor total, compensation assessment, employment forecast, or causal productivity estimate follows from this collection. The four categories, service example, accounting record, and procurement evaluation are this paper's proposals. They require local evidence before supporting a claim about a particular deployment. There is no claim that human oversight always improves outcomes or that synthetic feedback necessarily degrades them.

What remains is a less theatrical question than whether machines can work without people. Which decisions did the machine actually remove, which decisions did it inherit, and which decisions is somebody still expected to stand behind?

The empty chair in the demonstration can represent genuine progress. It can mean that a person no longer has to perform a tedious step. It can also mean that the person doing the consequential work is outside the frame. A credible automation claim tells us which, and gives the remaining people enough evidence and authority to do their jobs.

Questions worth asking.

Does human work behind AI mean the product is not really automated?

No. A system can genuinely remove repeated tasks while relying on earlier examples, reusable rules or occasional review. The question is which work it removes and which obligations remain. Training-time human involvement does not imply that a person operates each live response. Evaluate the complete outcome and its remaining costs rather than treating all human involvement as evidence against automation.

What is the difference between human feedback and AI feedback?

Human feedback records people’s comparisons, ratings or critiques. AI feedback uses a model to produce such judgments under a specified rubric. AlpacaFarm and UltraFeedback document model-generated judgments, so their annotation outputs should not be counted as fresh human ratings. Synthetic feedback can reduce repeated work, but its evaluator, instructions and continued suitability still need an accountable owner.

Why separate exception handling from verification?

Verification asks whether a result meets its standard. Exception handling determines what to do when the ordinary process cannot appropriately finish a case. A reviewer may detect that an authorization is missing without being allowed to grant it. Naming the two functions separately makes their evidence needs, decision rights and handoff routes clearer.

How can a team tell whether AI saves work or shifts it?

Define an equivalent completed outcome, then track setup, case work, exceptions, verification and maintenance over a declared period. Record who performs each activity, including user effort where observable, and retain rejected or abandoned attempts. Report total observed effort separately from its distribution. State what is unknown; an unmeasured supplier or customer task is not automatically zero.

Does this paper measure hidden workers or working conditions?

No. Its primary evidence is public documentation of feedback and review methods, not interviews, payroll records or a survey of deployed services. It supports distinctions between kinds of judgment and a proposed accounting method. It does not establish wage levels, current employment totals, exploitation claims or the staffing behind a particular commercial assistant.

SOURCES, NOT SCENERY

References.

1. **OpenAssistant: README and historical project plan**
   LAION / OpenAssistant · Repository snapshot inspected 2026–11–16 · Primary project documentation

   Inspected README describes submitting, ranking and labeling contributions; its historical plan separates human examples, ranked completions and reward-model training. Planned quantities are not treated as completed measurements. The repository marks the project completed. No full paper or live collection system was inspected. Captured revision: f1e6ed9526f5817531f3ab85441a40b3671ddccb.

   https://github.com/LAION-AI/Open-Assistant/blob/f1e6ed9526f5817531f3ab85441a40b3671ddccb/README.md
   Accessed copy: https://raw.githubusercontent.com/LAION-AI/Open-Assistant/f1e6ed9526f5817531f3ab85441a40b3671ddccb/README.md

2. **Helpful and Harmless RLHF: dataset documentation**
   Anthropic · Repository snapshot inspected 2026–11–16 · Primary dataset documentation

   Inspected README documents chosen/rejected human-preference records and human red-teaming fields. It directs readers to papers for collection and crowdworker details; those papers and external data were not inspected here. The repository is deprecated in favor of the hosted dataset. Captured revision: c72f5cee8eb7b4d2ea5617657f4430d5e333af07.

   https://github.com/anthropics/hh-rlhf/blob/c72f5cee8eb7b4d2ea5617657f4430d5e333af07/README.md
   Accessed copy: https://raw.githubusercontent.com/anthropics/hh-rlhf/c72f5cee8eb7b4d2ea5617657f4430d5e333af07/README.md

3. **Learning to Summarize from Human Feedback: dataset record structure**
   OpenAI · Repository snapshot inspected 2026–11–16 · Primary dataset and evaluation documentation

   Inspected archived README documents comparison records, worker identifiers, optional notes, evaluation axes, and the separation of model-selection and final-evaluation labels. External data and the full paper were not retrieved, and no dataset-scale labor estimate is made. Captured revision: 700967448d10004279f138666442bf1497d0e705.

   https://github.com/openai/summarize-from-feedback/blob/700967448d10004279f138666442bf1497d0e705/README.md
   Accessed copy: https://raw.githubusercontent.com/openai/summarize-from-feedback/700967448d10004279f138666442bf1497d0e705/README.md

4. **AlpacaFarm: simulated preferences, automatic evaluation and annotator change**
   Stanford Tatsu Lab · Repository snapshot inspected 2026–11–16 · Primary research implementation documentation

   Inspected README documents model-generated pairwise preferences, automatic evaluation, and the change of automatic annotator after model deprecation. Its comparability warning is preserved. No experiment, present-day performance figure, or business labor-saving estimate was reproduced. Captured revision: 30717ddae735365de756ee2085191b491a71788d.

   https://github.com/tatsu-lab/alpaca_farm/blob/30717ddae735365de756ee2085191b491a71788d/README.md
   Accessed copy: https://raw.githubusercontent.com/tatsu-lab/alpaca_farm/30717ddae735365de756ee2085191b491a71788d/README.md

5. **UltraFeedback: annotation method, score correction and limitations**
   OpenBMB / UltraFeedback authors · Repository snapshot inspected 2026–11–16 · Primary dataset methodology documentation

   Inspected README documents GPT-4 annotations along four named dimensions, a December 29, 2023 score-correction notice, and the limitation that GPT-4 can make mistakes. The linked issue, external dataset, correction script and full paper were not inspected; no repair labor count is inferred. Captured revision: bf80fd46a8c6ceecc86e8babb1ae8771f26a3cbb.

   https://github.com/OpenBMB/UltraFeedback/blob/bf80fd46a8c6ceecc86e8babb1ae8771f26a3cbb/README.md
   Accessed copy: https://raw.githubusercontent.com/OpenBMB/UltraFeedback/bf80fd46a8c6ceecc86e8babb1ae8771f26a3cbb/README.md

6. **What to look for in a code review**
   Google Engineering Practices · Repository snapshot inspected 2026–11–16 · Primary engineering guidance

   Inspected full guidance covers design, functionality, useful tests and disclosure of a reviewer's limited scope. Used as an explicit example of acceptance criteria; not an AI labor study or universal requirement for human review. Captured revision: 3bb3ec25b3b0199f4940b1aa75f0ac5c5753301c.

   https://github.com/google/eng-practices/blob/3bb3ec25b3b0199f4940b1aa75f0ac5c5753301c/review/reviewer/looking-for.md
   Accessed copy: https://raw.githubusercontent.com/google/eng-practices/3bb3ec25b3b0199f4940b1aa75f0ac5c5753301c/review/reviewer/looking-for.md