Iterative Data Curation with Theoretical Guarantees

Väinö Yrjänäinen
Department of Statistics,
Uppsala University

Johan Jonasson
Department of Mathematical Sciences
Chalmers University of Technology

Måns Magnusson
Department of Statistics,
Uppsala University

Abstract

In recent years, more and more large data sets have become available. Data accuracy, the absence of verifiable errors in data, is crucial for these large materials to enable high-quality research, downstream applications, and model training. This results in the problem of how to curate or improve data accuracy in such large and growing data, especially when the data is too large for manual curation to be feasible. This paper presents a unified procedure for iterative and continuous improvement of data sets. We provide theoretical guarantees that data accuracy tests speed up error reduction and, most importantly, that the proposed approach will, asymptotically, eliminate all errors in data with probability one. We corroborate the theoretical results with simulations and a real-world use case.

1 Introduction

High-quality data is crucial for making good predictions [Jain et al., 2020, Budach et al., 2022], valid statistical inference [Hurtado Bodell et al., 2022, Bernardi et al., 2023, Miller and Spiegel, 2025], decision making [Wang et al., 2024], and training reliable predictive models [Gunasekar et al., 2023]. Moreover, perhaps the most important aspect of data quality is data accuracy [Olson, 2003, p. 27], which can be seen as the absence of erroneous values. In this article, we focus on data accuracy and define errors as discrepancies between the stored values and some verifiable ground truth. These may be OCR errors, yielding a discrepancy between printed physical media and its digital representation; label errors, where an image, piece of text, or other piece of data is labeled as the incorrect category; a discrepancy between a value in a database describing a rental apartment and the real measurements of the apartment; missing or invalid values in a database table; or a number of other ways the stored pieces of data differ from some well-defined and verifiable ground truth.

Despite the importance of data accuracy, data sets commonly used in machine learning, such as Fashion MNIST [Xiao et al., 2017], Common Crawl [Common Crawl, 2024], and the Wikipedia corpus [Wikipedia contributors, 2023], often provide few guarantees of their accuracy. Instead, even the most commonly used data sets contain a notable amount of errors [Northcutt et al., 2021]. This is often the case in research data sets as well, eg. parliamentary data [Gentzkow et al., 2018, Erjavec et al., 2023], legal research datasets [Östling et al., 2023, Butler, 2025], or literature corpora [Davies, 2008] is well-known to contain errors or lack accuracy guarantees.

Improving and verifying the quality of large data sets is a difficult task [Kaddour et al., 2023]. Limited resources force a balance between accuracy, the richness of annotation, and the overall amount of curations that can be made [see Voormann and Gut, 2008, for an early example]. Hence, curation becomes increasingly complex on the large data sets that have become commonplace. Rule-based algorithms, regular expressions, and machine learning methods, are powerful tools for improving data quality on large data sets and can correct many thousands of errors simultaneously [Khirbat, 2017, Zhang et al., 2020, Fante et al., 2021, Yun et al., 2021] or filter out poor-quality data [Dakka et al., 2021]. However, such approaches do not have accuracy guarantees and usually come with non-negligible error rates, essentially introducing new errors to the data. Moreover, augmenting or curating a data set with crowdsourcing, non-expert annotators, and external data sources are also possible approaches to curation. However, these approaches also have a non-negligible amount of errors [Barchard and Pace, 2011, Gao et al., 2016, Brühlmann et al., 2020]. Hence, it is not possible today to curate data at scale efficiently without also risking the introduction of new errors.

1.1 Our Contributions

In this work, we address *data accuracy* in large datasets, particularly textual and tabular data, where full manual inspection is infeasible and automated methods must be both scalable and verifiable. Our main contributions are:

1. **Iterative curation framework:** A structured process for scalable data accuracy improvements in large textual and tabular datasets.
2. **Theoretical guarantees:** We prove the *asymptotic convergence to an error-free dataset*, the *exponential decay of errors* and a guarantee on the *rate of error reduction* on expectation at each revision of the data.
3. **Validation:** We support our results with simulations and a real-world case study on the Swedish Parliamentary Corpus.

2 Iterative Curation of Large Data

The central idea of iterative curation is to incrementally improve data quality over time, as initially described by Voormann and Gut [2008], extended to the context of large data sets and automated curation. We define the dataset as a mutable state that evolves through well-defined revisions and study the theoretical properties of this process.

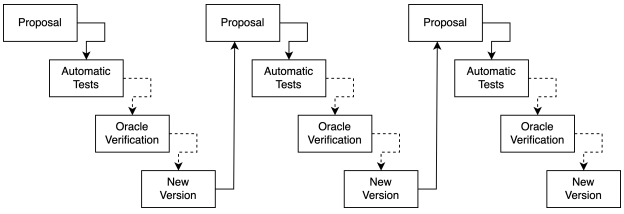


Figure 1: Iterative Data Curation

The process begins with setting up a prototype data set. After this initial step, the remaining three steps, revision, evaluation, and release, are repeated in short cycles (see Figure 1). To ensure that each iteration leads to measurable improvement, we assume we have access to an oracle to assess whether an individual change, or an *edit*, is correct or true. In practice, the oracle is most likely a human manually assessing whether a change is correct.

This formalism connects straightforwardly to version control, automated testing, and semantic versioning—principles familiar from software engineering—to data. Changes to the dataset, *revisions*, become analogous to git *commits* and each release is documented and versioned. Further, we treat the dataset state as an Application Programming Interface (API). The API defines how users and systems interact with the data: retrieving content, querying metadata, or validating data quality and integrity. Semantic versioning [Preston-Werner, 2013] is used to describe the changes to the (data) API by accepted revisions.

Step 1: Set up a Prototype

The process begins with the creation of a minimal prototype dataset, which serves as the initial state of the data, stored in the data format $\mathcal{S}$. At this stage, the primary goal is to define a suitable $\mathcal{S}$—ideally, one that is both human-readable and easily extensible. For example, in the case of a corpus of parliamentary proceedings, a prototype might consist of the full text of a small number of articles encoded in a ParlaClarin TEI XML schema [Erjavec and Pancur, 2021]. This initial prototype should be small enough to allow thorough manual inspection and validation. Furthermore, the format $\mathcal{S}$ can consist of multiple formats, such as text documents and metadata files, i.e., $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$. Once the structure is deemed satisfactory, the initial version of the complete data $S_0 \in \mathcal{S}$ is created.

Step 2: Create a Revision Proposal

Given the initial state S_0, subsequent revision proposals, $R_t \in \mathcal{S}$, are introduced as either (1) additions or (2) corrections, each representing a targeted update to the dataset and implying a proposal R_{t+1} for a new data state S_{t+1}. Additions mean that the data is expanded in different directions of interest, extending the format $\mathcal{S}_{new} := \mathcal{S} \times \mathcal{S}'$, e.g., new speeches or metadata on members of parliament are added in a parliamentary proceedings dataset. Conversely, a correction means that the data state is changed, but the format $\mathcal{S}$ remains unchanged, e.g., when correcting identified errors. Step 2 results in a new revision proposal $R_{t+1} \in \mathcal{S}$ from the previous state S_t.

Step 3: Accept or Reject the Proposal

Once a proposal R_{t+1} for a new state has been created, the next step is determining whether it should be accepted and applied or rejected. This is done in two steps: (1) automated data tests and (2) (oracle) accuracy control of the proposed revision. The automated tests do not, in general, cost much to run, while the (oracle) accuracy control usually entails costly manual work of assessing the correctness of R_{t+1}. Hence, the automated tests are run first, rejecting poor R_{t+1} proposals before the edit sampling and oracle checking.

Step 3.1: Automatic Data Testing Step

Automated testing of proposals can take several forms. Some involve format validation, which ensures that the proposal R_{t+1} adheres to the expected structural specifications; in other words, that $R_{t+1} \in \mathcal{S}$. This may include validating XML, checking that tables contain all required columns, and confirming that no files have been unintentionally removed or added. A second example is sanity checks of content, which evaluate whether specific properties of the data are preserved based on prior knowledge. These may include verifying known summary statistics—such as the total number of members of parliament, or checking that previously fixed errors or corrections have not reappeared.

Step 3.2 Edit Sampling Step

Suppose the new revision passes the automatic data tests. In that case, we control the quality of a new revision by taking a random sample of the individual *edits*, $\Delta_t = \Delta(R_{t+1}, S_t) = u_1, \ldots, u_{|\Delta_t|}$. As the sample is drawn from the set of edits, there needs to be a specific definition of edits for each data format $\mathcal{S}$. For tabular data, we propose uniform sampling from the set of changed entries (i.e. rows); for sequential data, we propose sampling unit additions and deletions from the addition-deletion diff, obtained via the Myers algorithm [Myers, 1986], e.g. using git `diff` [Torvalds et al., 2005] (see Appendix D for details).

The sample of edits u_i of size n is checked through the oracle to assess if a proposed edit u_i is correct or not. A revision is accepted, i.e. $S_{t+1} := R_{t+1}$, if the proportion of edits deemed correct by the oracle is sufficiently high, i.e. over a threshold $m \in \{1, \ldots, n\}$; otherwise the proposed revision is rejected and $S_{t+1} := S_t$.

Step 4: Release a new version

Finally, if a revision R_{t+1} has been accepted, we release a new state, S_{t+1}, of the data with the amended revision. Each release of data states, S_{t+1}, is associated with a version number which reflects the type of change compared to the previous version, S_t. We adapt the semantic versioning by Preston-Werner [2013] for data as follows:

- MAJOR version is increased when changes are made that are not backwards compatible for the API of the data ($\mathcal{S}$ has changed),
- MINOR version when adding functionality, e.g. as new metadata fields or text documents in a backwards-compatible manner ($\mathcal{S}$ has been extended $\mathcal{S}' := \mathcal{S} \times \mathcal{S}_{new}$), and
- PATCH version when you perform fixes or changes that do not introduce new features in the data, e.g. fixing errors in data or adding new data tests ($\mathcal{S}$ remains unchanged).

Similar to standard semantic versioning, only major and minor versioning is followed before the 1.0.0 release and the data is deemed stable upon the 1.0.0 release.

3 Theoretical Guarantees

To study the theoretical properties of the data accuracy under the proposed iterative curation process, we must make assumptions about the data format, the true data and the proposed changes. The analysis relies on the following assumptions

(a) There is a canonical form $\mathcal{S}$ of the data, i.e. there is a one-to-one mapping from what the data represents to its representation $S \in \mathcal{S}$. Further, there is a distance $d : \mathcal{S} \times \mathcal{S} \to \mathbb{N}$ in the space $\mathcal{S}$, and access to the set of edits $\Delta(S, S')$ (where $|\Delta(S, S')| = d(S, S')$) between any two versions or revisions S, S'. The distance to the true data $S^\star$ defines the number of errors $E_t := d(S^\star, S_t)$.

(b) The set of edits $\Delta_t = \Delta(R_{t+1}, S_t)$ between the proposal R_{t+1} and the current state consists of unit modifications $u_1, \ldots, u_{|\Delta_t|}$, which each modify the distances by $\{-1, 0, 1\}$, and are additive. in terms of distance: $d(S_t + u + u', S^\star) - d(S_t, S^\star) = d(S_t + u, S^\star) - d(S_t, S^\star) + d(S_t + u', S^\star) - d(S_t, S^\star)$ for all possible edits $u \neq u'$.

(c) There exists an oracle that can calculate the changes in the distances $d(S_t + u_i, S^\star) - d(S_t, S^\star)$ caused by any single edit $u_i, \forall i \in \{1, \ldots, |\Delta_t|\}$.

(d) The revision size $|\Delta_t|$ is binomially distributed with E_t trials and the success probability λ_t, where λ_t is i.i.d over all timesteps t.

(e) The number of correct edits $\sum_{u \in \Delta_t} \mathbb{I}(d(R_{t+1}, S^\star) - E_t = -1)$ for proposals R_{t+1} is binomially distributed with the accuracy $1 - r_t$. The proposal error rate r_t is distributed i.i.d. over all timesteps t, independently of λ_t, and has a nonzero density $p(r_t)$ whenever $r_t < 0.5$, bounded from below by $a > 0$.

Theorem 3.1. *Given Assumptions (a)-(e), when only proposals with at least m correct modifications out of a sample of n are accepted, provided that the following holds for (n, m)*

$$a > \frac{C(n, m)}{2 + C(n, m)} \tag{1}$$

where $C(n,m)$ is defined as

$$C(n,m) = \frac{2(n+1)(n+2)}{2m-n}\binom{n}{m} \cdot (r_{max}-1/2)(1-r_{max})^m r_{max}^{n-m}$$
$$r_{max} = \frac{3n-2m+2+\sqrt{(2m-n)^2+4(n+1)}}{4(n+1)} \tag{2}$$

the number of errors $\lim_{t\to\infty} P(E_t = 0) \to 1$.

Proof. Given Assumptions (a)–(b), all changes observed in a proposal R_{t+1} can be enumerated, and applying them in any order will yield a difference in distance to the true dataset corresponding to the sum of the distances of them being separately applied to S_t. Thus, we can estimate the difference made in the distance $d(R_t, S^\star) - d(S_t, S^\star)$ by taking a simple random sample (SRS) $u_1, \dots u_n$ of size n from $\Delta(R_{t+1}, S_t)$, and obtaining the distances $d(S_t + u_i, S^\star) - d(S_t, S^\star) \quad \forall i \in \{1, \dots, n\}$ by Assumption (c). The number of correct edits in the sample M is

$$M = \sum_{i=1}^{n} \mathbb{I}(d(S_t + u_i, S^\star) - d(S_t, S^\star) = -1) \tag{3}$$

which is binomially distributed $M \sim Bin(r_t, n)$. The posterior distribution of r_t is thus proportional to $p(M = m \mid r_t)p(r_t) \propto (1-r_t)^m r_t^{n-m} p(r_t)$, where $p(r_t)$ is the prior PDF of r_t.

By assumption (d), $|\Delta_t|$ is independent of the revision error rate r_t. Moreover, as it is binomially distributed $|\Delta_t| \sim B(\lambda_t, E_t)$ conditionally on λ_t, it can be decomposed to a sum of E_t independent Bernoulli random variables. By Assumption (e), the errors given an $|\Delta_t|$ are binomially distributed. E_t can thus be represented as a branching process with variable offspring probabilities $p_{0,t}, p_{1,t}, p_{2,t}$ that are only a function of λ_t and r_t. We obtain a branching process in random environments (BPRE) [Smith and Wilkinson, 1969], where $p_{0,t} = (1-r_t)\lambda_t$, $p_{1,t} = 1-\lambda_t$ and $p_{2,t} = r\lambda_t$.

A BPRE $Z_t \in \mathbb{N}$ where $Z_0 > 0$ is defined as a Markov process where Z_{t+1} is a sum of Z_t i.i.d. random variables $\zeta \in \mathbb{N}$ with probabilities $p_{0,t}, p_{1,t} \dots$ that are parametrized by θ_t which are i.i.d for each t. Moreover, $\lim_{t\to\infty} P(Z_t = 0) = 1$ for all BPREs that have $\mathbb{E}_\theta[\log \mathbb{E}[\zeta]] < 0$ [Smith and Wilkinson, 1969]. In our case, ζ is parameterized by r_t and λ_t. M follows a binomial distribution, thus $m < m' \implies P(r_t \le r \mid M = m) < P(r_t \le r \mid M = m') \quad \forall r \in (0,1)$, and $\mathbb{E}_{r_t}(\mathbb{E}[\zeta] \mid M = m])$ is thus a decreasing function of m. Therefore,

$$\mathbb{E}_{r_t,\lambda_t}[\mathbb{E}[\zeta] \mid M = m] < 1 \tag{4}$$
$$\implies \quad \mathbb{E}_{r_t,\lambda_t}[\log \mathbb{E}[\zeta] \mid M = m] < 0 \tag{5}$$
$$\implies \quad \mathbb{E}_{r_t,\lambda_t}[\log \mathbb{E}[\zeta] \mid M \ge m] < 0 \tag{6}$$

where step (5) uses Jensen's inequality. Consider the expectation $\mathbb{E}_{r_t,\lambda_t}[\mathbb{E}[\zeta] \mid M = m]$

$$\begin{aligned}&\mathbb{E}_{r_t,\lambda_t}[\mathbb{E}[\zeta] \mid M = m] \\ &= \mathbb{E}_{r_t,\lambda_t}\Big[\sum_{k=0}^{2} k \cdot p_{k,t} \mid M = m\Big] \\ &= 1 - 2\mathbb{E}[\lambda_t]\,(1/2 - \mathbb{E}[r_t \mid M = m])\end{aligned} \tag{7}$$

which is less than 1 whenever the $\mathbb{E}[r_t \mid M = m] < 1/2 \iff \mathbb{E}[1/2 - r_t \mid M = m] > 0$, i.e.

$$\int_0^1 (1/2 - r)p(r \mid m)dr \tag{8}$$
$$\propto \int_0^{0.5} (1/2 - r)p(m \mid r)p(r)dr \tag{9}$$
$$+ \int_{0.5}^1 (1/2 - r)p(m \mid r)p(r)dr$$
$$\ge a\Big(\frac{2m-n}{2(n+1)(n+2)}\Big) \tag{10}$$
$$- \max_r \binom{n}{m}\left[(r-1/2)(1-r)^m r^{n-m}\right](1-a/2)$$
$$\propto a - \left(\frac{2(n+1)(n+2)}{2m-n}\binom{n}{m}\cdot \right. \tag{11}$$
$$\left. \max_r \left[(r-1/2)(1-r)^m r^{n-m}\right](1-a/2)\right)$$
$$\propto a - C(n,m)/(2 + C(n,m)) > 0 \tag{12}$$

See Appendix B for a detailed derivation. The maximum of $(r-1/2)(1-r)^m r^{n-m}$ is found at

$$\begin{aligned}r_{max} &= \arg\max_r (r-1/2)(1-r)^m r^{n-m} \\ &= \frac{3n-2m+2+\sqrt{(n-2m)^2+4(n+1)}}{4(n+1)}\end{aligned} \tag{13}$$

and thus $C(n,m)$ becomes

$$C(n,m) = \left(\frac{2(n+1)(n+2)}{2m-n}\binom{n}{m} \cdot (r_{max}-1/2)(1-r_{max})^m r_{max}^{n-m}\right) \tag{14}$$

Details for the derivation of r_{max}, as well as bounds for the value of $C(n,m)$, are provided in Appendix B.2. Finally, the condition for the extinction of E_t becomes

$$a > \frac{C(n,m)}{2 + C(n,m)} \tag{15}$$

which can be attained for any a as $C(n,m)$ can be made arbitrarily small by increasing m and n (details in Appendix B.2). □

Remark 3.2. A subcritical BPRE decays exponentially, both in terms of the mean $\mathbb{E}[Z_t]$ of the process by timestep t [Smith and Wilkinson, 1969] and an upper bound for the survival probability $P(Z_t > 0)$ [Guivarc'h and Liu, 2001]. The decay happens wrt. the mean of the offspring probabilities $\mu = \mathbb{E}[\zeta]$, i.e. $E_t \sim \mu^t$, while μ is determined by the posterior distribution of r_t and λ_t. Specifically, we have $\mu = 1 - 2\mathbb{E}[\lambda_t](1/2 - \mathbb{E}[r_t \mid M \geq m])$ given acceptance. In other words, large and accurate revisions speed up the decay of errors.

Remark 3.3. The optimal value of m depends on n and the distribution of r_t, and tends to $n/2$ as $n \to \infty$. For small n, the optimal $m \in \{n/2 + 1, \ldots, n\}$ can be different, as seen in Figure 2.

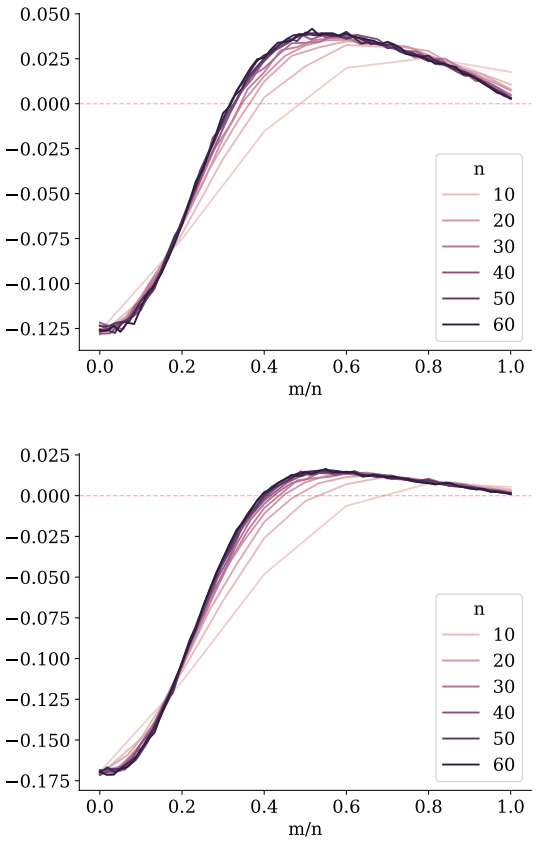


Figure 2: Improvement rate $-\log \mathbb{E}[E_{t+1}/E_t]$, by the acceptance threshold m/n. $P(r_t \leq 0.5) = 0.15$ on the left, $P(r_t \leq 0.5) = 0.05$ on the right. Values above the dashed line correspond to convergence, i.e. $E_t \xrightarrow{p} 0$.

Theorem 3.4. *If we have access to a noisy oracle that with probability* $1 - \varepsilon$ *observes the oracle, and with probability* ε *labels any edit as correct, (a), (b), (d) and (e) of Theorem 3.1 hold, and we set the acceptance threshold to* $m_{noisy} := m/(1 - \varepsilon)$*, the number of errors goes to zero, i.e.* $\lim_{t\to\infty} P(E_t = 0) = 1$.

Proof. A noisy oracle yields the following posterior

$$\begin{aligned} p(r \mid M = m) &\propto ((1-\varepsilon)(1-r) + \varepsilon)^m ((1-\varepsilon)r)^{n-m} \\ &\propto ((1-\varepsilon)(1-r) + \varepsilon)^m r^{n-m} \end{aligned} \tag{16}$$

Due to Lemma B.2 presented in the Appendix, this posterior stochastically dominates the following posterior

$$\begin{aligned} p(r \mid M = m) &\propto (1-r)^{(1-\varepsilon)m} r^{n-m} \\ &= (1-r)^{(1-\varepsilon)m} r^{(n+\varepsilon m)-(1-\varepsilon)m} \\ &= (1-r)^{m_{noisy}} r^{n_{noisy} - m_{noisy}} \end{aligned} \tag{17}$$

which in turn corresponds to the posterior of Theorem 3.1 obtained with $m_{noisy} := m(1-\varepsilon), n_{noisy} := n + \varepsilon m$. Inverting the mapping, noisy observations correspond to $m = m_{noisy}/(1-\varepsilon)$ and $n = n_{noisy} - \frac{\varepsilon}{1-\varepsilon} m_{noisy}$. The stochastical dominance means that the expectation of r with the posterior of Eq. (16) is smaller than with the posterior of Eq. (17), which means the expectation of Eq. (7) is going to be below 1 given the assumptions (a), (b), (d) and (e) of Theorem 3.1. □

From Theorem 3.4 we can see that we can easily account for a noisy oracle. This is , as expert knowledge perfect.

In Theorem 3.1, we do not specify the data format $\mathcal{S}$. On the other hand, assumptions (a) and (b) clearly depend on the data format. For that reason, we want to verify them for two common data formats: fixed-sized tables of arbitrary values, and sequences of text. For the former, the Hamming distance is a natural metric as corresponds to the number of incorrect values; for the latter, the addition-deletion edit distance (*Levenshtein distance* without substitutions) is a natural choice as it tells the number of edits needed to get to the correct data $S^\star$.

Theorem 3.5. *Tabular data, using the Hamming distance* $d(S, S') = \sum_{i=1}^{N} \mathbb{I}(s_i \neq s'_i)$*, fulfills the assumptions (a)-(e) of Theorem 3.1.*

Proof. Tabular data consists of a matrix $A \in \mathbb{R}^{N \times L}$. This is the data format of Assumption (a), for which the suitable distance is the number of non-equal entries, ranging from 0 when the tables are equal to $L \cdot N = K$ when each entry is different.

Without loss of generality, we can consider vectors $v \in \mathbb{Z}^K$ equipped with the metric $d(v, v') = \sum_{j=1}^{K} \mathbb{I}(v_j \neq v'_j)$. All possible unit modifications u are integer-scaled one-hot vectors, so $u = z \cdot [0, \ldots, 0, 1, 0, \ldots, 0]^T$. These changes are orthogonal; two non-orthogonal changes constitute one change. We thus can see that they are also additive in terms of the distances for Assumption (b) (see Appendix B for details).

The remaining assumptions do not concern the data structure and are fulfilled for tabular data. □

For Theorem 3.1 to hold for sequential data, such as textual data, we need further assumptions

(f) The true sequence $S^\star$ only contains distinct elements, i.e. $S^\star = [s_1^\star, \dots, s_N^\star]$, and $i \neq j \implies s_i^\star \neq s_j^\star \quad \forall (i,j) \in \{1, \dots, N\}$

(g) Each version S_t only contains distinct elements

(h) No version S_t contains any two elements $(s_i^\star, s_j^\star)$ of $S^\star$ in incorrect order

Theorem 3.6. *Sequences of text, using the addition-deletion edit distance, fulfil the assumptions (a)-(e) of Theorem 3.1 given further assumptions (f)-(h).*

Proof. The addition-deletion edit distance is defined for any two finite sequences of countable elements. It belongs to $\mathbb{N}$, i.e. $\mathcal{S}$ is equipped with a suitable metric d for Assumption (a). Furthermore, the edit distance can be obtained via the longest common subsequence (LCS)

$$d(S, S') = |S| + |S'| - 2|LCS(S, S')| \tag{18}$$

To verify Assumption (b) for any two edits (u_1, u_2), there are three possible cases: two additions, two deletions and an addition and a deletion. If the LCS is unique, we can inspect these cases via the effects of all potential pairs (u_1, u_2) on the LCS. This holds for sequences consisting of distinct elements that do not contain reordered elements of each other (proof in Appendix B), as guaranteed by Assumptions (f) and (h).

In the case of two erroneous additions, regardless of the order or position of these additions, E_t always increases by 2, given that the true data $S^\star$ and S_t only have distinct elements, guaranteed by Assumptions (f)–(g). This is because additions do not change the longest common substring if they are distinct from all elements of $S^\star$. Similarly, two correct additions are going to yield the same decrease of -2 to the distance, since they both increase the length of the LCS by 1, given that their order is not inverted. This is guaranteed by Assumption (h). Thus in the case any two additions u_1, u_2, $d(S_t + u_1 + u_2, S^\star) - d(S_t, S^\star) = d(S_t + u_1, S^\star) - d(S_t, S^\star) + d(S_t + u_2, S^\star) - d(S_t, S^\star)$.

Two deletions act independently, as it does not matter which one is done first; both either increase the edit distance by one if they are incorrect, or decrease it by one if they are correct, and the sum of their effects is the effect of their sums. Thus, $d(S_t + u_1 + u_2, S^\star) - d(S_t, S^\star) = d(S_t + u_1, S^\star) - d(S_t, S^\star) + d(S_t + u_2, S^\star) - d(S_t, S^\star)$ for any two deletions u_1 and u_2.

The third and final case involves the interactions of additions and deletions. The deletion and addition of two different elements do not interact, and thus, they are additive in terms of distance. The addition and subsequent deletion of the same element cancel each other and are not present in any deltas. Thus, additivity holds for all possible edits.

The remaining assumptions in Theorem 3.1 do not concern the data structure, and can also be fulfilled for text sequences. Thus, text sequences fulfil the assumptions of Theorem 3.1. □

Remark 3.7. $\mathcal{S}$ must be defined so that Assumptions (f)–(g) in Theorem 3.6 are reasonable. One way to do this is to organise a sequence into larger chunks. If a random sequence is split into chunks of K characters (or words), the probabilities of its composite elements can be made arbitrarily small. For example, chunks of K binary elements have 2^K possible categories, and their probabilities decrease exponentially as K is increased, guaranteeing that the elements in the sequence are distinct.

Theorem 3.8. *For tabular data, the number of errors E'_t with the tests converges faster than the number of errors without the tests E_t, i.e. $P(E_t = 0 \mid E_0 = E) < P(E'_t = 0 \mid E'_0 = E) \quad \forall t, E \in \mathbb{N}$.*

Proof. Without loss of generality, we consider K-dimensional vectors $v \in \mathbb{Z}^K$ of integers as $\mathcal{S}$. Let the data tests be a set of tuples $W = \{w_1, \dots w_J\}, 0 \leq J \leq K, \quad w_j \in \{1, \dots, K\} \times \mathbb{Z}$. The tests then *pass* when $\sum_{j=1}^{J} \mathbb{I}(v_{w_{j1}} \neq w_{j2}) = 0$.

For each sample, $J', 0 \leq J' \leq J$, the changes happen in the dimensions where there is a data test. Furthermore, $J^*, 0 \leq J^* \leq J'$ of these tests are both in dimensions where changes happen and not included in the sample. Moreover, $P(J' > 0) > 0$ for all timesteps t unless $E_t = 0$, and subsequently, since sampling is independent of the test $P(J^* > 0) > 0$ if $E_t \neq 0$. If only revisions where the tests pass are accepted, the likelihood is proportional to $B(J^* + m, n - m)$ instead of $B(m, n - m)$.

Let $p'(r_t \mid m)$ and $p(r_t \mid m)$ be the posteriors of the error rates corresponding to E'_t and E_t, respectively. Since the posterior $p'(r \mid m) \propto (1 - r)^{J^*} p(r \mid m)$, the posterior expectation $\mathbb{E}[p'(r_t)]$ is smaller than the posterior expectation $\mathbb{E}[p(r_t)]$. Moreover, $E_t - E_{t+1}$ and $E'_t - E'_{t+1}$ are binomially distributed with the rate parameter $r_t \lambda_T$, satisfying the monotone likelihood ratio property. This means the E'_{t+1} is stochastically smaller than E_{t+1} for all timesteps given any $E_t = E'_t > 0$, i.e. $P(E'_{t+1} \leq x) > P(E_{t+1} \leq x) \quad \forall x \in \mathbb{N}$, including $x = 0$ which denotes convergence. □

Theorem 3.8 illustrates the importance of adding tests as step 3.1 in the Proposal accept/reject step. By adding automated tests we will automatically reject poor proposals and speed up the convergence.

4 Experiments

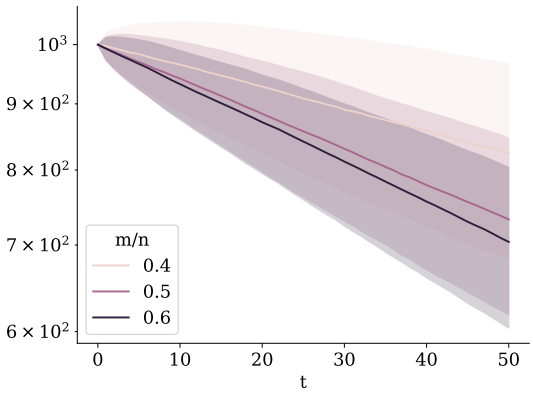
To assess the proposed iterative curation framework, we perform two simulation studies, which allow us to study the convergence behaviour of the method under varying assumptions. Moreover, we demonstrate the practical applicability of our approach via a use case involving the Swedish Parliamentary Corpus, a large dataset with both textual and tabular data [Yrjänäinen et al., 2024]. The code for the experiments is available in the Supplementary material.

4.1 Simulation Study

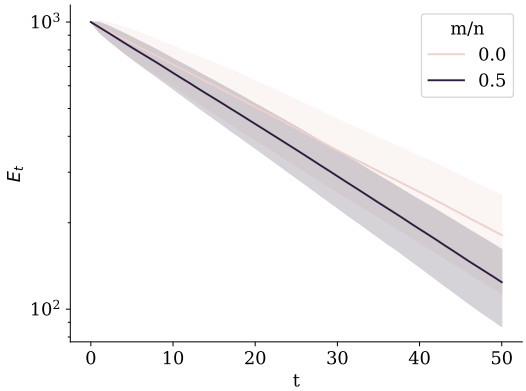
First, we simulate the number of errors E_t directly according to Theorem 3.1, and set the error rate $r_t \overset{i.i.d.}{\sim} Beta(\alpha, \beta)$, where $\mathbb{E}(r_t) = \alpha/(\alpha + \beta)$. The results are presented in Figure 3a. As expected from a BPRE process, the decay of the errors is exponential. Moreover, the threshold $m/n = 0.6$ outperforms $m/n = 0.5$ when $n = 10$, while $m/n = 0.5$ outperforms $m/n = 0.6$ when $n = 50$ (see Appendix A).

Furthermore, Figure 3b shows that even if the proposals are good enough to guarantee convergence, using the decision rule speeds it up. This is the case both for $n = 10$ and $n = 50$, where $m/n = 0.5$. We found this to be the case even with highly optimistic r_t, such as $Beta(1, 4)$, where 94% of the proposals would improve the data (see Appendix A). In addition, a simulation with a noisy oracle was done, where the errors converged similarly to a matching process with a true oracle, in line with Theorem 3.4 (see Appendix A).

As a second simulation, we generate a sequence $S^\star$ of 1,000,000 words sampled from the empirical distribution of the English Wikipedia. It includes lowercase words, spaces, and line breaks. To create an imperfect copy of $S^\star$ to be curated, we remove, add and change words with probability 1/300 each. This results in an initial data set with roughly 10,000 errors. For the deltas, we use the Myers difference algorithm on the word level. Figure 4 shows the evolution of E_t in the simulated text data over time. With the decision rule $m/n = 0.5$, E_t decreases exponentially, both in terms of the mean and the quantiles. This holds for different distributions of r_t (see Appendix A for details). Without the decision rule, when $r_t \sim Beta(1, 1), \mathbb{E}(r_t) = 1/2$, the mean stays static and variance increases with t. When $r_t \sim Beta(5, 3), \mathbb{E}(r_t) = 5/8$, E_t increases exponentially.



(a). $r_t \sim Beta(2, 1), \mathbb{E}(r_t) = 2/3$



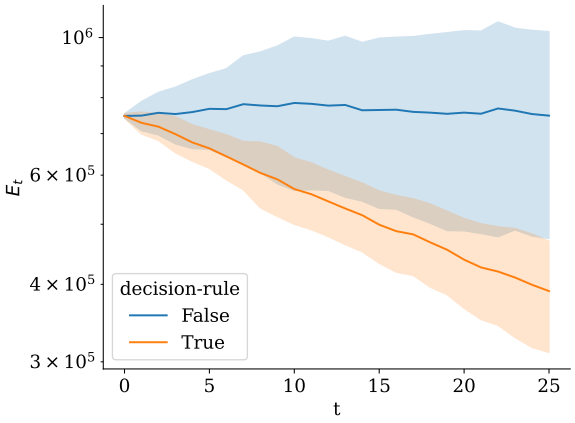
(b). $r_t \sim Beta(1, 2), \mathbb{E}(r_t) = 1/3$.

Figure 3: Number of errors E_t in the data as a function of t, $n = 10$. The exponential decay of the errors is seen on the logarithmic y axis.

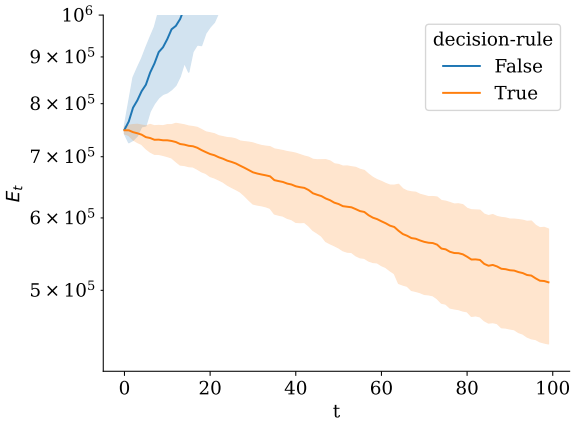
4.2 Case Study: The Swedish Parliamentary Records

As a practical use case, we present the curation of the Swedish parliamentary proceedings. In the corpus , the iterative approach has been used, and proposals are evaluated against automated testing and a random sample of 50 edits [see Yrjänäinen et al., 2024, for details]. The material is crucial for research in numerous scientific fields and Swedish parliamentary proceedings is utilised in training Swedish encoder models [Malmsten et al., 2020], audio-to-text models, and Nordic LLMs [Öhman et al., 2023]. The data comprises 17,938 parliamentary records between 1867 and 2024 with over 500 million words, in ParlaClarin XML format [Erjavec and Pancur, 2021], and metadata in a CSV format.

Mapping each speech in the text to the speaker in the metadata database ("speaker mapping") is a quality metric of particular importance to applied researchers

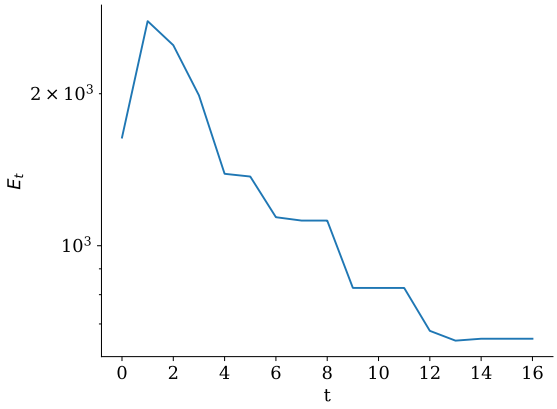


(a). $r_t \sim Beta(1,1)\,, \mathbb{E}(r_t) = 1/2$

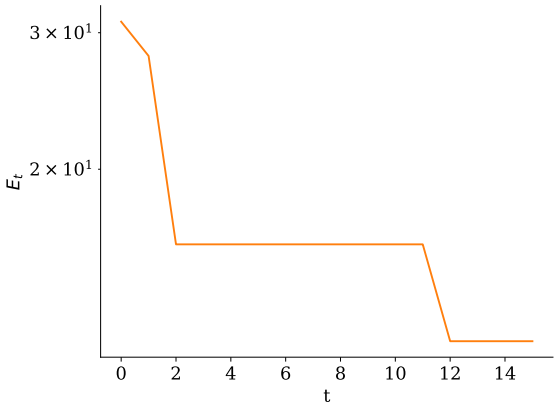


(b). $r_t \sim Beta(5,3)\,, \mathbb{E}(r_t) = 5/8$.

Figure 4: Errors in the simulated text dataset. Decision rule in orange (using $n = 50, m = 25$), no decision rule in green. 50 simulation runs, or until $E_t = 10^6$.



(a). Speaker mapping errors



(b). Paragraph classification errors

Figure 5: Errors over time in the Swedish Parliament data. The y axis is logarithmic to show the exponential decay of errors.

in the social sciences. An introduction is marked as 'unknown' if we can identify a speech but not map it to an identified MP. Figure 5a shows the number of unknown speakers over time, i.e. the reduction of missing annotations in the data. This accuracy has consistently improved over time (timestep 2 shows an increase due to the removal of false positives).

In addition to speaker introductions, the records contain transcriptions of speech, as well as non-speech elements such as transcriber notes, titles and margin notes. All paragraphs are classified into these classes ("paragraph classification"). It is not trivial to know which paragraphs are transcriptions of speech—the part of data that is of interest to most users—and thus they need to be programmatically detected due to the size of the corpus. The accuracy of this classification process is another data accuracy dimension of interest. Figure 5b shows the progression of the errors in this classification over the course of the corpus curation. The errors have either been reduced or remained the same for each new version.

5 Conclusion

We propose a scalable framework for iterative dataset data accuracy improvements. It supports iterative, high-quality updates with minimal manual overhead by evaluating revisions via automated tests and targeted sampling. Furthermore, we provide theoretical guarantees for the exponential decay of errors and convergence to an error-free dataset, the first such results to the best of our knowledge. Finally, we demonstrate the method's effectiveness via simulations and a real-world case study. Our approach is especially suited for large, evolving datasets where data accuracy is of direct importance, with immediate impact on robust social science research, machine learning benchmarks and high-quality LLM training corpora.

References

Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3561–3562, 2020.

Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. The effects of data quality on machine learning performance. *arXiv preprint arXiv:2207.14529*, 2022.

Miriam Hurtado Bodell, Måns Magnusson, and Sophie Mützel. From documents to data: A framework for total corpus quality. *Socius*, 8:23780231221135523, 2022.

Filipe Andrade Bernardi, Domingos Alves, Nathalia Crepaldi, Diego Bettiol Yamada, Vinícius Costa Lima, and Rui Rijo. Data quality in health research: integrative literature review. *Journal of medical Internet research*, 25:e41446, 2023.

Gregor Miller and Elmar Spiegel. Guidelines for research data integrity (grdi). *Scientific Data*, 12(1): 95, 2025.

Jingran Wang, Yi Liu, Peigong Li, Zhenxing Lin, Stavros Sindakis, and Sakshi Aggarwal. Overview of data quality: Examining the dimensions, antecedents, and impacts of data quality. *Journal of the Knowledge Economy*, 15(1):1159–1178, 2024.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

Jack E Olson. *Data quality: the accuracy dimension*. Elsevier, 2003.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Common Crawl. Common crawl dataset. `https://commoncrawl.org/`, 2024. Accessed on [2024-03-12].

Wikipedia contributors. Index of /enwiki/latest/, 2023. URL `https://dumps.wikimedia.org/enwiki/latest/`. [Online; accessed 19-Dec-2023].

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021.

Matthew Gentzkow, Jesse M Shapiro, and Matt Taddy. Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts. In *URL: https://data.stanford.edu/congress_text*, 2018.

Tomaz Erjavec, Maciej Ogrodniczuk, Petya Osenova, Nikola Ljubevic, Kiril Simov, Andrej Pancur, Michal Rudolf, Matyas Kopp, Starkadhur Barkarson, Steinthorr Steingrimsson, et al. The parlamint corpora of parliamentary proceedings. *Language resources and evaluation*, 57(1):415–448, 2023.

Andreas Östling, Holli Sargeant, Huiyuan Xie, Ludwig Bull, Alexander Terenin, Leif Jonsson, Måns Magnusson, and Felix Steffek. The cambridge law corpus: A dataset for legal ai research. *Advances in Neural Information Processing Systems*, 36:41355–41385, 2023.

Umar Butler. Open australian legal corpus, 2025. URL `https://huggingface.co/datasets/isaacus/open-australian-legal-corpus`.

Mark Davies. The corpus of contemporary american english (coca), 2008. URL `https://www.english-corpora.org/coca/`. Accessed: 2025-04-14.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.

Holger Voormann and Ulrike Gut. Agile corpus creation. 2008.

Gitansh Khirbat. Ocr post-processing text correction using simulated annealing (opteca). In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 119–123, 2017.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. Spelling error correction with soft-masked bert. *arXiv preprint arXiv:2005.07421*, 2020.

Del Fante, Di Nunzio, et al. A quantitative/qualitative approach to {OCR} error detection and correction in old newspapers for corpus-assisted discourse studies. In *CEUR WORKSHOP PROCEEDINGS*, volume 2816, pages 53–63, 2021.

Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Relabeling imagenet: from single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2340–2350, 2021.

MA Dakka, TV Nguyen, JMM Hall, SM Diakiw, M VerMilyea, R Linke, M Perugini, and D Perugini. Automated detection of poor-quality data: case studies in healthcare. *Scientific Reports*, 11(1):18005, 2021.

Kimberly A Barchard and Larry A Pace. Preventing human error: The impact of data entry methods on data accuracy and statistical results. *Computers in Human Behavior*, 27(5):1834–1839, 2011.

Chao Gao, Yu Lu, and Dengyong Zhou. Exact exponent in optimal rates for crowdsourcing. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 603–611, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/gaoa16.html`.

Florian Brühlmann, Serge Petralito, Lena F Aeschbach, and Klaus Opwis. The quality of data collected online: An investigation of careless responding in a crowdsourced sample. *Methods in Psychology*, 2: 100022, 2020.

Tom Preston-Werner. Semantic versioning 2.0. 0. `https://semver.org/`, 2013.

Tomas Erjavec and Andrej Pancur. The parla-clarin recommendations for encoding corpora of parliamentary proceedings. *Journal of the Text Encoding Initiative*, (14), 2021.

Eugene W Myers. An o (nd) difference algorithm and its variations. *Algorithmica*, 1(1-4):251–266, 1986.

Linus Torvalds, Junio Hamano, et al. Git — git-scm.com. `https://git-scm.com/`, 2005. [Accessed 20-Dec-2022].

Walter L Smith and William E Wilkinson. On branching processes in random environments. *The Annals of Mathematical Statistics*, pages 814–827, 1969.

Yves Guivarc'h and Quansheng Liu. Propriétés asymptotiques des processus de branchement en environnement aléatoire. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 332(4):339–344, 2001.

Väinö Yrjänäinen, Fredrik Mohammadi Norén, Robert Borges, Johan Jarlbrink, Lotta Åberg Brorsson, Anders P Olsson, Pelle Snickars, and Måns Magnusson. The swedish parliament corpus 1867–2022. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pages 16100–16112, 2024.

Martin Malmsten, Love Börjeson, and Chris Haffenden. Playing with words at the national library of sweden–making a swedish bert. *arXiv preprint arXiv:2007.01658*, 2020.

Joey Öhman, Severine Verlinden, Ariel Ekgren, Amaru Cuba Gyllensten, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, and Magnus Sahlgren. The nordic pile: A 1.2 tb nordic dataset for language modeling. *arXiv preprint arXiv:2303.17183*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [No]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [No]
   (c) New assets either in the supplemental material or as a URL, if applicable. [No]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Iterative Data Curation with Theoretical Guarantees: Supplementary Materials

A Additional Experiments and Figures

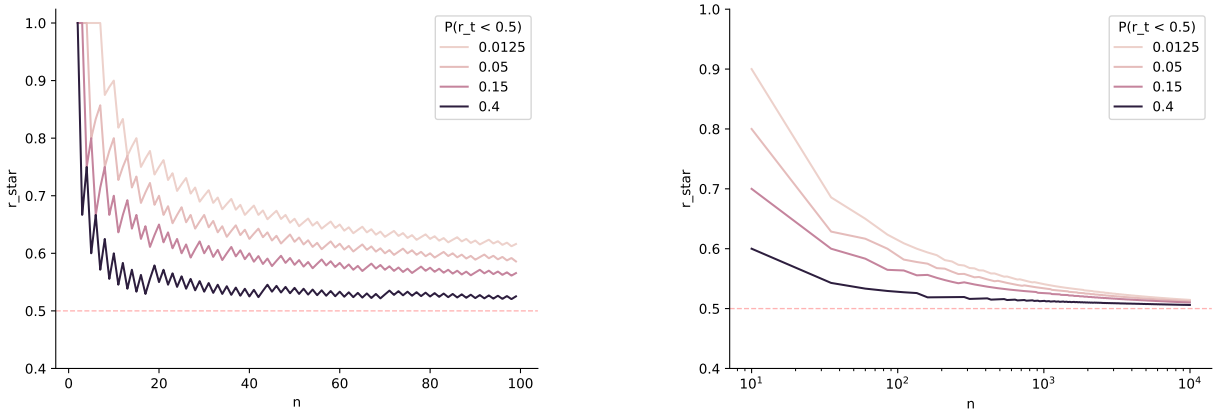
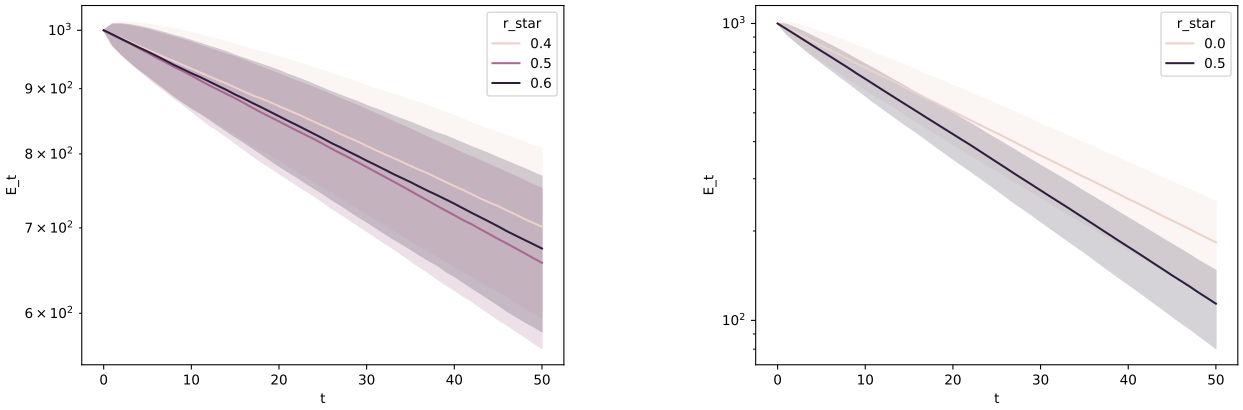


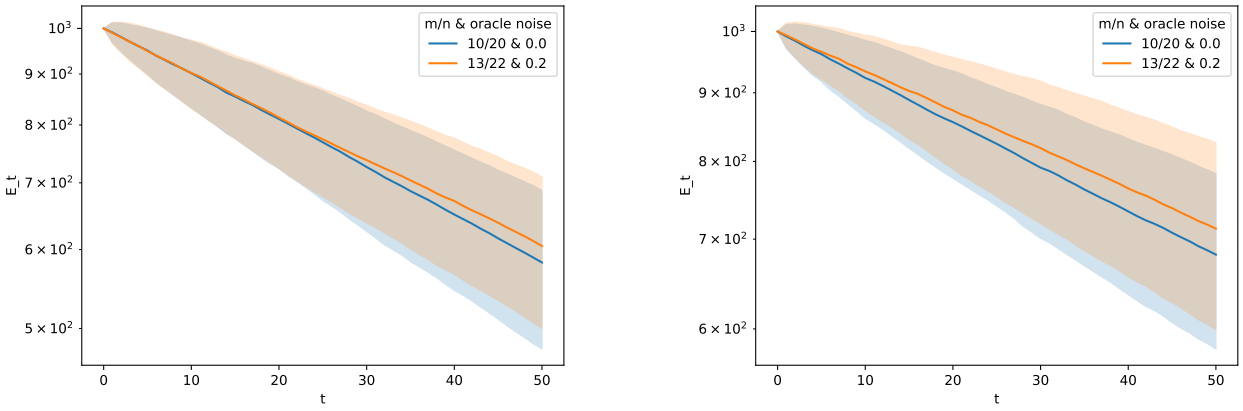
Figure 6: Convergence thresholds for different priors for r. The lines denote the smallest m that guarantees convergence as a ratio of the sample size m/n. On the left, values $n \in \{2, 3, \ldots, 99, 100\}$; on the right logarithmically spaced $n \in \{10^1, \ldots, 10^4\}$. The red dashed line denotes the asymptote $m = n/2$.



(a). $r \sim Beta(2, 1)$ (b). $r \sim Beta(1, 2)$.

Figure 7: Number of errors in the data by iteration, $n = 50$. $r_t \sim Beta(1, 2)$. The y-axis is logarithmic to show the exponential decay of the errors.

In Figure 8, results from a simulation with and without a noisy oracle are presented. By theorem 3.4, n and m for the noisy oracle are set so that they correspond to the true oracle. Specifically, $n = 20 = n_{noisy} - \frac{\varepsilon}{1-\varepsilon} m_{noisy}$ and $m = 10 = m_{noisy}/(1 - \varepsilon)$. Since non-integer values are obtained but cannot be used , n is rounded down, while m is rounded up.



(a). $r \sim Beta(1.5, 0.8)$ (b). $r \sim Beta(1, 2)$.

Figure 8: Number of errors in the data by iteration, $n = 50$. Oracle and noisy oracle with noise rate 0.2. n and m are set to correspond to the values of the theorem. The y-axis is logarithmic to show the exponential decay of the errors.

The figures show that E_t under a noisy oracle follows a rather similar distribution as E_t under the true oracle.

B Proofs and detailed derivations

B.1 Derivations for Theorem 3.1

The expectation of the number of descendants for one branch is

$$\mathbb{E}[\zeta \mid M = m] = \int_0^1 \int_0^1 \sum_{k=0}^{2} k \cdot p_{k,t} dr_t d\lambda_t \tag{19}$$

$$= \int_0^1 \int_0^1 (1 - \lambda + 2(1 - r_t)\lambda)\, p(r_t \mid M = m) dr_t d\lambda_t \tag{20}$$

$$= \int_0^1 \int_0^1 (1 - \lambda + 2\lambda - 2r_t\lambda)\, p(r_t \mid M = m) dr_t d\lambda_t \tag{21}$$

$$= \int_0^1 \int_0^1 (1 + \lambda - 2r_t\lambda)\, p(r_t \mid M = m) dr_t d\lambda_t \tag{22}$$

$$= 1 + \int_0^1 \lambda - 2\lambda \int_0^1 r_t p(r_t \mid M = m) dr_t d\lambda_t \tag{23}$$

$$= 1 + \mathbb{E}\Big[\lambda - 2\lambda \mathbb{E}[r_t \mid M = m]\Big] \tag{24}$$

$$= 1 + 2\mathbb{E}[\lambda]\,(1/2 - \mathbb{E}[r_t \mid M = m]) \tag{25}$$

and subsequently

$$1/2 - \mathbb{E}[r \mid M = m] \tag{26}$$

$$= \mathbb{E}[1/2 - r \mid M = m] \tag{27}$$

$$= \int_0^1 (1/2 - r) p(r \mid m) dr \tag{28}$$

$$\propto \int_0^{0.5} (1/2 - r) p(m \mid r) p(r) dr + \int_{0.5}^1 (1/2 - r) p(m \mid r) p(r) dr \tag{29}$$

$$\geq \int_0^{0.5} (1/2 - r) p(m \mid r) p(r) dr - \max_r \binom{n}{m} \left[(r - 1/2)(1 - r)^m r^{n-m}\right] \int_{0.5}^1 p(r) dr \tag{30}$$

$$\propto \int_0^{0.5} (1/2 - r)p(m \mid r)p(r)dr - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{31}$$

$$\geq a \int_0^{0.5} (1/2 - r)p(m \mid r)dr - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5)$$

$$\geq a \int_0^{1} (1/2 - r)p(m \mid r)dr - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{32}$$

$$= a/2 \int_0^1 p(m \mid r)dr - a \int_0^1 rp(m \mid r)dr \tag{33}$$

$$- \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{34}$$

$$= a/2 \int_0^1 \binom{n}{m}(1-r)^m r^{n-m}dr - a \int_0^1 \binom{n}{m}(1-r)^m r^{n-m+1}dr \tag{35}$$

$$- \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5)$$

$$= a\binom{n}{m} \left(1/2B(n-m+1, m+1) - B(n-m+2, m+1)\right) \tag{36}$$

$$- \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5)$$

$$= a\binom{n}{m} \left(1/2\frac{2m-n}{n+2}B(n-m+1, m+1)\right) - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{37}$$

$$= a\left(1/2\frac{2m-n}{n+2}\frac{1}{n+1}\right) - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{38}$$

$$= a\left(\frac{2m-n}{2(n+1)(n+2)}\right) - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] P(r \geq 0.5) \tag{39}$$

$$\geq a\left(\frac{2m-n}{2(n+1)(n+2)}\right) - \max_r \binom{n}{m} \left[(r-1/2)(1-r)^m r^{n-m}\right] (1 - a/2) \tag{40}$$

$$\propto a - \Bigg[\underbrace{\left(\frac{2(n+1)(n+2)}{2m-n}\right)\binom{n}{m} \cdot \max_r \left[(r-1/2)(1-r)^m r^{n-m}\right]}_{C(n,m)}\Bigg](1 - a/2) \tag{41}$$

$$= a - C(n,m)(1 - a/2) \tag{42}$$

$$\propto a(2 + C(n,m)) - C(n,m) \tag{43}$$

$$\propto a - C(n,m)/(2 + C(n,m)) \tag{44}$$

B.2 Derivations and properties of $C(n,m)$ for Theorem 3.1

To maximize $\binom{n}{m}(r-1/2)(1-r)^m r^{n-m}$ in the interval $(1/2, 1)$, we can maximize its logarithm since it's a product of positive factors

$$\log\left[\binom{n}{m}(r-1/2)(1-r)^m r^{n-m}\right] = \log\binom{n}{m} + \log(r-1/2) + m\log(1-r) + (n-m)\log r \tag{45}$$

This expression has the derivative

$$\frac{d}{dr}\log[\binom{n}{m}(r-1/2)(1-r)^m r^{n-m}] = \frac{1}{r-1/2} - \frac{m}{1-r} + \frac{n-m}{r} \tag{46}$$

Its roots are the same when scaled with the expression $(r-1/2)(1-r)r$, which is positive in the range $(1/2, 1)$

$$\frac{d}{dr}\log[(r-1/2)(1-r)^m r^{n-m}] = 0$$

$$\iff (r-1/2)(1-r)r\left(\frac{1}{r-1/2} - \frac{m}{1-r} + \frac{n-m}{r}\right) = 0$$

$$\iff 2mr - m + 2nr^2 - 3nr + n + 2r^2 - 2r = 0 \tag{47}$$

This polynomial has the roots

$$r = \frac{3n - 2m + 2 \pm \sqrt{(2m-n)^2 + 4(n+1)}}{4(n+1)}$$

If $n, m \geq 1$ and $2n \geq n$, the larger root is larger than $1/2$ since

$$\frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4(n+1)} > \frac{3n - 2m + 2 + \sqrt{(n-2m)^2}}{4(n+1)} \tag{48}$$

$$= \frac{3n - 2m + 2 - (n-2m)}{4(n+1)} \quad | \; 2m \geq n \tag{49}$$

$$= \frac{2n+2}{4(n+1)} = \frac{2(n+1)}{4(n+1)} = \frac{1}{2} \tag{50}$$

and smaller than 1 since

$$\frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4(n+1)} < \frac{3n - 2m + 2 + \sqrt{(n-2m)^2} + 2\sqrt{n+1}}{4(n+1)} \tag{51}$$

$$= \frac{3n - 2m + 2 - (n-2m) + 2\sqrt{n+1}}{4(n+1)} \tag{52}$$

$$= \frac{2n + 2 + 2\sqrt{n+1}}{4(n+1)} \tag{53}$$

$$= \frac{1}{2} + \frac{1}{2\sqrt{n+1}} \leq 1/2 + 1/2 \tag{54}$$

since $\|x\|_1 \geq \|x\|_2$ and $2m \geq n$.

Similar logic can be used to show that the smaller root is always smaller than $1/2$. Thus, the derivative has only one root in the range $(1/2, 1)$, namely

$$r = \frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4} \tag{55}$$

As the expression $(r - 1/2)(1-r)^m r^{n-m}$ is zero at the edges of the boundaries, and positive within the interval, its maximum is found at this root of the derivative. The maximum is

$$\begin{aligned} \max_r [(r-1/2)(1-r)^m r^{n-m}] = &\left(\frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4} - 1/2\right) \\ &\cdot \left(1 - \frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4}\right)^m \\ &\cdot \left(\frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4}\right)^{n-m} \end{aligned} \tag{56}$$

$$\begin{aligned} = &\left(\frac{n - 4m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4}\right) \\ &\cdot \left(\frac{n - 2m + 2 - \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4}\right)^m \\ &\cdot \left(\frac{3n - 2m + 2 + \sqrt{(n-2m)^2 + 4(n+1)}}{4n+4}\right)^{n-m} \end{aligned} \tag{57}$$

Further properties can be shown for the maximum $C(n,m)$. For instance, setting $m = n - d$ where $d \in \mathbb{N}$ and kept constant, $C(n,m)$ can be made arbitrarily small

$$\begin{aligned}
C(n, n-d) &= \frac{2(n+1)(n+2)}{2m-n}\binom{n}{n-d}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&< 2(n+1)(n+2)\binom{n}{n-d}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&< 2(n+2)^2\binom{n}{n-d}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&< 2(n+2)^{2+d}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&= 2(n+2)^{2+d}\max_r[(r-1/2)(1-r)^{n-d} r^d] \\
&\leq 2(n+2)^{2+d}\max_r[(r-1/2)(1-r)^{n-d}] && \mid\ r \leq 1 \\
&\leq 2(n+2)^{2+d} 1/2(1-1/2)^{n-d} && \mid\ 1/2 \leq r \leq 1 \\
&\leq 2(n+2)^{2+d} 1/2^{n-d+1} = O(2n^{2+d}2^{-n}) \to 0
\end{aligned} \tag{58}$$

i.e. $C(n, n-d)$ decays exponentially with growing n. Note that by the same steps, the same holds for even a ratio $m = qn$, where $q > 0.5$, though with the exponent changing depending on the ratio

$$\begin{aligned}
C(n, qn) &= \frac{2(n+1)(n+2)}{2m-n}\binom{n}{qn}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&< 2(n+2)^2\binom{n}{qn}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&\propto (n+2)^2\left(\frac{1}{q^q(1-q)^{1-q}}\right)^n \max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&< (n+2)^2(q')^{-n}\max_r[(r-1/2)(1-r)^m r^{n-m}] \\
&= (n+2)^2(q')^{-n}\max_r[(r-1/2)(1-r)^{qn} r^{(1-q)n}] \\
&\leq (n+2)^2(q')^{-n} 1/2^{qn} r_{max}^{(1-q)n} \\
&\leq (n+2)^2(q')^{-n} 1/2^{qn} {q'}^{(1-q)n} \\
&= (n+2)^2(q')^{-qn} 1/2^{qn} \\
&= O\left(n^2(2q')^{-n}\right) \to 0
\end{aligned} \tag{59}$$

since $\forall q \quad \exists q' < 1/2$ for which $\frac{1}{q^q(1-q)^{1-q}} < 1/q'$.

B.3 Derivations for Theorem 3.5

The additivity assumption in Theorem 3.5, which applies for K-dimensional vectors, can be directly shown

$$\begin{aligned}
& d(S_t + u_1 + u_2, S^\star) - d(S_t, S^\star) \\
&= -\sum_{j=1}^{K} \mathbb{I}(S_{t,j} + u_{1,j} + u_{2,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star) \\
&= -\sum_{j=1}^{K} \mathbb{I}(u_{1,j} \neq 0 \wedge u_{2,j} = 0)\left(\mathbb{I}(S_{t,j} + u_{1,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star)\right) \\
&\quad -\sum_{j=1}^{K} \mathbb{I}(u_{1,j} = 0 \wedge u_{2,j} \neq 0)\left(\mathbb{I}(S_{t,j} + u_{2,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star)\right) \\
&\quad -\sum_{j=1}^{K} \underbrace{\mathbb{I}(u_{1,j} \neq 0 \wedge u_{2,j} \neq 0)}_{\text{always } 0}\left(\mathbb{I}(S_{t,j} u_{1,j} + u_{2,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star)\right)
\end{aligned}$$

$$
\begin{aligned}
&= -\sum_{j=1}^{K} \mathbb{I}(u_{1,j} \neq 0 \wedge u_{2,j} = 0) \left(\mathbb{I}(S_{t,j} + u_{1,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star)\right) \\
&\quad -\sum_{j=1}^{K} \mathbb{I}(u_{1,j} = 0 \wedge u_{2,j} \neq 0) \left(\mathbb{I}(S_{t,j} + u_{2,j} = S^\star) - \mathbb{I}(S_{t,j} = S^\star)\right) \\
&= d(S_t + u_1, S^\star) - d(S_t, S^\star) + d(S_t + u_2, S^\star) - d(S_t, S^\star)
\end{aligned}
$$

which holds for any data state S_t, true data $S^\star$, and pair of edits (u_1, u_2).

B.4 Properties of the longest common subsequence

We are unaware of this proof being found in the literature. However, as it is rather straightforward it is likely that we are not the first to prove it.

Theorem B.1. *If S and S' are two sequences without duplicate elements, and for each two elements (x, y) that are present in both S and S', the relative order of such elements is the same in both sequences, then their longest common subsequence is unique.*

Proof. Let C be the set of elements common to both S and S'. Let $|C| = k$. Any common subsequence of S and S' can only be formed using elements from C. Therefore, the length of the longest common subsequence (LCS) is at most k.

Consider the subsequence formed by taking all the elements of C in the order they appear in S. Let this subsequence be L_S. Since for each pair of elements $(x, y) \in C \times C, x \neq y$, their relative order is the same in S' as it is in S, L_S is also a subsequence of S'. Thus, L_S is a common subsequence of length k. Therefore, the length of the LCS is at least k.

Combining these two points, the length of the LCS is exactly k, the number of common elements.

Now, we need to show that this LCS is unique. Suppose there exists another common subsequence L' of length k that is different from L_S. Since L' has length k, it must contain all the elements of C. However, because the relative order of the elements of C is the same in both S and S', any subsequence of length k formed by elements of C must have the elements in the same relative order. Therefore, L' must be the same as L_S.

Thus, the longest common subsequence is unique. □

B.5 Stochastic dominance of the alternative posterior for the noisy oracle

Lemma B.2. *The posterior proportional to $f(r \mid M = m) \propto ((1-\varepsilon)(1-r) + \varepsilon)^m r^{n-m} p(r)$ is stochastically dominated by the posterior proportional to $g(r) = (1-r)^{(1-\varepsilon)m} r^{n-m} p(r)$.*

Proof. If a posterior proprtional to $f(x)$ has the monotone likelihood property wrt. another posterior proportional to $g(x)$

$$
\frac{d}{dx} \frac{f(x)}{g(x)} \geq 0 \quad \forall x \tag{60}
$$

the first posterior also stochastically dominates the second.

We obtain the ratio

$$
f(x)/g(x) = \frac{((1-\varepsilon)r + \varepsilon)^m (1-r)^{n-m} p(r)}{r^{(1-\varepsilon)m}(1-r)^{n-m} p(r)} = \frac{((1-\varepsilon)r + \varepsilon)^m}{r^{(1-\varepsilon)m}} \tag{61}
$$

We can investigate its derivative by taking the logarithm

$$
\ln \frac{((1-\varepsilon)r + \varepsilon)^m}{r^{(1-\varepsilon)m}} = m \ln((1-\varepsilon)r + \varepsilon) - (1-\varepsilon) m \ln r \tag{62}
$$

and further removing the multiplier $m \geq 1$

$$
\frac{d}{dr} \ln((1-\varepsilon)r + \varepsilon) - (1-\varepsilon) \ln r = \frac{1-\varepsilon}{(1-\varepsilon)r + \varepsilon} - \frac{1-\varepsilon}{r} \tag{63}
$$

$$\propto \frac{1}{(1-\varepsilon)r+\varepsilon} - \frac{1}{r} \tag{64}$$

$$= \frac{1}{(1-\varepsilon)r+\varepsilon} - \frac{1}{(1-\varepsilon)r+\varepsilon r} \tag{65}$$

Since $\varepsilon r \leq \varepsilon$ for all $r \in [0,1]$, the derivative is nonnegative and the property holds. □

C Additional data types

C.1 Tabular data with an unknown number of indexed rows

$\mathcal{S}$ are sets consisting of tuples $(e,k,v) \in \mathbb{Z}^L \times \{1,\dots,K\} \times \mathbb{R}$. Each of the elements maps the index of an element, and the index of the value to the value. For example, an element indexed $e = 123$ at the column $k = 7$ has the value $v = 0.324$. This allows for an indexed tabular data format with an arbitrary number of elements with missing values. Furthermore, $\mathcal{S}$ is defined to not have duplicates in terms of $(e,k,\cdot)$.

A suitable distance metric is the size of the symmetric set difference

$$d(S,S') = \sum_{s\in S'} \mathbb{I}(s \notin S) + \sum_{s\in S} \mathbb{I}(s \notin S') \tag{66}$$

The symmetric set difference can also be used as $\Delta(S,S')$ from which to sample edits. Intuetively, each missing value is punished with 1, and each incorrect value with 1; mapping a valid index $(e,k,\cdot)$ to an incorrect value is thus punished with 2. This relies on the elements being unique.

Single additions have a straightforward effect on the distance

$$\begin{aligned} d(S+u,S') - d(S,S') &= \sum_{s\in S^\star} \mathbb{I}(s \notin S \cup \{u\}) \\ &\quad + \sum_{s\in S\cup\{u\}} \mathbb{I}(s \notin S^\star) \\ &\quad - \sum_{s\in S^\star} \mathbb{I}(s \notin S) - \sum_{s\in S} \mathbb{I}(s \notin S^\star) \\ &= -\mathbb{I}(u \in S^\star) + \mathbb{I}(u \notin S^\star) \\ &= 1 - 2\mathbb{I}(u \in S') \end{aligned} \tag{67}$$

Addition to S corresponds to deletion from S', which yields a similar formula for the deletion of u

$$d(S+u,S') - d(S,S') = 1 - 2\mathbb{I}(u \notin S') \tag{68}$$

Furthermore, the metric fulfils the additivity assumption

$$\begin{aligned} &d(S+u_1+u_2,S^\star) - d(S,S^\star) \\ &= \sum_{s\in S^\star} \mathbb{I}(s \notin S \cup \{u_1,u_2\}) + \sum_{s\in S\cup\{u_1,u_2\}} \mathbb{I}(s \notin S^\star) \\ &- \sum_{s\in S^\star} \mathbb{I}(s \notin S) - \sum_{s\in S} \mathbb{I}(s \notin S^\star) \\ &= \sum_{s\in S^\star} -\mathbb{I}(s \in S \cup \{u_1,u_2\}) + \sum_{s\in\{u_1,u_2\}} \mathbb{I}(s \notin S^\star) \\ &= \sum_{s\in\{u_1,u_2\}} -\mathbb{I}(s \in S^\star) + \sum_{s\in\{u_1,u_2\}} \mathbb{I}(s \notin S^\star) \\ &= 1 - 2\mathbb{I}(u_1 \in S^\star) + 1 - 2\mathbb{I}(u_2 \in S^\star) \\ &= d(S+u_1,S^\star) - d(S,S^\star) + d(S+u_2,S^\star) - d(S,S^\star) \end{aligned} \tag{69}$$

D Diff sampling of text

For the sampling of edits in text data, `git diff --stat` is run to obtain the number of changes by file. Then, a stratified sample is taken of these files to obtain the number of edits to sample per each file, which all sum up to n, resulting n_f for all $f \in F$, which is the set of files sampled. Then, `git diff` is run on the files where is at least 1 (which is n different files at maximum). The result of each of these commands is piped to a .diff file, and for each file f, n_f diffs are uniformly sampled. The procedure can be fully automated and is provided in the Supplementary Material as a Python package.

An example of an individual sampled edit is provided below:

```
data/1907/prot-1907--ak--023.xml

Diff starting from line [2155](https://github.com/swerik-project/riksdagen-
    ↪ records/tree/join-seg-II/data/1907/prot-1907--ak--023.xml#L2155)

@@ -2239,10 +2155,7 @@
          </note>
          <note xml:id="i-GpA3qd9kHzpzc5cXwJybo1">
            Angaende en ny forbindelseled mellan Baggensfjarden och en inre
-           Stockholmsskargarden.
-         </note>
-         <note xml:id="i-P9vjoFjDP3bFydxB7oD7M8">
-           (Forts )
+           Stockholmsskargarden. (Forts )
          </note>
          <note xml:id="i-PyYHcEah3YgC4sBc2fUQiQ" type="date">
            26 Onsdagen den 13 Mars, e. m.

- [x] Correct
- [ ] Incorrect
```

This particular edit was deemed correct, as it merged an accidentally split paragraph.