From Messages to Measurements

The Mathematics of LLM-Assisted Labeling

Expected Parrot • Urteil project

First working edition — September 2026
Urteil 0.2.0

How to read this manual

A label is a measurement made under a rule. The useful questions are not just whether a model can return a category, but what that category means, which population the results describe, how errors are counted, and whether repeating or combining judgments supplies new information.

This manual connects those questions to Urteil's auditable workflow. It uses a completed experiment on 100 messages from the UCI SMS Spam Collection: a keyword rule, GPT-4o mini, GPT-4.1, and GPT-5. The predictions are actual recorded outputs, not simulated observations. The packaged data contain row IDs and labels, rather than the message texts or account credentials.

Building the manual is entirely offline. It replays arithmetic over saved labels, using the same classification-score function as the metrics CLI. It does not repeat the LLM experiment, execute `ep`, or require an API key. The supplementary interval, paired-test, and kappa calculations are explicit teaching helpers. They are not additional general-purpose metrics commands.

Read Chapters 1–3 before designing a study. Chapters 4–6 explain consensus, uncertainty, and decisions. Chapter 7 recomputes the worked example from the packaged labels; Chapter 8 maps the mathematics to concrete commands. Chapters 9–10 connect the LLM annotation literature to independent human validation, prompt sensitivity, and corrected finite-population estimates. Chapters 11–12 stress-test those methods with known synthetic truth and report a controlled prompt experiment. The manual plots saved simulation summaries; use `urteil simulate` to regenerate the Monte Carlo study itself.

```
urteil docs show manual
urteil docs manual --output-dir urteil-book --pdf
```

The output directory must be new. Without `-pdf`, this still exports editable LaTeX, the data and provenance, generated tables, calculations, and a build manifest. PDF compilation needs an optional local LaTeX installation. Sources and data can be reproduced exactly in a compatible package version; PDF metadata and fonts can vary across TeX installations.

The worked subset is balanced and publicly available. It is useful for checking a workflow, but it is not a deployment benchmark. Neither a narrow confidence interval nor a verified file hash establishes external validity. Rubric construction and aggregate scorecards are covered in the companion scoring manual, migrated from Rudin. Use `urteil docs show scoring` for codebook-driven LLM ratings and local scoring of saved sources, or `urteil scoring docs manual -output-dir scoring-book -pdf` to build its worked grant and investment study. The two manuals serve different purposes: measurement quality here, and explicit decision rules in the companion.

Contents

Chapter 1

Define the measurement

1.1 Items, questions, and raters

Let x_i denote the information available about item i, for $i = 1, \ldots, N$. For question q, choose a label space $\mathcal{K}_q$ and an operational rubric $\mathcal{R}_q$. A model judgment is

$$\widehat{y}_{iqr} = f(x_i; \mathcal{R}_q, \theta_r, u_{ir}) \in \mathcal{K}_q, \tag{1.1}$$

where θ_r records the model, agent configuration, and execution settings; u_{ir} represents any stochastic variation. A new model or iteration defines a distinct label source. The same source should not contribute two votes for the same item and question merely because its result file was imported twice.

For SMS classification, $\mathcal{K} = \{\text{ham}, \text{spam}\}$. "Spam" must mean something operational, such as unsolicited promotional content. A riddle, an intimate message, or a notification might be unwanted by one recipient without meeting that definition. If sender or consent context is unavailable, the model cannot recover it simply by producing a confident explanation.

A recorded reference g_{iq} is a separate measurement. It may be a human annotation, a reviewed consensus, or a published corpus label. An unobserved construct y^*_{iq}, such as whether the recipient actually consented, need not equal g_{iq}. The observable quantity $\Pr(\widehat{y} = g)$ is reference agreement; it is not automatically $\Pr(\widehat{y} = y^*)$. Throughout the example we call the former "accuracy" for compatibility with the CLI, while keeping the reference explicit.

1.2 Freeze what the comparison means

A model comparison should hold the available items and rubric fixed unless the goal is explicitly to compare whole workflows. If the prompt is revised for one model after looking at its mistakes, both model and development process have changed. Record that as a new experiment. Similarly, a calibration or validation split must be separated before using its labels to choose a rubric.

The observational unit is usually an item, not an API call. Multiple messages from one sender, copies of a template, or translations of one document can be correlated. Define the unit used for splitting and uncertainty calculations before counting the apparent sample size.

Only declared fields should reach the model. In Urteil, `columns_used` and each question's `fields`

determine prompt-visible content; the stable ID is retained for matching. A reference column may remain in the source CSV and final export while being excluded from every model scenario. Inspect the actual Jobs artifact to verify that separation.

1.3 Missing, unclear, and deliberately skipped

Define $m_{iqr} = 1$ when a usable label is available and 0 otherwise. A missing result can reflect an execution failure, an intentional skip, or absence of review. Those causes have different interpretations even if a particular score excludes them all.

An explicit "unclear" answer is an element of a label space, not a blank. If included in classification, it becomes a category with its own support and confusion counts. If a workflow treats it as abstention, evaluate the retained population and report abstention coverage. Do not silently convert it to the negative class. A task-level ambiguity policy is descriptive unless supported by the actual question options, prompt, and routing rules.

For conditional questions, specify the relevant population. "Which product is advertised?" is undefined for a message with no advertisement. The export coverage calculation follows declared question columns and available labels; it does not infer a custom eligibility denominator from scientific intent. Report conditional eligibility separately when needed.

1.4 What provenance can establish

Let D, S, and J denote the dataset bytes, specification bytes, and Jobs artifact. A plan binds their fingerprints $(H(D), H(S), H(J))$. Recording results against that plan also checks their survey, scenarios, raters, iterations, and question coverage. This prevents a result from a different task being silently accepted as the planned run.

Hashes and structural checks establish consistency with the archived contract. They are not a signature from the model provider, a proof of correct human labels, or evidence that a classifier generalizes. Reproducibility also requires keeping the archived bytes: a hash alone cannot reconstruct a deleted dataset.

An approved sample can be reused in a full result only when the contract still matches. If A is the set of sample IDs and B the remaining IDs, the intended full source has $A \cap B = \varnothing$ and $A \cup B = \{1, \ldots, N\}$. This set identity is both an accounting rule and a guard against paying twice for, or double-counting, the same observations.

Chapter 2

Count errors and coverage

2.1 The comparison population comes first

For a chosen question and source, let G be the IDs with reference labels and P the IDs with predictions. The CLI compares the intersection $I = G \cap P$, with $n = |I|$. It reports the unmatched IDs as well as

$$C_G = \frac{|I|}{|G|}, \qquad C_P = \frac{|I|}{|P|}. \tag{2.1}$$

If predictions exist for 20 of 100 referenced messages, $C_G = 0.2$ and $C_P = 1$. A high score on those 20 must not be described as measured quality on all 100. No overlap is an error, rather than a report full of artificial zero scores.

Coverage in an export answers a different question. For N dataset rows and Q expected label columns from the selected source, let M_{iq} indicate a nonempty exported label. Overall label coverage is

$$C_L = \frac{\sum_{i=1}^{N} \sum_{q=1}^{Q} M_{iq}}{NQ}, \qquad C_{\text{complete}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left\{ \sum_{q} M_{iq} = Q \right\}. \tag{2.2}$$

Some rows may be partially labeled even when every individual column has good coverage. The joined export preserves those rows and original columns. Source coverage and reference coverage should both appear beside quality measures.

2.2 Confusion counts

For labels $a, b \in \mathcal{K}$, with reference on the rows and prediction on the columns, define

$$C_{ab} = \sum_{i \in I} \mathbf{1}\{g_i = a, \widehat{y}_i = b\}. \tag{2.3}$$

The reference support is $s_a = \sum_b C_{ab}$ and predicted support is $t_b = \sum_a C_{ab}$. For a binary task whose positive class is spam:

	Predicted ham	Predicted spam
Reference ham	TN	FP
Reference spam	FN	TP

Choose the positive class explicitly with `-positive-label spam`. Accuracy is unaffected by swapping class names, but precision and recall are not. Urteil's binary default is the first sorted observed label; that may not be the substantive positive class you intended.

2.3 Class scores and averaging

For any class k, the one-versus-rest counts are

$$\mathrm{TP}_k = C_{kk}, \qquad \mathrm{FP}_k = t_k - C_{kk}, \tag{2.4}$$

$$\mathrm{FN}_k = s_k - C_{kk}, \qquad \mathrm{TN}_k = n - s_k - t_k + C_{kk}. \tag{2.5}$$

Then

$$P_k = \frac{\mathrm{TP}_k}{t_k}, \qquad R_k = \frac{\mathrm{TP}_k}{s_k}, \qquad F_{1k} = \frac{2\mathrm{TP}_k}{2\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k}. \tag{2.6}$$

Precision asks how trustworthy predictions of class k are; recall asks how many reference items in that class are found. Neither is a universal substitute for the other. False-positive and false-negative costs determine their relative importance in an application.

Accuracy is $\sum_k C_{kk}/n$. With every class score defined, macro F1 is $K^{-1}\sum_k F_{1k}$ and weighted F1 is $\sum_k s_k F_{1k}/n$. Macro F1 gives rare and common classes equal weight. Weighted F1 reflects reference support. *Macro F1 is the mean of class F1 scores, not the harmonic mean of macro precision and macro recall.*

For complete single-label classification, summed false positives and false negatives both equal the number of mistakes. Micro precision, recall, and F1 therefore equal accuracy. This identity does not extend without qualification to multilabel outputs or pooled unrelated questions. The current classification command compares one question at a time; a serialized list is not automatically scored as a collection of independent binary decisions.

2.4 Undefined is not the same as zero

If a model never predicts k, precision for k has denominator zero. If no reference item belongs to k, recall is undefined. Direct count-based F1 can still be defined when one of these ratios is not: one false negative and no predicted positives gives F1 equal to zero.

Urteil supports `-zero-division null`, `0`, or `1`. With null, an average excludes undefined entries separately for each metric and renormalizes its denominator. Weighted averages use the support of included classes. Numeric replacements enter the average. The report records defined classes and included support, so two different conventions are not mistaken for the same calculation. The observed class set is the union of reference and candidate labels on I; a class absent from both is not invented as a zero row.

Chapter 3

Sampling and population transport

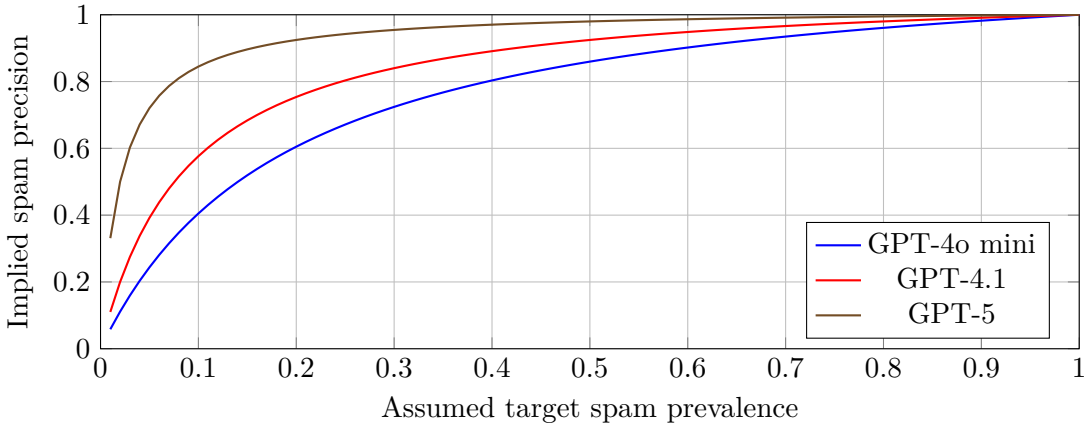
3.1 Balanced evaluation is a design choice

The SMS example samples 50 ham and 50 spam messages. Its 50% spam prevalence is imposed by design. It does not estimate either the corpus prevalence or the prevalence of messages reaching a future application.

Let $\pi = \Pr(Y = 1)$, sensitivity $s = \Pr(\widehat{Y} = 1 \mid Y = 1)$, and false-positive rate $f = \Pr(\widehat{Y} = 1 \mid Y = 0)$. If these two conditional rates transfer to a target population, Bayes' rule gives

$$\mathrm{PPV}(\pi) = \frac{s\pi}{s\pi + f(1-\pi)}, \qquad \mathrm{Acc}(\pi) = s\pi + (1-f)(1-\pi). \tag{3.1}$$

For illustration, with $s = 0.98$, $f = 0.02$, and only 1% spam, precision is $0.0098/(0.0098 + 0.0198) \approx 0.331$. Thus the same class-conditional behavior can produce 98% precision in a balanced sample and only about 33% in a low-prevalence setting. This arithmetic assumes stable conditional error rates; language, sender, time, and adversarial shifts can invalidate it.



Curves hold observed class-conditional error rates fixed; they are not forecasts.

3.2 Weights and inclusion probabilities

For a finite population and a probability sample with known inclusion probabilities $\rho_i > 0$, a weighted estimate of a confusion-cell proportion is

$$\widehat{p}_{ab} = \frac{\sum_{i \in I} \rho_i^{-1} \mathbf{1}\{g_i = a, \widehat{y}_i = b\}}{\sum_{i \in I} \rho_i^{-1}}. \tag{3.2}$$

This is a ratio estimator. Precision, recall, and other metrics can be formed from the weighted cells, with uncertainty appropriate to the sampling design. A simple average of class metrics is not a general replacement for weighting. For binary accuracy alone, a known target prevalence weights the two class-conditional correct rates as shown above.

The metrics CLI currently uses unweighted counts. The formula explains a possible external analysis, not an implemented weighting option. Keyword, longest-message, or disagreement sampling can be excellent for finding failure modes, but does not acquire a representative-population interpretation without additional design information. Unlabeled selection bias is not repaired by calling a sample "random" after it has been filtered on model errors.

3.3 Abstention and selective performance

If a workflow retains items according to a decision $A_i \in \{0, 1\}$, define coverage $c = \Pr(A = 1)$ and selective risk $R = \Pr(\widehat{Y} \neq G \mid A = 1)$. A low selective error rate can be obtained by declining difficult cases; compare risk and coverage together.

Suppose m of N referenced items have usable predictions, and k are correct. Without assumptions on the remaining predictions, the fraction that would be correct after completing all items is bounded by

$$\frac{k}{N} \;\leq\; \mathrm{Acc}_{\mathrm{completed}} \;\leq\; \frac{k + N - m}{N}. \tag{3.3}$$

These are logical bounds, not confidence limits. Selective accuracy k/m and the pessimistic convention k/N answer different questions. Report the chosen convention and reasons for missingness rather than silently switching between them.

Chapter 4

Agreement and aggregation

4.1 Agreement is observable without a gold standard

On items labeled by two raters A and B, observed agreement is $p_o = n^{-1} \sum_i \mathbf{1}\{\widehat{y}_{iA} = \widehat{y}_{iB}\}$. Using their empirical marginal class proportions, define $p_e = \sum_k p_{Ak} p_{Bk}$. Cohen's kappa [1] is

$$\kappa = \frac{p_o - p_e}{1 - p_e}. \tag{4.1}$$

It is undefined when $p_e = 1$. The baseline comes from independently pairing labels with those marginals; it is not a claim that real raters guess randomly. Kappa depends on class prevalence and marginal behavior. Report the marginals and observed agreement alongside it rather than applying a universal threshold.

Neither kappa nor agreement identifies correctness without a reference. Two models can consistently make the same mistake. In our example, all three models agree on 93 messages, including two disagreements with the corpus reference. Models' common training data, prompt conventions, and missing context can create correlated errors. Agreement is useful as a review signal, not a substitute for validation.

4.2 Majority vote and ties

For the available raters $\mathcal{A}_i$ on one item and question, let

$$v_{ik} = \sum_{r \in \mathcal{A}_i} \mathbf{1}\{\widehat{y}_{ir} = k\}, \qquad \widehat{y}_i^{\text{vote}} \in \arg\max_k v_{ik}. \tag{4.2}$$

Urteil implements unweighted majority vote over explicitly chosen sources. It retains support, source count, missing-source count, and tie status. A tied maximum requires a declared tie policy; alphabetical selection should not secretly decide the scientific answer. With two raters, every disagreement is a tie. If sources are missing, a single available vote can win while having much weaker evidence than agreement among all requested sources.

Deduplicate source IDs. Reimporting the same model output is not a new rater. Do not include the evaluation reference in an ensemble that is then evaluated against that same reference. The resulting gain would partly be leakage from the answer key into the prediction.

4.3 Why more votes need not mean more information

For an odd number R of independent binary raters, each with the same error probability $e < 1/2$, the majority-vote error probability is

$$\Pr(\text{vote wrong}) = \sum_{j=(R+1)/2}^{R} \binom{R}{j} e^j (1-e)^{R-j}. \tag{4.3}$$

For $R = 3$ and $e = 0.1$, this is $3(0.1)^2(0.9) + (0.1)^3 = 0.028$. The independence assumption does substantial work: if all three raters make identical mistakes, majority vote still has error 0.1.

For exchangeable error indicators with variance $e(1-e)$ and common pairwise correlation ρ, the variance of their mean is

$$\operatorname{Var}(\overline{E}) = \frac{e(1-e)}{R}\{1 + (R-1)\rho\}. \tag{4.4}$$

The variance-equivalent independent count is $R_{\text{eff}} = R/[1 + (R-1)\rho]$. This is an intuition for the value of diversity, not a general formula for ensemble accuracy or a guarantee that all correlation structures are feasible. Repeating a deterministic request may contribute almost no independent information, especially if it reuses a cached response.

Chapter 5

Quantify uncertainty without overstating it

5.1 A binomial interval and its assumptions

For independent Bernoulli successes with common probability p, observe k successes in n trials and let $\widehat{p} = k/n$. Inverting a normal score test gives the Wilson interval [2]:

$$\frac{\widehat{p} + z^2/(2n) \;\pm\; z\sqrt{\widehat{p}(1-\widehat{p})/n + z^2/(4n^2)}}{1 + z^2/n}. \tag{5.1}$$

Use $z \approx 1.96$ for the usual nominal 95% interval. Unlike the elementary Wald interval, it does not collapse to zero width after observing all successes or all failures. Its frequentist coverage refers to repeated samples under the assumed model, not a 95% posterior probability for the particular interval.

For a fixed class stratum, successes can mean correct labels and trials can mean the sampled reference items in that class. The worked tables illustrate intervals for ham recall and spam recall separately, each with denominator 50. They are not a confidence interval for an unknown deployment accuracy. Sampling without replacement, duplicate templates, and time dependence can require different uncertainty calculations. A balanced stratified design also requires appropriate class weights for a target overall accuracy.

5.2 Compare models on paired items

Separate confidence intervals for two model accuracies discard information: each model labeled the same items. Let b count items on which A is wrong and B is right, and c count items on which A is right and B is wrong. The accuracy difference is $(b - c)/n$. Items where both are right or both are wrong contribute zero to this difference.

Under the conditional equal-error null in the McNemar framework [3], $b \mid (b+c) \sim \text{Binomial}(b+c, 1/2)$. A two-sided exact p-value is

$$p_{\text{exact}} = \min\left\{1,\; 2\sum_{j=0}^{\min(b,c)} \binom{b+c}{j} 2^{-(b+c)}\right\}. \tag{5.2}$$

For $b + c = 0$, define $p = 1$. This doubled-tail convention is the one implemented by the manual helper; it is not a mid-p approximation.

GPT-5 versus GPT-4.1 has three wins and zero losses, so the calculation is $2(1/2)^3 = 0.25$. The observed improvement is real for these saved rows; the sample supplies limited evidence for a broader superiority claim. The exact calculation still assumes independent pairs and the appropriate symmetry under the null. It is illustrative here, not a preregistered confirmatory test.

If many prompts and models were tried before selecting a winner, the selected comparison is subject to search and multiple-testing effects. The generated pairwise table reports raw p-values for teaching, without a multiplicity correction. Do not select the smallest entry and present it as an isolated, prespecified hypothesis test. A fresh evaluation set is often more informative than further refinement of a retrospective p-value.

5.3 Bootstrap the unit that carries dependence

For metrics such as macro F1, a paired bootstrap can resample items with replacement and recompute the complete metric difference on each resample:

$$\Delta^{(t)} = T(g^{(t)}, \widehat{y}_B^{(t)}) - T(g^{(t)}, \widehat{y}_A^{(t)}). \tag{5.3}$$

Use the same resampled IDs for both models. Resample within strata to preserve a stratified design when that is the target, or resample whole clusters when messages belong to conversations or senders. Percentiles of replicate values are a common interval construction, but their validity depends on the design, regularity, and enough independent sampling units.

The manual does not implement a bootstrap command or report invented bootstrap intervals. More resamples reduce Monte Carlo noise; they do not compensate for few original clusters. A bootstrap over fixed observed model outputs also does not measure variability from rerunning the model or uncertainty from choosing the rubric. Those require additional design and repeated measurements.

Chapter 6

Connect labels to decisions and cost

6.1 Probability scores require their own evaluation

A hard label and an explanation do not constitute a calibrated probability. If an external workflow supplies $p_i \in [0, 1]$ for binary outcomes g_i, useful proper scoring rules include the Brier score and log loss:

$$B = \frac{1}{n} \sum_i (p_i - g_i)^2, \tag{6.1}$$

$$L = -\frac{1}{n} \sum_i \{g_i \log p_i + (1 - g_i) \log(1 - p_i)\}. \tag{6.2}$$

Smaller is better. Log loss is infinite for a confidently wrong endpoint; clipping probabilities changes the evaluated rule and must be documented. Calibration asks whether $\mathbb{E}[G \mid p] = p$. It is separate from the ability to rank spam above ham. Binned calibration estimates are noisy and depend on the bins, particularly in small samples.

The SMS experiment contains categorical answers, not calibrated probabilities. It cannot support a calibration curve or a probability threshold sweep. These formulas describe an extension a researcher could evaluate; the current classification CLI does not compute these probability scores.

6.2 Choose actions using error costs

Suppose the conditional probability of spam is p, a false positive costs C_{FP}, a false negative costs C_{FN}, and correct decisions cost zero. Flagging has expected loss $C_{\mathrm{FP}}(1 - p)$; allowing has expected loss $C_{\mathrm{FN}}p$. Flag when

$$p > \frac{C_{\mathrm{FP}}}{C_{\mathrm{FP}} + C_{\mathrm{FN}}}. \tag{6.3}$$

If a false spam flag is especially costly, the threshold rises. F1 does not encode these costs or establish that this policy is optimal. The derivation also requires p to be meaningful for the target population.

An idealized review action with cost C_R and a perfectly correct reviewer would have loss C_R. Choose the least of the three expected losses. Real reviewers are fallible, so their expected downstream

error cost must be added. A pipeline that sends only model disagreements for review can still miss shared errors, as our unanimous reference disagreements demonstrate.

6.3 Account for inference and human effort

For a per-million input price a and output price b, a simplified token cost is

$$C = 10^{-6} \sum_{j} \left[a T_{\text{in},j} + b(T_{\text{visible},j} + T_{\text{reasoning},j}) \right]. \tag{6.4}$$

Use provider-specific cached-input and reasoning treatment where applicable. Count retries and repeated interviews as well as successful final answers. An estimated cost that omits reasoning is not an upper bound. Job-status amounts may be rounded differently from the sum of token metadata; record both when available instead of asserting more billing precision than the source supplies.

The GPT-5 run cost \$0.1847 according to job status, versus \$0.0428 for GPT-4.1. It corrected three further reference disagreements, but that small denominator is not a stable forecast of cost per error avoided. Model cost alone also omits rubric development, human review, integration, and the loss from incorrect decisions. An expensive model can be economical if it reduces review burden; a cheap model can be preferable when its errors are harmless or easily checked.

Chapter 7

The reproducible SMS experiment

7.1 What is empirical and what the build recomputes

The UCI SMS Spam Collection [4] combines published spam and legitimate messages. Our preparation script selected 50 messages per reference class with seed 1729, kept source line numbers as IDs, and restored source order. The original CSV checksum and selected IDs were recorded before inference.

The fixed prompt was:

> Is this SMS unsolicited promotional spam or a legitimate personal/service message (ham)?

The model also received the message text and the answer options ham and spam. No reference labels entered the scenarios. A separate rule classified messages as spam when they contained one of `free,prize,claim,winner`; its output provides a deliberately simple baseline.

Each model ran a 20-message pilot with seed 1729 and then the other 80. The rubric stayed fixed. GPT-4o mini and GPT-4.1 used temperature 0. GPT-5 used the adapter's medium reasoning default and temperature 1; its returned model ID was `gpt-5-2025-08-07`. Thus the comparison fixes task and items, while using each model's runnable settings. It is not a claim of identical sampling parameters.

The book packages `sms_labels.csv`: 100 reference labels and four sets of predictions, with no message text. `sms_provenance.json` records the source CSV and exported-label checksums, configuration, reported costs, and hashes of verified run manifests' result and plan inputs. These records are a derived teaching snapshot. Full Jobs and Results remain in the original local project; the snapshot alone cannot reverify every response against an originating job.

At build time, `classification_scores` recomputes all classification tables from those row-level labels. Separate, small teaching helpers compute Wilson intervals, paired comparisons, and kappa. `calculations.json` stores the values before formatting and `build-manifest.json` fingerprints the build. No external execution occurs.

7.2 Generated tables

The first two tables describe the saved sample. The pairwise tests and recall intervals below are illustrative calculations under the assumptions in Chapter 5.

Source	Accuracy	Macro F1	Spam precision	Spam recall	Cost ($)
Keywords	0.70	0.682	0.885	0.46	0.0000
GPT-4o mini	0.91	0.910	0.860	0.98	0.0033
GPT-4.1	0.95	0.950	0.925	0.98	0.0428
GPT-5	0.98	0.980	0.980	0.98	0.1847

Source	TP	FP	FN	TN
Keywords	23	3	27	47
GPT-4o mini	49	8	1	42
GPT-4.1	49	4	1	46
GPT-5	49	1	1	49

Model A	Model B	B wins	B losses	Exact p	κ
GPT-4o mini	GPT-4.1	4	0	0.1250	0.919
GPT-4o mini	GPT-5	7	0	0.0156	0.860
GPT-4.1	GPT-5	3	0	0.2500	0.940

Source	Reference class	Illustrative 95% recall interval
Keywords	ham	[0.838, 0.979]
Keywords	spam	[0.330, 0.596]
GPT-4o mini	ham	[0.715, 0.917]
GPT-4o mini	spam	[0.895, 0.996]
GPT-4.1	ham	[0.812, 0.968]
GPT-4.1	spam	[0.895, 0.996]
GPT-5	ham	[0.895, 0.996]
GPT-5	spam	[0.895, 0.996]

7.3 Interpret the pattern of errors

Every LLM found 49 of the 50 reference spam messages. The gains came from reducing false spam flags: eight for GPT-4o mini, four for GPT-4.1, and one for GPT-5. GPT-4.1 corrected four earlier errors and GPT-5 corrected three more; neither introduced a new reference disagreement relative to its predecessor on this subset. This nesting is observed here, not a general property of larger or newer models.

All three models chose ham for a humorous observation that the corpus calls spam, and spam for a free-SMS contact notification that the corpus calls ham. The mismatch could involve missing context, a different operational definition, or annotation error. Model consensus cannot decide among these explanations. The saved reference remains unchanged, and the disagreements are available for review in the example's CSV outputs.

The public corpus may have appeared in model training. Prompt development also benefited from inspecting a familiar task. These facts limit claims about fresh data. The next useful scientific step is a frozen rubric, a separately sampled or newly collected evaluation set, and a design that represents the intended language, time period, and decision context.

Chapter 8

From mathematics to an auditable workflow

8.1 A minimal command sequence

Start with a project whose CSV has stable IDs and a reference column. The following commands illustrate setup; choose a new project directory and the actual paths for your data.

```
urteil init study --csv messages.csv \
  --id-column message_id --columns-used text
cd study
# Edit urteil.yaml: question name spam, options ham/spam,
# explicit model, and text as the only prompt field.
urteil validate
urteil next
urteil gold import ../messages.csv --id-column message_id \
  --label-columns gold_label:spam --name human_gold
urteil next
urteil generate --job first_model
urteil next
urteil sample --n 20 --seed 1729
urteil next
urteil plan sample --latest
```

The returned plan contains the exact `ep` execution command and its paired `record_after.argv`, including the originating plan path. Inspect the Jobs, obtain the required execution approval, and run with `ep`. Urteil owns specification and bookkeeping; it never executes inference itself.

After recording, use the returned source ID directly. Do not export answers and reimport them as if they were an independent human rater.

```
urteil preview latest --human
urteil metrics classification --reference human_gold \
  --candidate sample_run --question spam \
  --positive-label spam --zero-division 0
urteil next
```

Here `sample_run` is only an example: repeated runs and multiple raters produce other IDs. Record

and inspect coverage, not just the headline score. Review the sample before approving its reuse. Obtain user approval for the full run, then use `urteil approve sample` and `urteil plan full -reuse-approved-sample`. Execute and record against that full plan. It contains only remaining rows.

8.2 Export, review, and preserve

```
urteil export --final-label-source full_run \
  --output data/cooked/final_labels.csv
urteil next
urteil report context
urteil next
```

Again, substitute the actual source returned by recording. The export joins on IDs, preserves original rows and columns, appends label columns, and creates coverage artifacts. A new output path preserves earlier exports. The bounded report context supplies evidence for final prose; it is not itself a final scientific report.

Keep the source data, archived plans, Results, and normalized labels. Use the CLI for audit-store mutations. A changed rubric, dataset, or model calls for a new reviewed sample and plan; do not retrofit a historical result to a new contract. Reading `urteil next` after each stage makes the outstanding work explicit, but a suggested action may already be satisfied by an equivalent output under another path. Inspect state before creating duplicate work.

8.3 Implementation boundaries

Operation	Available here
Per-class, macro, micro, weighted scores	Metrics CLI; count based, one question
Reference and candidate overlap	Metrics CLI; unmatched IDs retained
Dataset and label-column coverage	Joined export and coverage artifacts
Unweighted majority vote and tie accounting	Aggregation CLI over explicit sources
Kappa, Wilson recall intervals, exact paired tests	Manual teaching helpers only
Population weighting, bootstrap, calibration	Mathematical guidance; no CLI implementation
Model execution and account billing	External `ep`, never Urteil

Before reporting a model as successful, state the rubric, reference, population, coverage, confusion counts, cost, and limitations. Distinguish observed agreement from the true construct, descriptive comparisons from inference, and consistent artifacts from externally valid conclusions. That is the practical purpose of the mathematics in this book.

Chapter 9

Design annotation as a measurement study

9.1 What the literature changes

Empirical evidence shows that LLMs can perform annotation well on particular tasks [5]; practical guidance emphasizes explicit instructions and validation [6]. This supports evaluating a concrete rubric on the data of interest. It does not establish a universal ranking of humans and models.

For research, the next question is what happens when those annotations become variables in a substantive analysis. Ludwig, Mullainathan, and Rambachan [8] distinguish prediction from estimation. Egami and coauthors [9] show how high prediction accuracy can coexist with biased estimates and invalid confidence intervals. Errors correlated with an explanatory variable can matter much more than their overall frequency suggests.

Write a predicted outcome as $\widehat{Y} = Y + E$. In a simple regression with an intercept, the population slope using the predicted outcome satisfies

$$\frac{\mathrm{Cov}(X, \widehat{Y})}{\mathrm{Var}(X)} = \frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(X)} + \frac{\mathrm{Cov}(X, E)}{\mathrm{Var}(X)}. \tag{9.1}$$

High classification accuracy alone does not bound the last term tightly enough to justify ignoring it. Errors in explanatory variables require their own analysis; this outcome-error identity is not a general regression correction.

9.2 Independent human judgments have distinct provenance

Schroeder, Roy, and Kabbara [12] find that presenting model suggestions can change human annotations and inflate subsequent evaluations using those assisted annotations as references. An independent validation design obtains human judgments before exposing model labels or rationales.

Urteil's `validation sample` command archives the full prediction table before creating a review packet. The packet contains only selected input fields, row IDs, the rubric, and blank responses. The caller must inspect those fields and the rubric for embedded answers; filtering columns alone cannot guarantee blinding. Only the review packet should be shared with annotators.

At import, record the annotation mode. The choices are `independent-human` and `model-assisted`, as well as `adjudicated` and `existing-reference`. These are declared properties of the annotation process, not facts the software can independently verify. Only the first is eligible for the corrected estimation command. All four can remain useful sources for descriptive comparisons.

Independent judgments are still fallible. In subjective tasks, disagreement may encode perspective rather than noise [13]. Preserve per-rater labels and define whose judgments the measurement is intended to represent. The sampling intervals below do not measure uncertainty about that construct.

9.3 Prompt sensitivity is a separate experimental dimension

Meaning-preserving rewordings and reordered answer options can change labels; Barrie and coauthors [7] study this as prompt stability. Keep the question, label meanings, model settings, and item text fixed when isolating prompt variation. Repeated unchanged prompts instead measure repeatability; different models change another factor. Vary wording and option order separately if the purpose is to attribute effects to either one.

For an item labeled by all R variants, let c_{ik} count votes for category k. Urteil reports the following descriptive quantities:

$$U_i = \mathbf{1}\{\max_k c_{ik} = R\}, \tag{9.2}$$

$$V_i = 1 - \frac{\max_k c_{ik}}{R}, \tag{9.3}$$

$$A_i = \frac{\sum_k c_{ik}(c_{ik} - 1)}{R(R-1)}. \tag{9.4}$$

They are unanimity, variation ratio, and agreement across unordered variant pairs. For labels A, A, B, these equal 0, 1/3, and 1/3. Summary values average over the same complete items. Incomplete items retain their labels and missing-source counts, but receive no item stability score. A single available label must not masquerade as agreement among a complete panel.

`metrics stability` implements these transparent diagnostics, not the published Prompt Stability Score (PSS). Verified sources supply prompt and model metadata directly from archived jobs and results; imported sources require explicit declarations. Agreement remains distinct from accuracy. For paired judgments of generated answers, the LLM-as-judge literature also identifies position and verbosity effects [14]; their importance for a particular annotation task is empirical.

Chapter 10

Correct population estimates with a probability sample

10.1 A fixed-frame generalized difference estimator

Let the target be a fixed dataset of N items. For each item, fix a prediction f_i before drawing the validation sample. Let y_i denote the reference judgment under a fixed rubric and define

$$\mu_N = \frac{1}{N}\sum_{i=1}^{N} y_i, \qquad e_i = y_i - f_i. \tag{10.1}$$

Draw S by simple random sampling without replacement, with fixed size n. Each item has inclusion probability $\pi_i = n/N$. The estimator is

$$\widehat{\mu}_{\text{diff}} = \frac{1}{N}\sum_{i=1}^{N} f_i + \frac{1}{n}\sum_{i\in S}(y_i - f_i). \tag{10.2}$$

The first term is known for the entire frame. The second estimates its average error. For a category k, use $y_i = \mathbf{1}\{\text{reference}_i = k\}$ and $f_i = \mathbf{1}\{\text{prediction}_i = k\}$ to estimate class prevalence.

The randomization is the selection of S. Since $\mathbb{E}_S[\mathbf{1}\{i \in S\}] = n/N$,

$$\mathbb{E}_S[\widehat{\mu}_{\text{diff}}] = \overline{f}_N + \frac{1}{n}\sum_i \frac{n}{N} e_i = \overline{f}_N + \overline{e}_N = \mu_N. \tag{10.3}$$

Thus biased model predictions do not themselves create design bias. Fixed predictions are essential: tuning a prompt after examining these references can make the residuals depend on the selected sample and invalidate this argument. Incomplete annotation also breaks the stated design if difficult items are silently dropped.

10.2 Relation to PPI and design-based supervised learning

Prediction-powered inference [10] combines predictions with reference observations to support statistical inference. A mean estimator in its separate labeled/unlabeled-sample setting is

$$\widehat{\mu}_{\text{PPI}} = \frac{1}{M}\sum_{i=1}^{M} f(X_i^U) + \frac{1}{n}\sum_{j=1}^{n}\{Y_j^L - f(X_j^L)\}. \tag{10.4}$$

The population and sampling assumptions determine the variance calculation. In particular, its random unlabeled sample contributes uncertainty; Urteil's full archived frame in Equation 10.2 is fixed.

Design-based supervised learning [9] supplies a broader framework for using imperfect predicted variables in downstream analysis. With known unequal inclusion probabilities, a related mean correction has the form

$$\overline{f}_N + \frac{1}{N}\sum_{i\in S}\frac{y_i - f_i}{\pi_i}. \tag{10.5}$$

Its variance depends on the sampling design, including joint inclusion probabilities. Urteil currently implements only fixed-size SRS means and class prevalences. It does not implement weighted designs, regression correction, cross-fitting, or optimized PPI weights. These equations describe the relationship to the literature rather than claiming the full frameworks are implemented.

10.3 Estimated variance and a conservative confidence interval

Let $\overline{e}_S$ be the sample residual mean and

$$s_e^2 = \frac{1}{n-1}\sum_{i\in S}(e_i - \overline{e}_S)^2. \tag{10.6}$$

For SRS with $n \geq 2$, an unbiased estimator of design variance is

$$\widehat{\mathrm{Var}}_S(\widehat{\mu}_{\mathrm{diff}}) = \left(1 - \frac{n}{N}\right)\frac{s_e^2}{n}. \tag{10.7}$$

The finite-population correction follows because sampling without replacement reduces uncertainty, reaching zero when the entire frame is annotated. With enough residual variation and an appropriate finite-population normal approximation, one can use $\widehat{\mu}_{\mathrm{diff}} \pm z\,\widehat{\mathrm{SE}}$. Urteil exposes this as an *approximate* interval. Zero observed residual variance in a noncensus sample suppresses this interval, because a sample with no errors does not establish perfect population predictions.

For an alternative that does not estimate variance, declare bounds $y_i, f_i \in [L, U]$ that apply throughout the frame. Then $e_i \in [-(U-L), U-L]$. Hoeffding's inequality for sampling without replacement [11] gives

$$\Pr_S\{|\overline{e}_S - \overline{e}_N| \geq t\} \leq 2\exp\left\{-\frac{nt^2}{2(U-L)^2}\right\}. \tag{10.8}$$

Set the right side to α to obtain half-width

$$h = (U-L)\sqrt{\frac{2\log(2/\alpha)}{n}}. \tag{10.9}$$

The default reported confidence set is $[\widehat{\mu}_{\mathrm{diff}} - h, \widehat{\mu}_{\mathrm{diff}} + h] \cap [L, U]$. This bound is conservative and may be wide. At a census, use the exact reference mean instead. The untruncated point estimate can fall outside $[L, U]$; Urteil retains it with a warning to preserve the distinction between the estimator and the known parameter space. An empty intersection is represented explicitly.

These are fixed-sample statements. Bounds chosen from the observed sample, selective stopping, human error, or transport to a different population are not covered. In particular, the prevalence of spam in our balanced 100-message frame is not the prevalence of spam among incoming messages.

10.4 An offline arithmetic illustration

For teaching, take the saved GPT-4o mini predictions on the 100-message frame, draw 20 row IDs by SRS with seed 1729, and reveal their existing corpus labels. The following table is regenerated by the same mean estimator used by the CLI:

Quantity	Value
Full-frame prediction mean	0.5700
Reference sample mean	0.4500
Residual correction	-0.1000
Corrected estimate	0.4700
Estimated standard error	0.0616
Known frame reference mean	0.5000
Conservative 95% interval	[0.0000, 1.0000]

This is a retrospective calculation using an existing benchmark, not newly collected independent human validation. The complete frame reference mean is known here and included as a check. The calculations file retains selected IDs, the seed, and that provenance distinction. The production estimation command does not accept `existing-reference` imports as independent validation.

10.5 Commands and auditable evidence

```
urteil docs show literature
urteil docs show validation
urteil validation sample --candidate full_run \
  --question topic --n 100 --seed 1729
urteil next
# Give only the returned review packet to the annotator.
# Import the completed packet with the returned sample path.
urteil validation import .urteil/validation/sample_<id> \
  --path data/review/sample_<id>/review.csv \
  --rater alice_validation --annotation-mode independent-human
urteil estimate prevalence \
  --sample .urteil/validation/sample_<id> \
  --reference alice_validation --positive-label Sports
urteil next
urteil report context
```

For a numeric rubric, use `estimate mean` with declared `-lower` and `-upper` bounds. Estimates remain bound to their archived frame even if the working dataset or rubric later changes. The report handoff includes sampling assumptions and provenance alongside the numbers.

Chapter 11

Stress-test estimators with known truth

11.1 A reproducible fixed-population experiment

The simulation study fixes six synthetic populations of $N = 1000$ items and varies prevalence, sensitivity, and false-positive rates. It draws $n \in \{10, 25, 50, 100, 250, 500, 1000\}$ reference items by SRS without replacement, repeating each cell 1000 times. Both reference-based methods use the same selected IDs on every draw. Seed 1729 and cell-specific random streams make the study reproducible independently of the order of requested sample sizes.

Reference labels are known truth by construction. Class and error counts are rounded to integers before randomizing their identities, and the resulting frame is held fixed across replicates. This examines sampling variation in validation, not uncertainty in human judgment, changing prompts, or deployment.

Compare three estimators: the full-frame model mean, the human-only sample mean, and the generalized difference estimator. The first can have fixed bias and does not receive an invented confidence interval. Human-only sampling has estimated variance $(1-n/N)s_y^2/n$; correction replaces s_y^2 by s_e^2. Their relative precision therefore depends on residual variance, rather than classification accuracy alone.

11.2 Bias, precision, and interval availability

For replicate errors $d_b = \widehat{\mu}_b - \mu_N$, the study reports

$$\widehat{\text{Bias}} = \frac{1}{B}\sum_b d_b, \qquad \widehat{\text{RMSE}} = \sqrt{\frac{1}{B}\sum_b d_b^2}. \tag{11.1}$$

Bias Monte Carlo standard error is $s_d/\sqrt{B}$. Estimated interval coverage is the fraction of available intervals containing the known mean; its plug-in Monte Carlo standard error is $\sqrt{\widehat{c}(1-\widehat{c})/B_a}$, where B_a counts available intervals. A reported zero MCSE after observing no failures does not establish exact coverage.

Always report availability with conditional coverage. Noncensus normal intervals are

withheld when observed variance is zero. Dropping these draws without reporting them can give a misleading impression of performance. The tables below give point-estimate results at $n = 100$ and interval results at several sizes. All proportions and RMSEs are in units from zero to one.

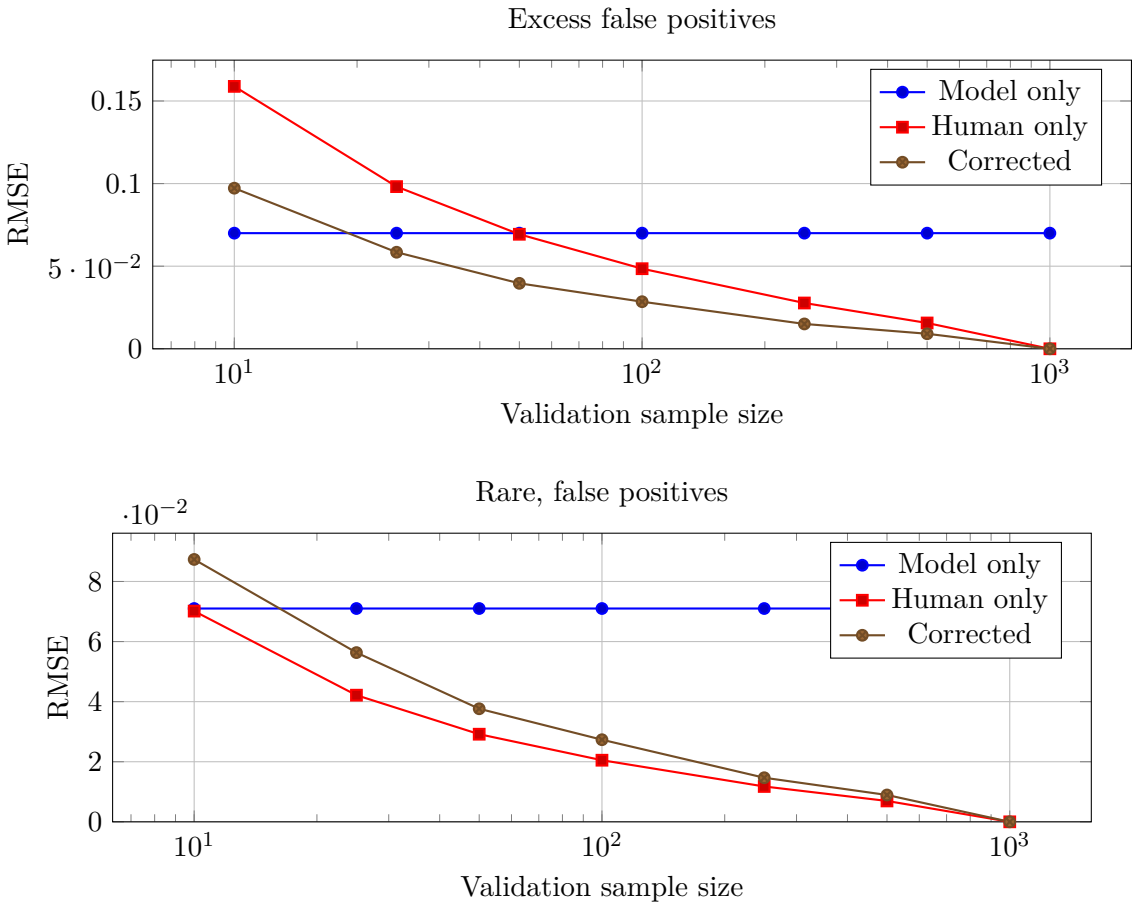
Scenario	Accuracy	True share	Model share	Human RMSE	Corrected RMSE
Balanced, good	0.950	0.500	0.500	0.0471	0.0213
Excess false positives	0.910	0.500	0.570	0.0485	0.0285
Excess false negatives	0.840	0.500	0.360	0.0467	0.0347
Rare, false positives	0.919	0.050	0.121	0.0205	0.0273
Uninformative	0.500	0.500	0.500	0.0467	0.0655
Perfect	1.000	0.500	0.500	0.0489	0.0000

Scenario	Method	n	Availability	Normal coverage	Bound coverage
Excess false positives	Human	25	1.000	0.959	0.995
Excess false positives	Corrected	25	0.919	0.960	1.000
Excess false positives	Human	100	1.000	0.946	0.995
Excess false positives	Corrected	100	1.000	0.933	1.000
Excess false positives	Human	500	1.000	0.950	1.000
Excess false positives	Corrected	500	1.000	0.952	1.000
Rare, false positives	Human	25	0.724	1.000	1.000
Rare, false positives	Corrected	25	0.877	0.961	1.000
Rare, false positives	Human	100	0.995	0.892	1.000
Rare, false positives	Corrected	100	1.000	0.938	1.000
Rare, false positives	Human	500	1.000	0.936	1.000
Rare, false positives	Corrected	500	1.000	0.945	1.000

In the rare-class case, a classifier with 91.9% accuracy predicts a share of 12.1% when the true share is 5%. Correction removes its design bias, but has larger variance than human-only sampling in this constructed case. Symmetric errors in the uninformative case instead cancel in the full-frame mean even though half the individual predictions are wrong. Neither result supports treating accuracy as a sufficient criterion for prevalence estimation.

The human-only conservative interval uses the binary range of one; the corrected interval uses the residual range of two. These bounds do not adapt to prediction quality. They may be wide even when a useful predictor substantially improves point-estimate precision. Normal intervals can be much narrower but under-cover, particularly with small samples and rare outcomes.

11.3 How sample size changes the tradeoff



The horizontal model-only line reflects fixed measurement bias in a fixed frame. At a census, both reference-based estimators recover the reference mean exactly. Analytical design bias and standard errors accompany empirical results in the CSV, providing checks independent of Monte Carlo noise.

```
urteil docs show simulation
urteil simulate --output-dir simulation-study --replicates 1000
```

This produces synthetic frames, the design, summary statistics, and file/source hashes. It neither calls a model nor creates a human reference source.

Chapter 12

A controlled prompt-sensitivity experiment

12.1 Hold the task and inputs fixed

Four GPT-4o mini variants label the same 100 SMS messages used earlier, with temperature zero and one answer per item and variant. The original prompt is compared with two rewordings and with reversed answer order. The rewordings keep `ham, spam`; the order variant keeps the original wording and presents `spam, ham`. Definitions continue to distinguish unsolicited promotional messages from legitimate personal or service messages.

Each variant starts with the same five-message pilot, followed by the remaining 95 messages. Pilot answers are reused exactly once. Urteil builds the jobs; `ep` performs execution. All model scenarios contain only the ID and message text. Existing corpus labels are used only for evaluation. The experiment was fixed before examining the new variant outputs.

Variant	Accuracy	Spam F1	Changes	Missing cache flags	Cost*
Original	0.91	0.916	0	100	0.0000
Reworded	0.94	0.942	3	100	0.0030
Directive	0.92	0.922	5	100	0.0028
Reversed options	0.91	0.916	2	100	0.0032

*Rounded USD reported by the completed jobs, not an invoice.

All variants agree on 94 of 100 messages; 6 of those unanimous labels disagree with the corpus reference.

Changes are counted against the original prompt on matched IDs. Default cache behavior was allowed. These Results artifacts omit per-answer cache flags, so cache status remains unknown. The original jobs report zero cost, consistent with cache reuse, but this does not identify per-answer cache status. A cached answer is not a new independent response. The four-source comparison is descriptive. With one output per prompt/item cell, it does not separately estimate stochastic response variation or support a general model/prompt ranking. The balanced public frame also limits deployment claims and may overlap model training data.

12.2 Derive controls from archived jobs

```
urteil metrics stability --question spam --sources \
prompt_original,prompt_reworded,prompt_directive,prompt_reversed
urteil next
urteil report context
```

The new `-sources` path requires verified originating plans. It checks job, rubric, plan, result, and normalized-label hashes, reconstructs answers from the results, and follows any approved pilot reuse. The comparison verifies the actual recorded model parameters, agent and iteration, other survey controls, and per-item scenario hashes. Changes to current working files cannot silently rewrite the historical prompt descriptions.

If a source is imported without a verified plan, an explicit `-variants` declaration remains available and is marked as caller-declared. If archived evidence exists, conflicting declarations are rejected. Missing or changed verified artifacts cause an error rather than a silent downgrade. The full evidence remains linked from the comparison artifact.

These checks verify recorded execution controls. They do not establish semantic equivalence of prompts or identify unobserved changes behind a provider's model alias. The calling researcher retains responsibility for the scientific design.

Bibliography

[1] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20, 37–46. https://doi.org/10.1177/001316446002000104.

[2] Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22, 209–212. https://doi.org/10.1080/01621459.1927.10502953.

[3] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12, 153–157. https://doi.org/10.1007/BF02295996. The book uses the conditional exact binomial form, not the asymptotic chi-squared approximation.

[4] Almeida, T. and Hidalgo, J. (2011). *SMS Spam Collection*. UCI Machine Learning Repository. https://doi.org/10.24432/C5CC84. Reference labels are redistributed under the dataset's stated CC BY 4.0 license; IDs retain original line positions. Message text is omitted from the manual snapshot. Selection and model-output provenance accompany the packaged CSV.

[5] Gilardi, F., Alizadeh, M., and Kubli, M. (2023). ChatGPT outperforms crowd workers for text-annotation tasks. *PNAS*. https://doi.org/10.1073/pnas.2305016120.

[6] Törnberg, P. (2024). Best Practices for Text Annotation with Large Language Models. *Sociologica*. https://sociologica.unibo.it/article/view/19461.

[7] Barrie, C., Palaiologou, E., and Törnberg, P. (2024; revised 2026). Prompt Stability Scoring for Text Annotation with Large Language Models. https://arxiv.org/abs/2407.02039.

[8] Ludwig, J., Mullainathan, S., and Rambachan, A. (2025 working paper). Large Language Models: An Applied Econometric Framework. NBER Working Paper 33344. https://www.nber.org/papers/w33344.

[9] Egami, N., Hinck, M., Stewart, B. M., and Wei, H. (2026 manuscript). Using Large Language Model Annotations for the Social Sciences: A General Framework of Using Predicted Variables in Downstream Analyses. https://naokiegami.com/paper/dsl_ss.pdf.

[10] Angelopoulos, A. N., Bates, S., Fannjiang, C., Jordan, M. I., and Zrnic, T. (2023). Prediction-Powered Inference. *Science*, 382, 669–674. https://arxiv.org/abs/2301.09633.

[11] Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58, 13–30. https://doi.org/10.1080/01621459.1963.10500830.

[12] Schroeder, H., Roy, D., and Kabbara, J. (2025 preprint). Just Put a Human in the Loop? Investigating LLM-Assisted Annotation for Subjective Tasks. https://arxiv.org/abs/2507.15821.

[13] 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*, 10. https://transacl.org/index.php/tacl/article/view/3173.

[14] Zheng, L. et al. (2023). Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. https://arxiv.org/abs/2306.05685.