

Corrections Need Memory

Version history can show what changed. Correction memory should preserve why it changed, what it affected and where the old state may remain.

DRAFT PREVIEW DENNIS HEDEGREEN ANALYSE DRAFT-LOCAL

[HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/CORRECTIONS-NEED-MEMORY/](https://hedegreenresearch.com/articles/corrections-need-memory/)

Version history preserves states. Correction notes explain changes. Correction memory connects the two.

An article moves from version 1.1 to version 1.2.

The reader can open both.

That is already better than silently replacing the old page.

But one question still sits between them:

Why did v1.1 become v1.2?

A version history can show before and after.

A correction note can explain the mistake.

Correction memory should connect those two things.

A version is a state. A correction is a transition.

Suppose version 1.1 says:

The project spent 1,000 EUR.

Version 1.2 says:

The project spent 1,000 USD.

A diff can show that EUR became USD.

But the diff does not explain why.

Was the first version wrong?

Did the source itself change?

Did the newsroom misread it?

Did new evidence arrive?

Was this a correction or an update?

Those are facts about the editorial transition.

The useful record is not only:

v1.1 → v1.2

It is:

v1.1 → reason → v1.2

Knowing what changed is not the same as knowing why it changed.

Journalism already preserves parts of this

This is not an argument that newsrooms lack correction systems.

Associated Press requires factual errors to be explicitly labeled as corrections rather than disguised with softer terms. Its guidance also treats correction as a distribution problem: consumers, subscriber editors, graphics, captions, video and audio may all need corrective handling. For live online text stories, AP says the earlier version is overwritten while separate corrective stories may also be sent when warranted.

The Guardian says significant errors should be corrected as soon as possible on the relevant web page and/or in print, with significant corrections also collected in its Corrections and Clarifications column.

ProPublica maintains a public corrections page whose notices often preserve a very useful form:

the story originally said X; the correct information is Y.

It also uses **Update** separately when later information changes the state of an article without necessarily implying that the earlier reporting was wrong.

Reuters' journalism handbook provides a distribution-heavy model: corrections, refiles, repeats and withdrawals are distinct actions, errors should not be buried in later updates, and advisory lines explain what is being corrected.

These are strong practices.

They also expose the missing connection.

A correction note may explain the error.

A version history may preserve the page.

A distribution workflow may push the fix downstream.

Correction memory joins those pieces.

A correction note explains the change

A good correction note might say:

This article originally stated that the project spent 1,000 EUR. The correct currency is USD.

That is useful.

But the note can still be disconnected from the underlying article state.

Which amount changed?

Which claim depended on it?

Which source established the correction?

Which version contained the error?

Did the headline also contain it?

Did a PDF export still carry the old state?

Correction memory makes that explanation addressable.

It can connect the change to:

- the prior version;
- the new version;
- the affected claim, amount, source, percentage, question, headline or section;
- the reason;
- the evidence that triggered it;
- the surfaces that may still carry the old state.

The reader does not need to see every field.

The newsroom may still need to remember them.

A correction note explains the change. Correction memory makes the explanation addressable.

An update is not a correction

One easy way to destroy this history is to call every change an update.

A **correction** means previously published information was wrong.

An **update** adds later information to a story that may have been accurate when first published.

A **clarification** makes wording more precise when the original presentation could be misunderstood.

A **retraction** withdraws a claim, section or article.

Those are different editorial events.

Suppose an agency revises a dataset after publication.

If the newsroom accurately reported the original release, changing the article later may be an update reflecting a new source state, not a correction of newsroom error.

That distinction should survive.

An update is not a correction.

A correction should point to what changed

Structured articles make this easier.

If an article already contains identifiable amounts, sources, claims, percentages and open questions, then a correction can point directly to the affected thing.

For example:

Correction — v1.2

Affected: Amount M04 Before: 1,000 EUR After: 1,000 USD Reason: the original article used the wrong currency.

That is more precise than:

This article has been corrected.

But not every correction maps neatly to one object.

Sometimes a paragraph's framing changes.

Sometimes two individually accurate claims create a misleading overall impression.

Sometimes the headline overstates the evidence.

So the standard should be:

A correction should point to the thing that changed when that thing can be identified.

The fallback can be a headline, paragraph, section or the article as a whole.

Preserve the prior state. Expose it proportionately.

Transparency does not require every old version to remain publicly exposed forever.

An earlier version may contain:

- defamatory material;
- private information;
- dangerous instructions;
- a false accusation;
- an accidental disclosure;
- material that must be removed for legal or safety reasons.

The editorial record can preserve enough history to reconstruct the change without amplifying the harmful state to every reader.

That creates three layers:

Current state — what the reader should understand now.

Correction memory — why the state changed.

Historical state — what was previously published.

They do not need identical access rules.

Preserve the prior state. Expose it proportionately.

A correction is not only a page edit

Corrections also travel.

If an error appeared only in the body text, correcting the page may be enough.

But the same error may also appear in:

- a headline;
- a push notification;
- a PDF;
- a graphic;
- a subscriber feed;
- a social card;
- a caption.

AP's correction rules and Reuters' wire-oriented handbook both make this especially visible: correction handling includes distribution, not only rewriting a page.

A publisher cannot always retrieve every old copy.

A screenshot can circulate elsewhere.

A push may already have been read.

A cached page may sit outside newsroom control.

So the rule should be modest:

Correct the surfaces you still control, and record where the old state may remain.

Correction memory does not promise to erase an error from the internet.

It prevents the newsroom from forgetting where its own earlier state travelled.

Correction memory can itself change

The explanation for a correction can later turn out to be incomplete.

A newsroom may initially believe it misread Source A.

Later it may discover that Source A itself was corrected.

The correction record therefore cannot be treated as immutable truth.

It may itself need revision.

That sounds recursive, but the principle is straightforward:

the newsroom should preserve the history of its explanation as well as the history of its article.

The reader does not need every edit

Most edits should remain invisible.

Spelling fixes.

Punctuation.

Formatting.

Minor wording changes that do not alter meaning.

The threshold is materiality.

A change becomes journalistically meaningful when it alters what a reasonable reader would understand about the facts, evidence, attribution, uncertainty, responsibility or status of a claim.

That threshold will always involve editorial judgment.

The system can preserve more internally than it exposes publicly.

Store more than you show.

Implementation status at Hedegreen Research

Hedegreen Research already preserves article version numbers and allows readers to inspect earlier versions.

The correction-memory layer proposed here — including explicit change type, reason, affected object, trigger evidence and propagation state — should be treated as a proposed extension unless those fields are already deployed in the current production system at publication time.

The article's journalistic standard does not depend on one specific UI.

Four principles

A minimal standard could be:

1. **Classify material changes.** Distinguish corrections, updates, clarifications and retractions.
2. **Preserve the reason.** Record why the published state changed.
3. **Link the change to what it affected.** Tie the change to the relevant claim, amount, source, percentage, question, headline, section or article where possible.
4. **Preserve and propagate proportionately.** Keep enough prior state to reconstruct the transition, expose old states appropriately, and correct downstream surfaces still under publisher control.

The exact interface can vary.

The exact retention policy can vary.

The exact object model can vary.

But the distinction is stable.

Version history tells us what changed.

Correction notes tell us why.

Correction memory should keep the connection between them.

Journalism should remember not only what it says now.

It should remember why it stopped saying what it said before.

RELATION MEMORY

NEARBY [Sources Should Travel With Claims](#) · Previous instalment in Journalism, Reassembled. Article 005 preserves the evidence relationship attached to a claim; Article 006 preserves the reason a published state changed.

NEARBY [Claims Need States](#) · Claim state and correction memory meet when a published proposition changes verification status, wording or evidentiary basis.

NEARBY [Questions Should Not Disappear](#) · Both articles argue that material editorial state should survive the moment an article reaches its current ending.