What the Corpus Does Not Say

Reply structure, time and coverage in YouTube comment data — what linguists and other researchers should check before treating comments as conversation

Max Sedlmair · ORCID 0009-0007-2471-5595 · Independent Research, Germany · research@woidwide.net

Version 1.7 · 30 October 2026 · Not peer reviewed. Refutations, review, or comments are much appreciated.
Premises last checked: *24 October 2026, tool source and platform behaviour (yt-dlp 2026.10.21): all unchanged. Re-checked every six months under a fixed protocol; next check May 2027 (Appendix D).*
A consolidated edition that supersedes three preprints by the same author: Six Methodological Pitfalls in YouTube Comment Extraction (v2.7.1), Flattened Conversations (v3.8.1) and Reconstructed Time (v1.2).
All versions (resolves to the latest): https://doi.org/10.5281/zenodo.23339873 · Verification package: https://doi.org/10.5281/zenodo.23442987
Part of a series on YouTube comment corpora: What the Corpus Can No Longer Find, https://doi.org/10.5281/zenodo.23642703 · What the Corpus Still Holds, https://doi.org/10.5281/zenodo.23805575

Abstract. Linguists increasingly read YouTube comment sections as conversation: who answers whom, how address is marked, what follows what, and when. None of this is observed directly. It is reconstructed twice — by the platform when it stores and serves a comment, and by the extraction tool when it reads it back. That thread position and actual address can come apart is established in CMC research, including for YouTube. This report adds measurements on the author's own YouTube corpora of how far, where and why they come apart under two extraction routes, and closes with checks that take minutes to run. For discourse and conversation analysts, pragmaticians and CMC corpus linguists, the result is a map of which readings of comment data — adjacency, address, sequence and time — the delivered fields can support, and which they cannot.

Six findings are reported. (1) The official Data API attaches every reply to the top-level comment: across 43,896 replies its parent field never pointed anywhere else, and in two videos collected by both routes 48.5% and 42.3% of replies were assigned different parents. (2) Interface-level extraction recovers deeper structure but is capped: across 210 extractions of 133 videos (480,925 comments) no reply was returned below depth 3 under the yt-dlp release and serving conditions tested, a parameter sweep shows that the extractor's depth setting does not explain the cap, and at the capped level the platform-inserted address marker and the recorded parent disagree in 45.8% of resolved cases against 3 in 13,407 one level up. (3) The timestamp emitted by interface-level extraction is a back-calculation from a relative label such as "1 year ago"; in one extraction 182,784 comments occupy two values, and median error against the API ranges from 4.1 to 358.3 days. (4) The address marker is a pointer with artefacts of its own: invisible characters hide 3–4% of markers from a standard pattern, the API glues the marker to the following text in 15.7% of cases where the interface does not, and resolving markers against today's handles manufactures part of an apparent behavioural trend. (5) Corpora silently lose comments through deletion, through reply limits set to zero, and through blocked runs whose files carry no trace of the interruption. (6) A manual coding pass specified in advance confirmed 78 of 100 divergent cases as genuine, and the conclusion drawn from it was wrong, because the mechanism operated at the level of tree position, which no case-level coder can see.

The paper also records how the measurements came about — collection on a phone, an adversarial working loop between the author and several language models, correspondence with the developer of an independent tool, and thirty-three documented self-corrections — because that record may be useful to others working the same way. Apart from the dated re-checks of Appendix D, no figure in this edition is newly computed; all are carried over from the component reports. Every computable claim can be checked with one command: a deposited verification package recomputes 72 published figures from text-free tables and compares each with the paper (https://doi.org/10.5281/zenodo.23442987). This checks that the figures follow from the deposited tables; it is not

an independent validation of the original structural assignments. Because tools and platforms change, the premises are re-checked every six months under a fixed protocol and each result is published as a new version (Appendix D). The first re-check, on 24 October 2026 with a newer yt-dlp release, found every premise unchanged: no reply below depth 3, marker–parent divergence of 2 in 7,085 at depth 2 against 46.4% at depth 3, reconstructed timestamps, and a flat API. Section 11.1 sets out what the findings mean for large text studies — sentiment, stance, topic — that never look at threads.

Keywords: YouTube comments, reply structure, data quality, corpus construction, extraction methodology, yt-dlp, YouTube Data API, timestamps, computer-mediated communication, research provenance

The short version

- **Two ways in, two different trees.** The official API files every reply under the first comment of its thread. Interface tools such as yt-dlp keep who answered whom — but only three reply levels deep.
- **The deepest level is not what it looks like.** Replies to deeper conversations are filed at the third level regardless of whom they answer. There, a reply's recorded parent and the person it addresses disagree almost half the time; one level higher they almost never do.
- **The @-name is mostly not typed by the speaker.** The interface inserts it in about nine of ten nested replies. It is a platform feature first and an address form second.
- **The timestamps are estimates.** yt-dlp turns "1 year ago" into an exact-looking date. In one corpus 182,784 comments share two dates.
- **Files can be incomplete without saying so.** Replies switched off by one setting, runs that were blocked halfway, a 2,500-comment ceiling — none of it is visible in the data.
- **Careful hand-coding did not catch the main problem.** It confirmed a pattern that turned out to come from the tree structure. One grouped count would have shown it.
- **This matters even if you never look at threads.** Sentiment and topic studies on millions of comments inherit the time, coverage and text problems; more data does not cancel them (§11.1).
- **The premises are kept current.** Tools and platforms change, so the key facts are re-checked every six months; the first page says when they were last checked (Appendix D).

Appendix A lists ten checks that find all of this in minutes.

Why this paper exists, and for whom

This paper is meant to be useful before it is meant to be cited. It is written above all for linguists — discourse and conversation analysts, pragmaticians, CMC and corpus researchers — who use YouTube comments as data, and for anyone about to build such a corpus: a student with a thesis, a researcher new to platform data, someone handed a file another person collected.

The stakes are linguistic. Adjacency, sequence, address and turn-taking are the categories such work is built on, and on YouTube each of them is read off a field the platform or the tool has reconstructed. Where a reply sits in a thread, whom an @-name refers to, which turn came first: all three can be wrong in ways that look like findings about speakers. The author made that mistake himself, which is how this paper began.

None of the failures described here was visible in the data when it happened. Each was found later, usually after a result had been written down. The paper therefore opens every finding with a check that costs one line, and gathers the checks in Appendix A. Readers short of time can read the box above, §1.1, §11 and Appendix A, and come back to the evidence when they need it.

The measurements first appeared in three overlapping preprints that cited and corrected one another. This edition replaces them with one text, together with an account of how an untrained researcher working with language models arrived at them (§13). A short glossary follows for readers who do not work with platform data every day.

This paper is the first part of a small series on YouTube comment corpora. *What the Corpus Can No Longer Find* (Sedlmair, 2026f) asks whether a deposited corpus can still be traced back to the platform years later, and *What the Corpus Still Holds* (Sedlmair, 2026g) asks how much of a deposited corpus survives and whether the loss changes its results. The three papers complement each other: each looks at one stage in the life of a corpus, from collection to deposit to the years after.

Terms used throughout. *API* — the YouTube Data API, Google's official interface for retrieving data; it needs a key and has a daily quota. *Interface-level extraction* — reading the data the YouTube website itself receives, here with the open-source tool *yt-dlp*. *Top-level comment* — a comment posted directly under a video. *Reply, parent* — a comment posted in answer to another; the comment it answers is its parent. *Depth* — how many reply steps a comment is from the top level (top-level = 0). *Address marker* — the @-name at the start of a reply. *Thread* — a top-level comment with all its replies.

1. Introduction

A large and growing body of work treats online comment threads as interaction. Studies ask which contributions get taken up, how disagreement propagates through a thread, whether reciprocity predicts reply quality, and how participants organise address across turns. All of this rests on a structural claim: that the data record who replied to whom, and when.

The claim is weaker than it looks. A comment thread as delivered by a platform is not a transcript of an exchange. It is a data structure the platform maintains for its own purposes — display, notification, ranking — and that a researcher recovers through an extraction tool built by a third party. Between the interactional event and the analysable row of a corpus lie at least two reconstructions, and neither is guaranteed to preserve what the analysis assumes.

1.1 The case that started it

The work began as a substantive project. YouTube replies often carry a leading address marker — `@handle` — and the platform separately records a parent comment. In a corpus of 143,421 comments from a single debate video, the two disagreed in 29.0% of resolvable cases. That looked like a phenomenon: speakers structurally continuing in one place while addressing someone else.

A kill test was specified in advance: draw 100 divergent cases at random, reconstruct the thread context, code conservatively, and abandon the project if more than 75% proved to be artefacts. Seventy-eight of 100 were coded as genuine address divergence. The project passed its own test.

It should not have. Breaking the same rate down by structural depth produced this:

Depth	resolved leading-@ replies	divergent	rate
1	628	564	0.898
2	13,407	3	0.0002
3	20,663	9,468	0.458

Three divergent cases in 13,407 at depth 2, and 45.8% one level down. (Figures under the documented resolver rule of Appendix B; the earliest versions reported 4 of 13,439 and 9,485 of 20,724 from an undocumented first implementation, §5.4.) A behavioural regularity produces gradients. A categorical jump between adjacent structural levels is a property of the data structure. The coding had been correct case by case, and the inference drawn from it

had been wrong. Sections 4–9 explain why, and what else the same corpora showed once they were examined as instruments rather than as evidence about speakers.

1.2 Contributions

1. A head-to-head measurement of the official API against interface-level extraction on the same material, confirming that the API's documented two-level model discards immediate reply-to-reply parentage, and quantifying how much (§4).
2. Documentation of a second, undocumented limit in the interface structure — four levels in total, depths 0 to 3, in the release and serving conditions tested — with a parameter sweep showing that the extractor's depth setting does not explain it, and a measurement of the address divergence the limit manufactures (§5).
3. A measurement of the timestamp that interface-level extraction emits: deterministic in the relative label, error up to a year, non-reproducible across a clock boundary, and in one configuration emitted without the label that would reveal it (§6).
4. Four further failure modes — artefacts of the marker layer, handle migration, orphan re-parenting, silently suppressed or interrupted collection — each with a diagnostic and an effect size (§5.4, §7–8).
5. A documented case in which a correctly specified and executed manual validation supported an incorrect inference, and the structural check that would have caught it (§9).
6. A consequence that links structure and time: reconstructing reply targets from address markers requires a chronology that interface timestamps cannot supply (§10).

None of these is offered as a discovery about YouTube. Platform engineers know their limits, and several of the mechanisms are acknowledged in tool source code. Nor is the general point new to linguistics: that thread position and actual reference come apart on YouTube is described by Herring and Chae (2021), that formal replies miss textually addressed ones was noted early (Thelwall, Sud & Vis, 2012; Ksiazek, Peer & Lessard, 2016), and that platform data reflect the conditions of their collection is a long-standing warning (§2). The contribution is measurement: the size of each effect, where in the structure it sits, the diagnostics that find it, and the demonstration that each limit propagates into research data where it can be read as behaviour.

2. Related work

Coherence and reply relations in CMC. That computer-mediated interaction disrupts the adjacency of turns is long established (Herring, 1999). CMC corpus annotation already separates *technical* replies, realised through a platform's reply affordance, from *interpretative* replies inferred from address markers, quotation or greeting forms (Herzberg & Lüngen, 2019); the TEI CMC-core schema encodes the same separation (Beißwenger & Lüngen, 2020). For YouTube specifically, the closest conceptual precedent is Herring and Chae (2021). They show that the platform makes many addressees available at once, that what a comment responds to must be established from addressee and topic together rather than read off thread form, and that the flat display of threads can hide whom a reply to a reply answers. They separate technical linkage, address and content relation. The general insight on which this paper builds — that thread position and actual reference diverge — is therefore theirs, and earlier work had already noted that counting formal replies misses answers given without the reply function (Thelwall, Sud & Vis, 2012; Ksiazek, Peer & Lessard, 2016). Coherence on YouTube survives this fragmentation in other ways, for instance through lexical chains across comments (Tanskanen, 2021). What this paper adds is measurement on one platform: how far the two pointers diverge, at which depth, and that on YouTube the divergence is largely determined by a storage or serving limit rather than by speakers. Self-replies, which recur in what follows, are a described CMC phenomenon in a quite different architecture (Tanguy, Poudat & Ho-Dac, 2024).

Reconstructing threads. Recovering who replies to whom from flat or partial structure is an established task for discussion forums (Aumayr, Chan & Hayes, 2011). For YouTube, Bassi et al. (2024) reconstruct comment threads

with a combination of rules and language models, explicitly to compensate for what the API does not provide, and later work uses the reconstructed trees, for instance to extract chains of depth 4 and more (Bassi et al., 2025; see also Bassi, Vieira & Pereira-Fariña, 2026). Annotation tools now give analysts the video, metadata and thread context of each comment (Fernández-Martínez, 2026). These works provide and use context; they do not measure which reply relations the delivered context preserves, which is the question here. Recent YouTube comment datasets document scale and diversity but carry no explicit reply edges (Dutta & Khan, 2025), and their extraction and completeness checks are documented more briefly than the measurements below require.

Platform artefacts and API audits. Platforms are not neutral observation windows; researchers who ignore how data are stored and served may be measuring engineering decisions (Tufekci, 2014; Ruths & Pfeffer, 2014). Unannounced platform change can silently invalidate an instrument (Lazer et al., 2014), shifting access regimes threaten replicability (Bruns, 2019; Tromble, 2021), and different collection tools applied to the same stream yield materially different datasets and network results (Weber et al., 2021). Error frameworks for digital trace data separate errors of platform coverage, data collection, processing and inference, and ask that each be documented (Sen et al., 2021; Olteanu et al., 2019); platform control over API access adds a further threat to replicability (Davidson et al., 2023). The six findings below can be read as measurements of specific collection and processing errors in this sense. For YouTube comments, Thelwall, Sud and Vis (2012) already recorded an API ceiling of 1,000 comments per video and noted that retrieved replies may answer comments outside the retrieved set — an early case of the coverage problems in §8. For time, Liou (2026) shows on API post timestamps that comment likes cannot be interpreted without arrival time; his design is more systematic than this paper's on that sub-problem, and it concerns top-level comments with exact API times, not the reconstructed timestamps of interface extraction measured in §6. For YouTube search, Rieder, Padilla and Coromina (2025) audit the search endpoint of the Data API and find incompleteness, inconsistency between identical queries and temporal decay; Efstratiou (2025) reports converging evidence for the search endpoint and, in a secondary test, finds the comment endpoints stable across collections, returning almost all comments for a queried video. These audits concern which videos and comments a researcher receives; this paper concerns how the comments received are linked to one another.

Research software as mediator. Rieder, Peeters and Borra (2024) argue that research software is not a neutral conduit, and that the practical investigation of data structures, API specifications and technical idiosyncrasies — *technical fieldwork* — is itself epistemically productive. This paper is an instance of that practice and adopts the term. Tutorials and ready-to-use tools now make API-based collection accessible to researchers with little programming experience (Campos-Moinier, 2026); the checks in Appendix A are meant to complement such tools.

What was already on record about the limits. The API's two-level model is in its published reference: replies are supported only for top-level comments (Google Developers, 2026a, 2026b). yt-dlp added support for nested comment subthreads on 31 December 2025 (pull request #15448), and repository discussion describes a staged rollout (issue #14723); neither states a maximum depth. One user remark on Reddit notes that nesting appears to go about three layers deep. A developer answer on Stack Overflow describes a `replyLevel` field in the web client's payload that does not appear in our extracted files. The closest prior statement is in research software: YouTube Data Tools v2 (Rieder, 2026) reconstructs reply targets from address markers and notes that inferred levels can exceed the threading depth the interface records, supplying a constant to cap them. A comment in its source gives two informal values: the interface indents four levels visually, and threading has been observed only to about level 6. Its author describes both as results of informal probing in July 2026, not of measurement, and explains that the deeper value is counted along the mention chain by design (Rieder, personal communication, 21 September 2026). Read that way, both values agree with the cap measured in §5. For time, the yt-dlp release that introduced comment timestamps (2021.04.11) called them "approximate", the source carries a comment marking the value as an estimate, and issue #12158 reports day-level precision; a recent practitioner note makes the same point about relative ages and shows how a reply budget concentrates a small sample in few threads (Iakovlev, 2026). None of these sources quantifies the loss. The informal sources carry no weight in any claim made here.

Version 1.2 adds the works on thread reconstruction, collection-tool variation and YouTube comment data above after an extended literature search (§14). Version 1.6 adds the error-framework and access literature and a collection tutorial (Sen et al., 2021; Olteanu et al., 2019; Davidson et al., 2023; Campos-Moinier, 2026).

3. How the corpora were built

This section is written in more detail than a methods section usually is, because the way the material was collected is itself part of the evidence for §8, and because readers who collect in similar conditions may recognise their own setup.

3.1 Setting

The author worked without institutional infrastructure. Collection ran in Google Colab notebooks, operated from a mobile phone, with files written to Google Drive so that a disconnected session could resume; some processing ran in Termux on the same phone. Each notebook installed the current yt-dlp release at runtime (`pip install -U yt-dlp`). Version 2026.08.19 was current throughout the main collection window. Where the raw `.info.json` files survive, the version is stored inside each file (`_version`: 2026.08.19, git 594bd50c); for exports without that field it is not recoverable per run (§12). API collection used a personal API key under the standard daily quota. All analysis scripts were written to run with the Python standard library only.

The corpora were not collected for this paper. They were collected between August 2025 and September 2026 for substantive projects — a debate video on a science controversy, news events, school shootings, storm names, debate channels — and reused here as measurement material. No claim about their content is made.

3.2 Two groups of extractions

Only extractions in which replies were actually collected can test a depth limit, and the distinction is essential.

Group A — with replies. 210 extractions of 133 distinct videos, 480,925 comments, of which 238,700 are replies. Extractions outnumber videos because several videos were collected more than once, under different rankings or limits; both counts are reported throughout and repeated extractions are never treated as independent videos.

Corpus	extractions	distinct videos	comments	replies	max. depth
Debunking video (main)	1	1	143,421	89,046	3
Leipzig	12	12	14,104	6,299	3
School (capped)	12	12	30,000	8,367	3
School shootings	4	4	9,000	3,768	3
Further collections	181	104	284,400	131,220	3
Total	**210**	**133**	**480,925**	**238,700**	**3**

Group B — without replies. 270 extractions of 265 distinct videos, 92,504 comments, no replies. Two of these collections, together 258 videos and 89,074 comments, are the evidence for §8.2: the reply layer was absent and nothing in the files said so.

A binding ceiling. Every one of the 12 videos in the School subcorpus returned exactly 2,500 comments, an administrative per-video ceiling. Across the collection, 33 extractions of 28 distinct videos sit at exactly 2,500. Where it binds, the retained comments are those the platform surfaced first under the requested ranking. Depth findings are within-video and unaffected; no count in this paper should be read as a comment-section total.

3.3 Extraction parameters

All interface-level corpora were extracted with yt-dlp's `--write-comments`. Comment limits are passed as a positional list under the extractor argument `max_comments`; at version 2026.08.19 the extractor reads five positions:

```
max_comments, max_parents, max_replies, max_replies_per_thread, max_depth, *_ = (
    int_or_none(p, default=sys.maxsize)
    for p in self._configuration_arg('max_comments') + [''] * 5)
```

(`yt_dlp/extractor/youtube/_video.py`, lines 2578–2579.) Every unset position receives `sys.maxsize`. The depth position was left unset in every collection except the sweep runs of §5.2, which set it deliberately and are counted in the totals of §3.2. In one early sweep run the parameter did not take effect; it is listed in the verification package (§16). The user-facing README documents four positions and not the fifth; a researcher who reads the documentation rather than the source will not learn that depth is controllable at all.

3.4 API collection

For two videos the comments were also retrieved through the YouTube Data API v3: `commentThreads.list` (`part=snippet`, `maxResults=100`, `order=time`) paginated to exhaustion, then `comments.list` with `parentId` set to each top-level comment that reported replies. Both the identifier passed to `comments.list` and the `parentId` returned in each reply were recorded, so that the API's own parent assignment could be examined rather than assumed. For the first video (`70vYj1KPyT4`) the API collection is complete. For the second (`JDy95_eNPzM`, the main corpus) it was interrupted by quota exhaustion at 68,793 comments, and comparisons are restricted to the intersection of the two exports.

3.5 Three layers that must not be merged

Layer 1 — the API data model. A thread is a top-level comment with a flat array of replies. Two levels. This is the published specification and not a defect.

Layer 2 — the interface structure. A reply's parent may itself be a reply, down to depth 3. This layer carries parentage that Layer 1 does not, and it has a limit of its own.

Layer 3 — what speakers do. Who addresses whom, marked in the text.

Layer 1 loses the structure and Layer 2 loses the time; no single source gives both. Reading Layer 2 as though it were Layer 3 is the central error this paper documents. It is the researcher's error, not the platform's.

3.6 Analytic conventions

Two conventions, both adopted after a failure. **Ancestry is structural, never temporal:** every statement about what came earlier is derived by walking the `parent_id` chain, never by sorting on `timestamp` (§6 explains why). **Retrieval counts are not findings:** pattern matching produces candidates; counts reported as results were hand-inspected or are labelled as retrieval only.

4. Finding 1 — The API flattens the tree

In short: the official API cannot tell you who answered whom below the first level.

Mechanism. The Data API represents every reply as attached to the thread root.

Evidence. Across all 43,896 replies retrieved via `comments.list` for the main corpus, the returned `parentId` never differed from the identifier passed, never pointed to anything other than a top-level comment, and was never empty. The documented model behaves as documented.

Head-to-head on the same material:

Video	replies compared	parent identical	parent differs	share differing
70vYj1KPyT4	6,248	3,216	3,032	48.5%
JDy95_eNPzM (intersection)	41,885	24,181	17,704	42.3%

Every divergent case has the same shape: the interface export assigns the reply to another reply, the API export to the top-level comment. A chain A → B → C becomes A → B, A → C; C's prior turn becomes A instead of B. Across four interface-level corpora (n = 267,505), the share of replies whose parent is itself a reply ranged from 10.8% in an entertainment video to 65.5% in a debate video — highest exactly where interaction research looks.

Coverage. For the one video where both collections completed, the API export contained 603 comments absent from the interface export, and the interface export none absent from the API export. One completed comparison is a thin basis for a coverage claim; it is reported as observed, and it is consistent with Efstratiou's (2025) finding that the comment endpoints return almost all comments for a queried video.

Diagnostic. Share of replies whose parent is another reply. Zero means an API-derived corpus with no reply-to-reply structure.

Consequence. Studies that infer immediate reply-to-reply structure from API parent relations risk systematic flattening. Treat API parents as thread membership, not adjacency. Keep `api_parent_id` and `interface_parent_id` under separate names and never collapse them.

5. Finding 2 — The interface structure is capped

In short: interface tools recover the deeper tree, but only to depth 3, and the last level is partly forced.

5.1 The cap

Across 210 extractions of 133 distinct videos, no extraction returned a comment deeper than depth 3. Depth 3 itself is reached in 189 of them; the others are 18 small comment sections that never run that deep and three sweep runs with the depth parameter set to 2 (§5.2). Version 1.0 stated that maximum depth was 3 in every case, which was too strong. The cap is a finding about the yt-dlp release (2026.08.19) and the serving conditions under which these extractions ran, not a permanent property of the platform (§12).

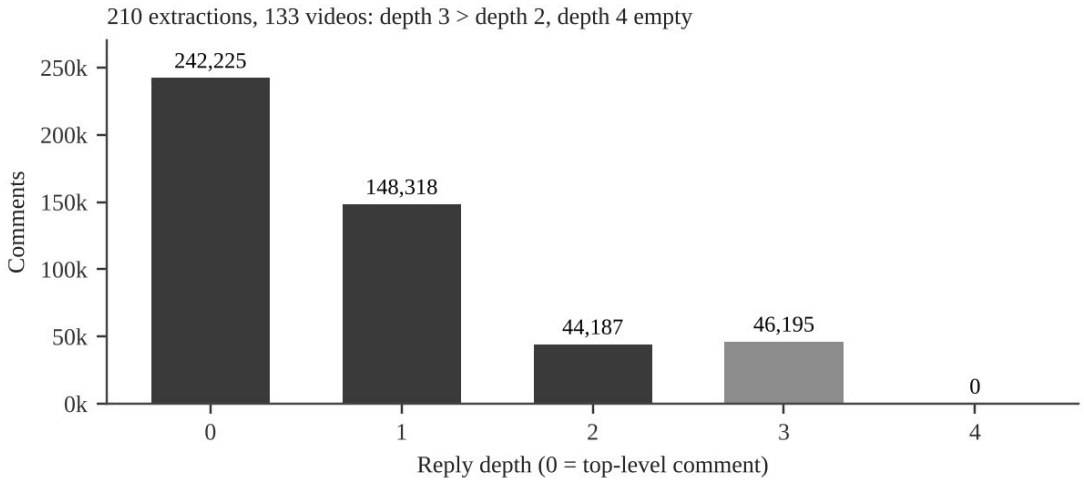


Figure 1. Depth distribution pooled across 210 extractions. Depth 3 carries more comments than depth 2; depth 4 is empty. Drawn from the counts in the table of §5.1 of Sedlmair (2026d).

Depth	0	1	2	3	4
comments	242,225	148,318	44,187	46,195	0

A distribution that ends because conversations run out thins towards the tail. This one thickens and then stops: pooled, depth 3 carries more comments than depth 2, and in 55 extractions covering 37 distinct videos the deepest level is larger than the level above it. Absent a cap, some of that mass would appear at depth 4. It appears nowhere.

5.2 The cap is not the extractor's depth setting

Source. The depth parameter defaults to unbounded and was not set (§3.3). **Annotation.** yt-dlp's own extractor code describes YouTube's subthread structure as four levels deep. **Parameter sweep.** Setting the depth parameter explicitly produces a response that saturates below the permitted value:

Video	max_depth = 1	= 2	= 9
tCk7oY-q2-s	0	1	**3**
udulQG5aYv8	0	1	**3**
uwm_iGo0cx8	0	1	**3**

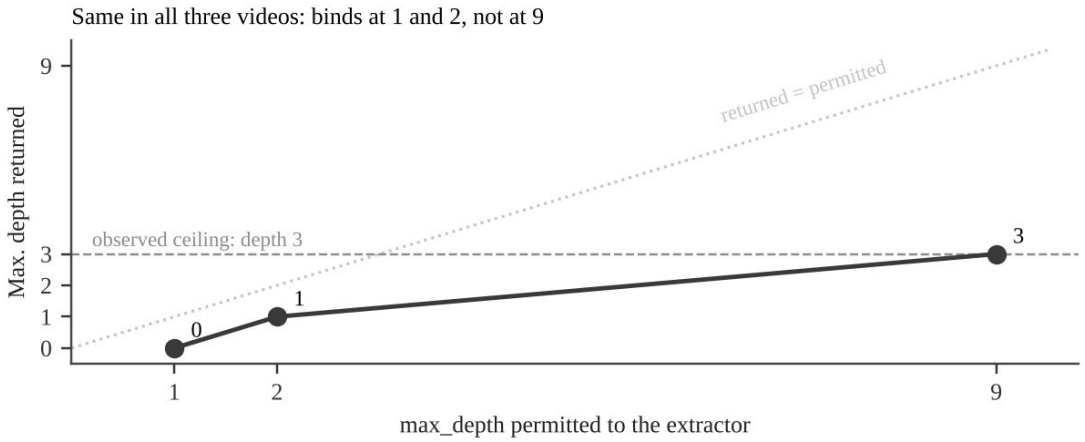


Figure 2. Parameter sweep. The extractor was permitted nine levels and received four (depths 0–3).

The parameter binds at 1 and 2 and does not bind at 9. This shows that the configured depth limit does not explain the cap. It does not by itself exclude every other cause in the extraction path, such as the way the client requests subthreads, and it does not locate the limit further: a storage rule, a response-assembly rule and a rendering limit would look identical in these data. "Platform rather than tool" is the most plausible reading, not a demonstrated one. A comparison with an earlier yt-dlp release was attempted and is no longer possible; releases from before mid-2025 no longer retrieve YouTube content.

The two informal values in the source of YouTube Data Tools (§2) agree with this. Four visually indented levels are four levels in total, depths 0 to 3. The deeper value is counted along the mention chain by design, because the API still reports whom a reply addresses after the interface stops nesting (Rieder, personal communication). It therefore describes the marker layer, not where the constraint sits, and this paper does not rely on either value.

5.3 What the cap manufactures

Once a conversation runs deeper than the cap, further replies are attached at depth 3 regardless of whom they address. The marker then names the actual recipient while the recorded parent names a node at the deepest available level. The attachment rule is not observed directly. What is observed is that at the capped level the recorded parent stops agreeing with the marker. **Edges at the deepest returned level are therefore censored interface attachments, not validated conversational parent relations.**

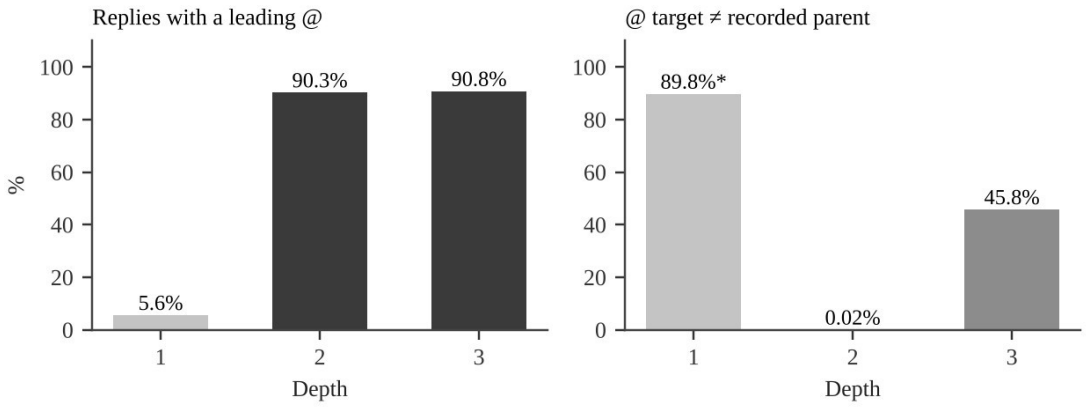


Figure 3. Main corpus. Left: the address marker is rare at depth 1 and near-universal below, i.e. largely interface-supplied. Right: where the interface supplies it and the chain is one step long, it agrees with the recorded parent almost perfectly; at the capped level agreement collapses.

Depth	replies	with leading @	rate	resolved	divergent	divergence
1	50,596	2,853	0.056	628	564	0.898
2	14,973	13,525	0.903	13,407	3	**0.0002**
3	23,477	21,312	0.908	20,663	9,468	**0.458**

For linguistic work on address this matters before anything else: in nested replies the @-name is, in nine cases out of ten, put there by the interface when the user presses *reply*. It records the button, not a choice of address form. One of the author's own papers described it as the speaker's signal and had to be corrected (Sedlmair, 2026e, v1.1).

Pooled across three further corpora the marker rates replicate (0.039 / 0.951 / 0.961); depth-2 divergence is 0 of 4,707, zero in each corpus separately, and depth-3 divergence is 0.502. Depth-3 divergence targets point upward in the chain — the grandparent in 53.8% of cases, another thread participant in 43.5%, a sibling of the parent in only 1.7%, the root author in 1.1% (9,467 of 9,468 classified). Free switching of recipients would produce siblings.

A separate corpus of 16 videos, analysed for a different question, shows the same split: no divergence in 4,838 cases at depth 2 against 41.5% at depth 3 (Sedlmair, 2026e).

Sensitivity to the resolver. The counts depend on how an in-text mention is matched to an author (Appendix B). Under the earlier published rule — ASCII handles only, invisible characters not removed — they are 3 of 12,804 at depth 2 and 9,234 of 20,037 at depth 3 (0.0002 and 0.461). Absolute counts move by a few per cent between rules; the categorical difference does not.

5.4 The marker has artefacts of its own

The argument so far compares two pointers and treats one of them, the marker, as clean text. It is not. The developer of YouTube Data Tools reported three properties of the marker as the API serves it: the inserted mention is often glued to the following text without a space; it can sit behind zero-width characters; and it is editable text while the parent field is not (Rieder, personal communication). A fourth was found in checking our own resolver: its published character class admitted ASCII letters only.

Each was tested on the main corpus under three resolver variants: the earlier published rule; the same with invisible format characters (Unicode category Cf) removed; and that with Unicode letters admitted in handles.

Depth	Variant	leading @	resolved	divergent	rate
2	earlier published rule	12,914	12,804	3	0.0002
2	Cf characters	13,512	13,385	3	0.0002

Depth	Variant	leading @	resolved	divergent	rate
	removed				
2	Cf removed, Unicode handles	13,525	13,407	3	0.0002
3	earlier published rule	20,664	20,037	9,234	0.461
3	Cf characters removed	21,291	20,639	9,460	0.458
3	Cf removed, Unicode handles	21,312	20,663	9,468	0.458

Invisible characters hide markers. Removing them adds 598 markers at depth 2 and 627 at depth 3 — 4.4% and 2.9% of all markers. A regular expression's whitespace class does not match them, so a reply beginning with one is not recognised as marked at all. Marker rates computed without removing them are lower bounds. The divergence rates barely move.

Glued mentions are an API artefact. For the one video collected completely both ways, all 13,484 comment identifiers occur in both exports. Of 2,848 replies whose interface text begins with a marker followed by a space, the API text omits the space in 446 (15.7%). In interface text the problem is practically absent: only 5 of about 34,000 resolutions at depths 2 and 3 needed the prefix fallback. Invisible characters occur in the same number of comments in both texts (671), so they are not specific to either source.

Why none of this explains the contrast. The marker is inserted at almost the same rate at depths 2 and 3, and nothing in these artefacts depends on depth. Whatever divergence they generate should appear at depth 2 as often as at depth 3; at depth 2 it appears 3 times in 13,407. That bounds their contribution at depth 3 to about 0.02%, against an observed 45.8%. The three residual depth-2 cases did not go through the prefix fallback and remain unexplained until they are inspected by hand. Edited markers could not be tested on this export.

A documentation failure found on the way. Rerunning the published rule on the original export of the main corpus reproduced the figures earlier versions had labelled an independent reimplementation, to the case, and not the headline figures (4 of 13,439; 9,485 of 20,724). Those came from a first implementation that was never documented. All mention-based figures in this paper are therefore given under the documented rule of Appendix B. Every conclusion holds under both. One figure moves noticeably: the pooled depth-3 divergence in three further corpora is 0.502 under the documented rule against 0.413 reported before.

5.5 The parent field is not noise

The argument above would prove too much if the parent field were noise. A negative control on reply-to-reply pairs compares the resolved @-target with three candidates:

Corpus ranking	pairs	recorded parent	thread root	random reply
TOP	4,104	82.9%	1.0%	0.9%
NEW	5,822	77.9%	6.0%	3.7%

The field carries real information. But these pairs pool depth 2 (agreement ≈ 100%) and depth 3 (54.2%); at roughly equal cell sizes the expected blend is about 77%, and 77.9% is observed. **An agreement rate computed across depths looks like moderate noise when it is a near-perfect regime and a broken one, averaged.** The TOP/NEW pair is also the only ranking contrast available; it is weak, and a matched ranking control remains outstanding. Both rows rest on the undocumented first resolver and have not been recomputed under the rule of Appendix B.

5.6 How much usable structure remains

The reply-to-reply edges gained over the API are not uniformly usable. Split at the cap:

Corpus	depth 2 (below cap)	depth 3 (capped)	share below cap
70vYj1KPyT4	1,521	1,511	50.2%
JDy95_eNPzM	14,973	23,477	38.9%
pxEWXGSIpAI	3,129	7,506	29.4%
UPrkC1LdlLY	470	121	79.5%

Restricted to edges below the cap, the usable gain over the API is 8.6% to 24.3% of replies across the four corpora, not the 42–49% of §4. That reduction is the honest figure. What remains is the best-supported structure in the data — across all four corpora, the marker agrees with the depth-2 parent in all but 3 of 18,114 resolved cases, against 48% to 82% at depth 3 — and the API supplies none of it — though agreement between two platform-mediated pointers is evidence of consistency, not proof that either records the comment actually responded to.

A blind coherence check on 50 parent–child pairs, with all @-mentions masked, judged 36 of 40 decidable pairs coherent (90.0%, 95% CI 76.9–96.0; 10 undecidable). Self-reply edges scored lower than cross-author edges (83.3% vs 95.5%; not statistically established). The check was designed before the cap was known and is not stratified by depth, so it provides no evidence about capped edges. A depth-stratified re-run needs only the existing data and instrument and is the most useful single test outstanding.

Diagnostic. Maximum depth per video and the depth distribution. Mass piling up at the maximum means a cap is binding.

Consequence. Record depth on every edge. Exclude deepest-level edges from sequence analysis, or report results with and without them.

6. Finding 3 — The timestamp is reconstructed

In short: a yt-dlp timestamp is a relabelled "x weeks ago", with errors up to a year.

6.1 Mechanism

YouTube's web interface shows only a coarse relative label ("3 weeks ago", "1 year ago"). yt-dlp computes an absolute Unix timestamp by subtracting the bucket width from the clock at parse time. The field looks like a precise publication time and is not one.

6.2 Evidence

Eleven extractions of seven videos (n = 461,083), collected 27–30 August 2026. In ten of eleven the mapping from label to timestamp is deterministic: each distinct label yields exactly one value, so the timestamp column is a reformatting of the label column.

Extraction	n	distinct labels	distinct timestamps	tie rate	effective time points
kirk (new)	182,784	4	2	0.961	1.10
oconnor (new)	57,748	66	57	0.396	4.57
mg_black (new)	48,580	39	26	0.868	1.37
mg_atheist (new)	46,589	26	18	0.237	4.88
shootings (final)	36,548	32	20	0.897	1.30
shootings (pilot)	34,476	0	42	0.234	7.63
Leipzig	14,104	28	20	0.664	1.89
kirk (top)	13,402	4	2	0.806	1.41
oconnor (top)	11,432	61	39	0.463	4.69

Extraction	n	distinct labels	distinct timestamps	tie rate	effective time points
mg_atheist (top)	9,244	24	16	0.176	6.43
mg_black (top)	5,176	37	24	0.642	2.35

Tie rate: the probability that two randomly drawn comments carry the same timestamp. *Effective time points:* 2 to the power of the Shannon entropy of the timestamp distribution.

At a tie rate of 0.961, “earlier than” is undefined for 96% of comment pairs. In the main corpus of §3.2, 143,421 comments carry 37 distinct values, and the largest single value holds 25.7% of the corpus. In three further corpora there are fewer distinct timestamps than labels (Leipzig 28 → 20; School 69 → 42; School shootings 24 → 17), so the label cannot even be recovered from the timestamp.

Validation against the API. Five videos were matched on comment identifier against `publishedAt`.

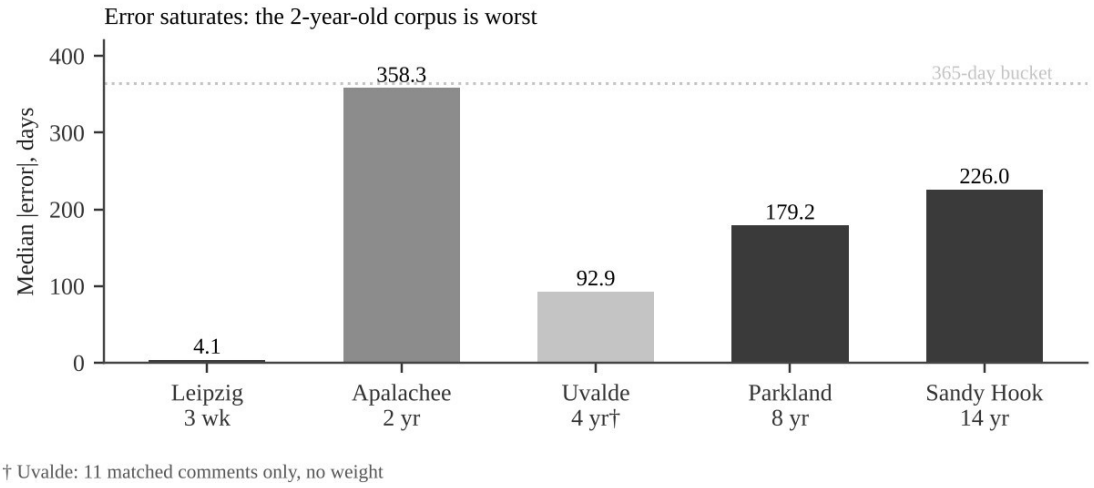


Figure 4. Median absolute error of the derived timestamp against the API, by age of the comments at collection. Above one year YouTube’s labels stop becoming finer, so the realised error depends on position within a 365-day bucket rather than on age.

Video	comment age	matched n	median abs. error (days)	p90 (days)
Leipzig	3 weeks	—	4.1	—
Apalachee	2 years	325	358.3	358.8
Uvalde	4 years	11	92.9	94.1
Parkland	8 years	147	179.2	365.3
Sandy Hook	14 years	619	226.0	362.9

The error saturates: a two-year-old corpus is worse off than a fourteen-year-old one. No observed error exceeds 365.3 days. The matched subsets are not random samples and Uvalde carries no weight.

Two silent failures. In one corpus the relative label is absent for all 34,476 rows while the derived timestamp is populated with 42 values; nothing in the file reveals that the value is reconstructed. In one extraction that ran across midnight UTC, 37 of 66 labels map to two different timestamps, so the derived value depends on when each comment was parsed and encodes extraction order as apparent time. This mode was searched for in four further corpora and not found.

6.3 Diagnostic

```
import collections
ts = collections.Counter(c.get("timestamp") for c in data["comments"])
print(len(ts), "distinct timestamps for", len(data["comments"]), "comments")
```

A ratio below roughly 0.01 indicates reconstruction; in these corpora it lies between 0.000011 and 0.0017. Where the label is present, check that each label maps to exactly one value; if not, the job crossed a clock boundary.

6.4 Consequence

Do not use interface timestamps for time series, latency, ordering within a thread, or temporal filtering. A temporal filter on reconstructed values silently admits or excludes the wrong rows; the author applied one in earlier work and reported its output as a result before the artefact was identified. Treat the raw label as ordinal and retain it in every export. Where time matters, retrieve `publishedAt` from the API; for 143,421 comments this costs roughly 2,870 quota units. A search for published work affected by the artefact examined five candidates and found the timestamp extracted and then not used in all five. It was not systematic, and the argument here is preventive rather than corrective.

7. Finding 4 — Handle migration

In short: an @-name written years ago may not match anyone's name today, and counting that as a mismatch invents a trend.

Mechanism. An @-mention in a comment body contains a name, never a channel identifier, so resolving it requires matching a string written at one time against a name field read at another. YouTube introduced handles in October 2022 and auto-assigned them from 14 November 2022. A comment from 2019 carries the display name of 2019; a 2026 extraction carries today's handle.

Evidence. Of 37,690 replies with a leading @ in the main corpus, 2,992 (7.9%) resolve to no author in the corpus, and unresolvability falls with the year of the comment (11.6% in 2019, 2.5% in 2024; years taken from the reconstructed timestamp, which is coarse but adequate at that resolution). Counting unresolvable mentions as mismatches produces an apparent decline in divergence from 42.8% to 23.3% over the corpus lifetime — a trend that looks like speakers changing their practice, about a third of which is a resolution artefact. Over resolved mentions only, the decline is shallower and non-monotonic (35.2% to 21.4%).

Diagnostic. Report the unresolvable-mention rate by time bucket alongside anything computed from resolved mentions. Parallel trends point at the resolver.

Consequence. Never treat "did not resolve" as "resolved to someone else". This rule was broken in an earlier version of one component report. Over resolved mentions only, aggregate agreement between marker and parent is 79.6%, 71.1%, 63.1% and 96.2% in the four corpora of §4 — figures that should only ever be read by depth (§5.6).

8. Finding 5 — Comments that are not there

In short: a corpus can lose comments without leaving any trace in the file.

Three failures with one signature: comments are missing, the absence leaves no trace, and the missing comments are not missing at random.

8.1 Orphans after deletion

When a parent comment is removed, its replies are re-attached to the thread root and keep their address marker, which now names someone whose comment is gone. Depth-1 replies with a leading @ are therefore anomalous (5.6% of depth-1 replies against about 90% below). Of 628 resolvable ones, 564 (89.8%) name someone other than the top-level author, and 176 of those name someone who appears nowhere in the thread; at depths 2 and 3 the corresponding figure is essentially zero (0 of 3; 1 of 9,468). They sit in very large threads (median 24 comments, mean 100, against a corpus-wide median of 1 and mean of 2.6). **Diagnostic:** leading-@ rate by depth; an excess at depth 1 is an orphan signature.

8.2 Replies switched off without anyone noticing

A value of 0 in the third or fourth position of `max_comments` suppresses replies entirely. The output still carries a parent field; the field is populated; every value is the root sentinel. The corpus looks complete. This happened twice, in two independent collections by the same author:

Collection	videos	comments	replies	parent field
Storm-name corpus (27 Aug 2026)	157	16,229	0	“root” in 100%
Debate-channel corpus	101	72,845	0	“root” in 100%

Both scripts carried `"max_comments": [str(MAX_KOMM), "all", "0", "0"]`. The first produced a preprint that stated replies were included (Sedlmair, 2026c); a corrected version was issued. Its proportions are unaffected, but its population is first comments, not comments. A later re-collection of the 28 largest videos of that corpus with the reply limits opened returned 9,953 replies under the same ranking, and an increase of 9,926 comments over the original files — almost exactly the replies, so the top-level layer had been complete and only the replies were missing. A mistake that recurs across projects for one researcher is a property of the interface as much as of the researcher. **Diagnostic:** distribution of parent values — one distinct value means one level only. It should be the first line run on any corpus.

8.3 Runs that were blocked

A long run can be rate-limited partway. The files already written remain well-formed and carry no marker of the interruption; only the run log knows. The storm collection’s log ends with a second pass blocked after 25 of 442 videos and a closing line that, read alone, reports an empty corpus. The debate-channel log stops at video 66 of 139 without a completion line; the 101 files on disk are what that run happened to deliver. Larger comment sections take longer to retrieve and are over-represented among the casualties. **Diagnostic:** keep and deposit the run log; compare videos attempted with videos returning data. A long tail of empty returns or a missing completion line is a block signature.

8.4 Round numbers

A per-video or per-corpus total that lands on a round value — 2,500, 1,000 — is an administrative ceiling until shown otherwise (§3.2). The same check found that one subcorpus of this study was capped in every video.

9. Finding 6 — Manual validation cannot see the tree

In short: coding cases one by one can confirm a pattern and still get its cause wrong.

This is the finding that would not have been reached by inspecting data alone.

9.1 What happened

The kill test of §1.1 drew 100 divergent cases with a fixed seed (20260831), reconstructed full thread context including ancestors and siblings, and coded conservatively. The specification was written into working notes before any case was drawn but was not deposited with a timestamp; it is described as specified in advance, not as pre-registered.

Category	n
@ names an earlier participant in the structural path	42
@ names another demonstrable thread participant	36
Addressing, target not identifiable in thread	17
Clear name/handle artefact	2

Category	n
Genuinely ambiguous	3

Seventy-eight of 100 — 78 of 83 identifiable — were coded as genuine. Every coding was correct: a coder looking at the thread does see the marker addressing someone other than the recorded parent. What no coder can see is that the same configuration occurs 3 times in 13,407 at depth 2 and 9,468 times in 20,663 at depth 3. The divergence is a property of where the comment sits in the tree.

9.2 Objections

"You selected on the dependent variable." Yes, by design: the question was whether observed divergence is genuine or artefactual, not how often it occurs. *"The coders were set up to fail."* They had full context and used it correctly. What they lacked was a view of the case's position in a distribution, which no case-level coding provides.

9.3 The check that would have caught it

Before coding anything, break the dependent variable down by structural position — depth, parent type, thread size — and look at the shape. A gradient across adjacent positions is consistent with behaviour. A categorical jump between adjacent levels is hard to reconcile with behaviour and easy to reconcile with a structural constraint. The figure was available before coding began and would have cost one grouped count.

9.4 What generalises

The claim is narrow and conditional: **where a generating mechanism operates at the level of structural position, case-level coding will confirm the phenomenon and mislead about its cause.** That case-level checks cannot establish a mechanism is not a new methodological principle. How often this happens is unknown. That it can happen in a correctly specified and executed test is what is reported, as a documented instance. Manual validation shows that a pattern is not a parsing artefact; it cannot show that the pattern is behavioural.

10. A third pointer, and why it needs a clock

Structure can be recovered a third way: from the address marker rather than from any stored edge. YouTube Data Tools v2 (Rieder, 2026) does this from API output: it processes the replies of a thread in chronological order and, where a leading mention resolves to an earlier participant, assigns the reply to that participant's most recent preceding comment, deriving an inferred parent and an inferred depth.

This changes the picture in two ways.

The cap acts differently on the marker. At the capped level the stored edge is forced while the marker still names a recipient, overwhelmingly upward in the chain (§5.3). A marker-based reconstruction is therefore not bounded by the cap in the same way, and may recover pointers the stored edge has lost. Its developer reports that the reconstruction has reached level 9 on a real thread (Rieder, personal communication). It also inherits every artefact of the marker layer (§5.4): on API text it meets a glued mention in about one marker in six, and the marker can be edited after posting while the edge cannot. The three pointers — API parent, interface parent, marker-inferred parent — fail at different depths and can be run on the same comments. Other marker- or content-based reconstructions exist: Bassi et al. (2024) rebuild YouTube conversation trees with rules and language models and report a manual check of reconstructed links (30 incorrect of 2,387; Bassi et al., 2025), and a recent dataset assigns @-replies to the mentioned user's earlier comment in the same thread (Mints, 2026). A comparison of such a reconstruction with both stored edges, by depth, has not been carried out here or anywhere the author is aware of. Where they agree below the cap and diverge at it, the divergence would locate the censoring. Inferred levels beyond

3 are by construction not interface edges and should not be reported as thread depth. None of the three is ground truth; §9 is an argument against reading any single pointer as such.

The marker-based method needs a real chronology. Two of its decisions depend on order, not merely on the set of comments: which comments count as *preceding* a reply, and which of a participant's comments is the *most recent* before it. The tool's source states the requirement and notes that the API guarantees no reply order. On interface timestamps that order does not exist (§6): at a tie rate of 0.961 the resolver would be choosing among tied candidates by file order, and across a clock boundary by an artefact of the parse clock. The tool cited draws from the API and is not exposed to this. A researcher combining an interface-level export with a mention resolver would be. **The API's time axis is therefore not only an input to temporal analysis. It is an input to structural analysis**, and the dependence is easy to miss because neither the question nor the output mentions time. This is an argument from the tool's stated requirements, not a measurement.

11. Recommendations for corpus builders

1. **Run the diagnostics in Appendix A before any analysis.** All of them together take minutes.
2. **Record the extraction stack in full:** tool, version, date, every extractor argument, including those left at default. "We used the defaults" is not a sufficient report when a parameter is undocumented.
3. **Keep the run log** and deposit it with the corpus. Coverage is a property of the run, not of the files.
4. **Use two sources for one corpus.** Structure from interface-level extraction, below the cap only. Time and coverage from the API. Keep the parent fields under separate names: `api_parent_id`, `interface_parent_id`, and, where available, a marker-inferred parent under a third name. Do not resolve disagreements between them by majority or preference; report which pointer supports which claim.
5. **Record depth on every edge**, and report any quantity computed over replies by depth. Treat deepest-level edges as censored: exclude them from sequence analysis or report results with and without them.
6. **Never use interface timestamps as a time axis** — not for series, latency, ordering, filtering, or as input to a reply reconstruction. Keep the raw label.
7. **Clean the marker before matching it.** Remove invisible format characters, admit non-ASCII handles, record every resolution that needs a prefix fallback, and separate "could not resolve" from "resolved to someone else". Report the resolution rate over time.
8. **Treat manual validation as necessary and insufficient.** Before coding, break the outcome down by structural position.
9. **Check for round numbers** and for a single distinct parent value — the two cheapest signatures of a corpus that is not what it appears to be.

For tool maintainers: a derived timestamp would be more honest as a null with the label preserved, or as an explicit interval. For reviewers: a paper reporting comment-level time series from scraped YouTube data should be asked how many distinct timestamps its corpus contains.

11.1 If you do not analyse threads

Most large studies of YouTube comments never use reply structure. Sentiment, stance, toxicity and topic analyses treat a corpus as a collection of texts, often millions of them. Three of the findings above still apply to that work, and scale does not dilute them. They are systematic rather than random, so a larger sample narrows the confidence interval around the same distorted estimate instead of correcting it.

Time (§6). Every curve over time, every before-and-after split around an event and every time window used to select comments inherits the reconstructed timestamp if the corpus was collected through the interface. Within a

year, comments collapse onto a handful of values; beyond a year, the median error reached 358 days in one video (§6.2). Temporal claims need the API's `publishedAt`, or the relative label treated as an ordered category and nothing more.

Who is in the sample (§3.2, §8). What a file contains depends on settings that are rarely reported. A reply limit set to zero removes every reply without trace; one re-collection of 28 videos recovered 9,953 replies that had been missing (§8.2). A per-video ceiling keeps the comments the platform surfaced first under the chosen ranking (§3.2). A blocked run drops a set of videos that is not random, because larger comment sections take longer to retrieve (§8.3). Deleted comments are absent while their replies remain (§8.1). The extractions labelled "top" and "new" in §6.2 differ several-fold in size; they also differ in other settings, so this paper does not attribute the difference to ranking, but it shows how far two collections of the same video can diverge. Each of these changes the population a rate describes. Whether a given sentiment or topic estimate moves with it has not been measured here and is an empirical question for each study. What can be required now is that the settings are reported, and that results are shown with and without replies where both are available.

What the text contains (§5.3, §5.4). In nested replies the leading @-name is inserted by the interface in about nine cases out of ten. For a text classifier it is platform text rather than the speaker's, and it is personal data: strip it before tokenisation and count it separately. Invisible format characters precede 3–4% of markers and can break tokenisers and regular expressions; normalise them first. API text glues the marker to the following word in about one marker in six, so the same comment can tokenise differently in two exports.

A minimum report for large text studies. Tool and version; collection date; every extractor argument, including those left at default; ranking; any per-video ceiling and how many videos reached it; whether replies were collected and their share of the corpus; the number of distinct timestamps; videos attempted against videos returned; how duplicates across repeated extractions were removed. Checks 1, 5, 8 and 9 of Appendix A produce most of this in minutes.

12. Limitations and open tests

One platform, one tool. All findings concern YouTube and yt-dlp. The class of failure is expected to generalise; the specific mechanisms and values are not claimed to.

Versions. The cap and the timestamp behaviour are findings about the releases and serving conditions tested, not permanent properties of the platform. Some collection notebooks installed the current yt-dlp release at runtime. For exports that do not keep yt-dlp's `_version` field the exact release is not recoverable per run; every raw `.info.json` checked for the verification package records 2026.08.19. The cap was tested at 2026.08.19; earlier releases can no longer be run against YouTube. YouTube introduced nested subthreads as a staged experiment, and a differently served client might see a different limit. For this reason the premises are re-checked every six months (Appendix D). The first check, on 24 October 2026, found the comment-extraction and timestamp code of the current release (2026.10.21) identical to the release tested and the platform behaviour unchanged on re-collection (Appendix D).

Repeated extractions are not independent videos and were not tested for stability under identical parameters.

Two extractions of the main corpus. The figures for JDy95_eNPzM come from one export (143,421 comments). A second extraction of the same video (143,140 comments, 31 August 2026) gives the same results under the same resolver (depth 2: 3 divergent of 13,389; depth 3: 0.457). Version 1.0 listed 143,140 as an unresolved recount and did not say which file carries the figures.

Incomplete delivery. The collection logs of the single-video extractions in §6.2 record repeated incomplete-data warnings (2 to 67 per run). Their comment counts are lower bounds. The timestamp findings, which are properties of the delivered files, are unaffected.

Resolved in version 1.1. The 182,784-comment extraction that version 1.0 listed as not located has been found in the author's archive and reproduces every figure given for it.

A control not recomputed. The negative control of §5.5 rests on an undocumented first resolver implementation and has not been recomputed under the documented rule.

Validation. The coherence check rests on one rater, 50 pairs and a coherence criterion, and is not stratified by depth (§5.6).

No independent execution. Every number here was produced by one implementation, written by a language model and run by the author, who cannot fully audit it. The resolver sensitivity check is a partial internal replication only. An independent party running the deposited scripts, or a second implementation, is needed before these figures should be treated as established. Running the verification package (§16) checks that the published figures follow from the deposited tables; it does not validate the parent and depth assignments in the original files, which would need an independent collection.

The prior-art search is not systematic. It missed the closest prior statement of the cap for seventeen days (§13), and an extended search for version 1.2 found further relevant work on thread reconstruction and YouTube comment data (§2).

Open tests, all possible on existing material: a depth-stratified coherence check; the three-pointer comparison by depth (§10); hand inspection of the three residual depth-2 divergences; a test of edited markers on an extraction that keeps the edit label; recomputation of the negative control; a matched ranking control. Where the constraint sits could be tested cheaply by following one long thread in the web client and recording the depth its own responses assign to each reply.

13. How this paper came about

This section records the path, because the path is part of what others may be able to use. It describes the author's decisions modestly and the model's work plainly. The author's own understanding of the technical parts is limited, and that is stated here rather than hidden.

13.1 From a phenomenon to an instrument

The work started in August 2026 with a linguistic question about address on YouTube (§1.1). The author collected the corpora on a phone, through notebooks a language model wrote for him, and asked whether a pattern in them was real. He set a kill test before looking at the cases, coded them himself, and the test passed. The model then broke the same figure down by depth, and the phenomenon dissolved into a property of the data structure. The author decided not to rescue the original framing. That decision turned a linguistic project into the measurement work reported here, and it was rejected twice more in later framings when the data did not carry them.

13.2 The working loop

The work ran in a loop of roles, adopted gradually and later written down by the author as a working protocol. One model (Claude, Anthropic) wrote collection and analysis code, computed figures and drafted text. Other models (ChatGPT, OpenAI; Gemini, Google) were asked to attack the drafts: to find prior work that would kill a novelty claim, to check statistics, and to act as hostile reviewers. The author carried material between them, decided which criticisms to accept, and decided what each paper would claim.

The adversarial reviews were productive in specific ways. One supplied the CMC annotation literature (Herzberg & Lüngen; the TEI CMC-core schema) and the platform-audit literature that the first drafts lacked, and pushed the framing from discovery to validation audit. Its own citation details were partly wrong and were later corrected

against publisher records; one reference it supplied did not exist and was removed. A second review found that the recommendation to take structure from interface-level extraction had to be limited to below the cap, that the cap could not be attributed to rendering, and that one coverage sentence did not follow from its own figures. Not every criticism was accepted: one review read an audit of the search endpoint as anticipating a finding about comment trees, and the paper keeps the two apart.

13.3 Correspondence

In September 2026 the author supplied the repository of YouTube Data Tools to the model during an unrelated session. Its documentation turned out to be the closest prior statement of the cap — missed for seventeen days because the prior-art search had looked for statements of the problem, not for tools that work around it. In the same session the model had drafted four revisions of a manuscript measuring what the author had already published, and did not recognise the overlap until he supplied his own reports.

The author then wrote a short note to the tool's developer, Bernhard Rieder, describing the measurement and asking whether he had had a value in mind for the depth his documentation mentions. Rieder replied on 21 September 2026. He showed where the values stood — in a source-code comment, where the model had overlooked them when it wrote that the tool states no value — stressed that they came from informal probing rather than measurement, explained that his deeper levels are counted along the mention chain by design, reported that his marker-based reconstruction had reached level 9 on a real thread, and described artefacts of the marker layer — glued mentions, zero-width characters, an editable marker against a non-editable parent field — as a possible source of residual divergences.

His observations were treated as hypotheses and tested. Glued mentions turned out to be an API-side artefact, present in 15.7% of marked API texts and practically absent from interface text; invisible characters occur in both and hide 3–4% of markers; none of it explains the contrast between depths 2 and 3 (§5.4). Rerunning the published resolver rule for that test exposed a documentation failure in the author's own papers, and every mention-based figure was restated. The correspondence therefore improved the work in ways no check of the author's had. Its points are used in §2, §5.2, §5.4 and §10 with the status Rieder gave them, and he is thanked in the acknowledgements.

13.4 Corrections

Across the component reports, two related preprints and this paper, thirty-three corrections are on record, several against results the author had already published. They are listed in Appendix C. Most were found by reading the series against itself, by the external reviews, or by a later project hitting the same data from a different side. Recurrent patterns: a figure obtained by subtraction rather than counting; a published method that was not the one behind the published figures; an unresolvable value scored as a substantive one; a recommendation about one field made without asking whether the next field was bounded too; and a literature search that looked for statements of a problem rather than for tools that quietly solve it. The author decided early that corrections would be recorded rather than silently applied. That decision is the one he would most recommend to others.

13.5 Why consolidate

The author's protocol for later projects asks for one measurement chain per paper, open to recomputation by strangers. Three overlapping preprints that corrected one another did not meet that standard for a reader arriving fresh. Consolidation was proposed by the model when the author asked which of his papers stood the best chance of peer review; the decision to consolidate, and to include this account of the path, was the author's.

14. Note on sources and scope

This paper cites work from fields the author does not work in. Reference entries have been checked against publisher or indexing records, and errors in earlier versions were corrected on that basis. That check establishes only that each cited work exists as described, not that every work has been read in full; the author has no institutional library access, and some citations rest on an abstract or a published quotation. The literature review is not exhaustive, as §13.3 demonstrates. For version 1.2 it was extended with an AI-assisted search engine (Consensus) and a second model's literature check; for version 1.6 a further Consensus check suggested the four works added in §2. Every added work was verified against publisher or repository records; Campos-Moinier (2026) was read in its abstract and front matter only. Where a citation is imprecise or a better source has been missed, the author would rather be told than defended against.

No claim of academic standing is made. The measurements are a separate matter: they are reproducible from deposited material, and whether they hold does not depend on how the surrounding literature is placed.

15. Provenance and AI disclosure

The author holds a Staatsexamen in English philology from the University of Regensburg. He has no institutional affiliation and no formal training in corpus methods, statistics or computer science. He understands the linguistic parts of this paper and only a small part of its technical machinery.

Language models did most of the technical work. Claude (Anthropic) wrote all collection, extraction and analysis code; computed every figure; located the depth parameter in the extractor source and designed the parameter sweep; produced the depth breakdown that overturned the original inference; found the reply-limit setting in the author's earlier scripts; read the logs that revealed blocked runs; read the source of YouTube Data Tools once the author supplied it; and drafted every component report and this consolidated text, including this section. ChatGPT and Gemini were used for literature searches and adversarial review.

The author's part was smaller and different. He chose the original question; collected every corpus; specified the kill test and its threshold before seeing the cases and carried out that coding; ran the API validation; supplied his earlier reports and, at the decisive point, the repository of the second tool; wrote to its developer; rejected framings the data did not support; decided which reviews to accept; and decided what each paper would claim and that corrections would be published rather than hidden. The responsibility for every claim is his.

Errors on both sides are on record. The model proposed explanations of the depth distribution that the data refuted (a rollout artefact; a version-dependent extractor limit), built an argument on a time trend that was a resolution artefact, twice drew conclusions from summary statistics the underlying tables did not support, presented as new a result the author had already published, missed the closest prior work, reported that a tool states no value when its source states two, and published a resolver rule that was not the one behind the headline figures — wrote both implementations and did not notice which one the appendix described. The author's kill test, correctly specified and executed, supported a wrong inference.

A note from the author. My interest is primarily linguistic: I want to know what YouTube comment data can tell us about language and conversation. That question kept leading back to the data themselves, and so this paper and the two that followed it have become, I hope, useful measurement work as well. I have worked as carefully as I could. Even so, I may have made mistakes or overlooked work that should have been cited, and I apologise to any author whose study is missing here. I would be grateful to anyone who checks the figures against the verification package, and who tells me where I am wrong.

How to read this paper. Every number was produced by code the author did not write and cannot fully audit. The claims most in need of independent checking are the location of the cap outside the extractor's depth setting (§5.2),

the near-zero divergence at depth 2 (§5.3), and the inference in §9. A reader who can refute any of them is asked to do so. As research by non-specialists working with language models becomes more common, the field will need documented cases, including their failures, against which to judge it. This paper is offered as one.

16. Data and code availability

Verification package. All computable figures in this paper can be recomputed with one command from text-free tables deposited at Zenodo: https://doi.org/10.5281/zenodo.23442987. In the reference run 72 checks pass, none fails, and five are reported as blocked because their input is manual coding or not included. The reference run was carried out by the author; it has not yet been repeated independently. Recomputation shows that the figures follow from the tables; it is not an independent validation of the structural assignments in the source files (§12). The package also lists the discrepancies found while building it (`KNOWN_ISSUES.md`), all corrected in this version. The script that re-checks the premises every six months (`premises_check.py`, Appendix D), the result file of each check, and a standalone implementation of the checks of Appendix A (`ytc_diagnostics.py`) are deposited with the verification package.

Apart from the dated re-checks of Appendix D, no figure in this edition is newly computed for the paper itself. Every number is carried over from the component reports and can be traced to them: structure and cap figures to Sedlmair (2026d), head-to-head and coherence figures to Sedlmair (2026a), timestamp figures to Sedlmair (2026b). Analysis scripts, per-extraction count tables, the depth-sweep extractions and the collection logs are deposited with Sedlmair (2026d) under CC0, including `recompute_v27.py`, which computes every mention-based figure under the rule of Appendix B, and `rieder_checks.py`, which reruns the marker-layer checks of §5.4. Scripts and tables for the other two reports are available on request and are to be deposited. Scripts use the Python standard library only. No comment text, comment identifiers or author handles are redistributed, in line with platform terms. Collection dates are recorded because comment sections change under moderation and deletion; exact reproduction is not promised.

Relation to earlier versions. This edition supersedes Sedlmair (2026a, v3.8.1), (2026b, v1.2) and (2026d, v2.7.1) as the reference text. Those records remain available with their full version histories; the correction register in Appendix C summarises them.

Acknowledgements

I thank Bernhard Rieder for pointing to the values in the YouTube Data Tools source, for his account of their status, and for the observations on the marker layer that §5.4 tests. He is not responsible for how they are used here, and any error in their interpretation is mine.

References

Aumayr, E., Chan, J., & Hayes, C. (2011). Reconstruction of threaded conversations in online discussion forums. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1), 26–33. https://doi.org/10.1609/icwsm.v5i1.14122

Bassi, D., Maggini, M. J., Vieira, R., & Pereira-Fariña, M. (2024). A pipeline for the analysis of user interactions in YouTube comments: A hybridization of LLMs and rule-based methods. *2024 11th International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 146–153. https://doi.org/10.1109/SNAMS64316.2024.10883781

Bassi, D., Marino, E. B., Vieira, R., & Pereira-Fariña, M. (2025). Old but gold: LLM-based features and shallow learning methods for fine-grained controversy analysis in YouTube comments. *Proceedings of the 12th Workshop on Argument Mining*, 46–57. https://doi.org/10.18653/v1/2025.argmining-1.5

Bassi, D., Vieira, R., & Pereira-Fariña, M. (2026). Navigating the disagreement space: A case study on persistent YouTube users' interactions in immigration-related discussions. *Proceedings of the 18th ACM Web Science Conference*, 165–175. https://doi.org/10.1145/3795766.3799757

Beißwenger, M., & Lüngen, H. (2020). CMC-core: A schema for the representation of CMC corpora in TEI. *Corpus*, 20. https://doi.org/10.4000/corpus.4553

Bruns, A. (2019). After the 'APIcalypse': Social media platforms and their fight against critical scholarly research. *Information, Communication & Society*, 22(11), 1544–1566. https://doi.org/10.1080/1369118X.2019.1637447

Campos-Moinier, K. (2026). Collecting data from YouTube using the official API: A tutorial and a ready-to-use tool. *Behavior Research Methods*, 58(8), 236. https://doi.org/10.3758/s13428-026-03116-8

Davidson, B. I., Wischerath, D., Racek, D., Parry, D. A., Godwin, E., Hinds, J., van der Linden, D., Roscoe, J. F., Ayravainen, L., & Cork, A. G. (2023). Platform-controlled social media APIs threaten open science. *Nature Human Behaviour*, 7(12), 2054–2057. https://doi.org/10.1038/s41562-023-01750-2

Dutta, H. S., & Khan, B. (2025). YTCommentVerse: A multi-category multi-lingual YouTube comment corpus (formerly YT-30M) [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2509.11057

Efstratiou, A. (2025). On YouTube search API use in research. *Proceedings of the 2025 ACM Internet Measurement Conference*, 919–927. https://doi.org/10.1145/3730567.3764492

Fernández-Martínez, N. J. (2026). YouTA: A context-aware corpus annotation tool for critical discourse analysis of YouTube comments. *Applied Corpus Linguistics*, 6(3), 100226. https://doi.org/10.1016/j.acorp.2026.100226 (online 17 June 2026)

Google Developers. (2026a). YouTube Data API v3 reference: CommentThreads. https://developers.google.com/youtube/v3/docs/commentThreads — retrieved 2 September 2026.

Google Developers. (2026b). YouTube Data API v3 reference: Comments: list. https://developers.google.com/youtube/v3/docs/comments/list — retrieved 2 September 2026.

Herring, S. C. (1999). Interactional coherence in CMC. *Journal of Computer-Mediated Communication*, 4(4). https://doi.org/10.1111/j.1083-6101.1999.tb00106.x

Herring, S. C., & Chae, S. W. (2021). Prompt-rich CMC on YouTube: To what or to whom do comments respond? *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2906–2915. https://doi.org/10.24251/HICSS.2021.354

Herzberg, L., & Lüngen, H. (2019). Types and annotation of reply relations in computer-mediated communication. *European Journal of Applied Linguistics*, 7(2), 305–332. https://doi.org/10.1515/eujal-2019-0006

Iakovlev, N. (2026, October 14). Ten YouTube comments, one commenter: What a reply budget does to your sample [Blog post]. DEV Community. https://dev.to/nikita_iakovlev_415524c19/ten-youtube-comments-one-commenter-what-a-reply-budget-does-to-your-sample-31ii

Ksiazek, T. B., Peer, L., & Lessard, K. (2016). User engagement with online news: Conceptualizing interactivity and exploring the relationship between online news videos and user comments. *New Media & Society*, 18(3), 502–520. https://doi.org/10.1177/1461444814545073

Lazer, D., Kennedy, R., King, G., & Vespignani, A. (2014). The parable of Google Flu: Traps in big data analysis. *Science*, 343(6176), 1203–1205. https://doi.org/10.1126/science.1248506

Liou, B. H. (2026). YouTube comment exposure: 1,345,353 comments with post timestamps [Dataset and preprint]. https://brianhliou.com/publications/youtube-comment-exposure/

Mints, O. (2026). YouTube comment dialogues: 399 parent comment – reply pairs from a socio-political discussion, translated into English (Version 1.0) [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.23014057

Olteanu, A., Castillo, C., Diaz, F., & Kıcıman, E. (2019). Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2, 13. https://doi.org/10.3389/fdata.2019.00013

Rieder, B. (2026). *YouTube Data Tools*, version 2 [Computer software]. Source `src/ytdt/modules/video_comments.py` (`resolve_reply_targets`, `MAX_INFERRED_LEVEL`) and README; snapshot `youtube-data-tools-v2-main`.

Rieder, B. (2026, September 21). Personal communication to the author (e-mail).

Rieder, B., Padilla, A., & Coromina, Ò. (2025). Forgetful by design? A critical audit of YouTube's search API for academic research. *Information, Communication & Society*, 1–20. https://doi.org/10.1080/1369118X.2025.2591767

Rieder, B., Peeters, S., & Borra, E. (2024). From tool to tool-making: Reflections on authorship in social media research software. *Convergence*, 30(1), 216–235. https://doi.org/10.1177/13548565221127094

Ruths, D., & Pfeffer, J. (2014). Social media for large studies of behavior. *Science*, 346(6213), 1063–1064. https://doi.org/10.1126/science.346.6213.1063

Sedlmair, M. (2026a). Flattened conversations: Reply-structure loss in YouTube comment data. A measurement report on two extraction methods (Preprint v3.8.1). Zenodo. https://doi.org/10.5281/zenodo.22117940

Sedlmair, M. (2026b). Reconstructed time: Comment timestamps in interface-level YouTube extraction. A measurement report (Preprint v1.2). Zenodo. https://doi.org/10.5281/zenodo.22214602. Also deposited at SSRN.

Sedlmair, M. (2026c). Storm names as comparative benchmarks: Lay referential uptake of institutional weather names in German YouTube comments (Preprint v3.4). Zenodo. https://doi.org/10.5281/zenodo.22431589

Sedlmair, M. (2026d). Six methodological pitfalls in YouTube comment extraction: What an interface constraint does to a research corpus (Preprint v2.7.1). Zenodo. https://doi.org/10.5281/zenodo.22836737

Sedlmair, M. (2026e). Two pointers, one recipient: Linguistic orientation where reply edges and @-addresses diverge in YouTube comment trees (Preprint v1.1). Zenodo. https://doi.org/10.5281/zenodo.22254594

Sedlmair, M. (2026f). What the corpus can no longer find: Legacy YouTube comment IDs in a 2017 research deposit no longer resolve, although most of the comments are still there [Preprint]. Zenodo. https://doi.org/10.5281/zenodo.23642703

Sedlmair, M. (2026g). What the corpus still holds: Testing keys, survival and result stability in deposited YouTube comment corpora [Preprint]. Zenodo. https://doi.org/10.5281/zenodo.23805575

Sen, I., Flöck, F., Weller, K., Weiß, B., & Wagner, C. (2021). A total error framework for digital traces of human behavior on online platforms. *Public Opinion Quarterly*, 85(S1), 399–422. https://doi.org/10.1093/poq/nfab018

Tanguy, L., Poudat, C., & Ho-Dac, L.-M. (2024). Talking to oneself in CMC: A study of self replies in Wikipedia talk pages. *Proceedings of the 11th Conference on CMC and Social Media Corpora for the Humanities*. https://doi.org/10.48550/arXiv.2411.19007

Tanskanen, S.-K. (2021). Fragmented but coherent: Lexical cohesion on a YouTube channel. *Discourse, Context & Media*, 44, 100548. https://doi.org/10.1016/j.dcm.2021.100548

Thelwall, M., Sud, P., & Vis, F. (2012). Commenting on YouTube videos: From Guatemalan rock to El Big Bang. *Journal of the American Society for Information Science and Technology*, 63(3), 616–629. https://doi.org/10.1002/asi.21679

Tromble, R. (2021). Where have all the data gone? A critical reflection on academic digital research in the post-API age. *Social Media + Society*, 7(1). https://doi.org/10.1177/2056305121988929

Tufekci, Z. (2014). Big questions for social media big data: Representativeness, validity and other methodological pitfalls. *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, 505–514. https://doi.org/10.1609/icwsm.v8i1.14517

Weber, D., Nasim, M., Mitchell, L., & Falzon, L. (2021). Exploring the effect of streamed social media data variations on social network analysis. *Social Network Analysis and Mining*, 11, 62. https://doi.org/10.1007/s13278-021-00770-y

yt-dlp contributors. (2021). Release 2021.04.11, “Extract comments’ approximate timestamp”. https://github.com/yt-dlp/yt-dlp/releases

yt-dlp contributors. (2025). Issue #12158: Precise comment timestamp. https://github.com/yt-dlp/yt-dlp/issues/12158

yt-dlp contributors. (2025). Issue #14723: Comment extraction inconsistency. https://github.com/yt-dlp/yt-dlp/issues/14723

yt-dlp contributors. (2025). Pull request #15448: [ie/youtube] Fix comment subthreads extraction (commit 76c31a7, 31 December 2025). https://github.com/yt-dlp/yt-dlp/pull/15448

yt-dlp contributors. (2026). `yt_dlp/extractor/youtube/_video.py`, version 2026.08.19. https://github.com/yt-dlp/yt-dlp

Informal web sources mentioned in §2 (a Reddit remark and a Stack Overflow answer) were located with an automated research tool and carry no weight in any claim.

Appendix A — Diagnostics checklist

Run on any comment corpus before analysis. Each is one line or close to it.

#	Check	Signal of trouble	§
1	Distribution of parent values	one distinct value → replies were never collected	8.2
2	Share of replies whose parent is another reply	0 → API-derived, no reply-to-reply structure	4
3	Maximum depth per video; depth distribution	mass piling up at the maximum → cap binding	5
4	Outcome broken down by depth before coding	categorical jump between adjacent levels → structural, not behavioural	9
5	Distinct timestamps vs. corpus size; values per raw label	ratio < 0.01 → reconstructed time; two values per label → clock boundary	6
6	Unresolvable-mention rate by time bucket	trend parallel to the outcome → resolver artefact	7
7	Leading-@ rate by depth	excess at depth 1 → orphans after deletion	8.1
8	Videos attempted vs. videos returning data, from the run log	empty tail or no completion line → blocked run	8.3

#	Check	Signal of trouble	§
9	Per-video and per-corpus totals	round numbers → administrative ceiling	8.4
10	Invisible characters before a leading @; resolutions via prefix fallback, by depth	hidden markers; ambiguous fallbacks to inspect by hand	5.4

Appendix B — Mention resolution rule

0. Remove every character of Unicode category Cf (zero-width and directional marks) from the comment text.
1. Match the leading mention with `^\s*@([\w.\-]+)` (Unicode letters and digits admitted) against the cleaned text.
2. Strip a trailing period.
3. Look the result up against the set of author handles with the leading @ removed.
4. On failure, retry with successively shorter prefixes down to three characters, taking the longest match.
5. If nothing matches, the mention is unresolvable and is excluded from numerator and denominator.

Every mention-based figure in this paper was computed with this rule. Earlier versions of the component reports published a rule without step 0 and with an ASCII-only class in step 1 (§5.4). Step 4 is rarely needed on interface text and affects about one marker in six on API text; an implementation should record when it is used and flag cases in which more than one prefix is a known handle.

Appendix C — Correction register

Corrections made in the three component reports, two related preprints and this paper, in the order of the reports. Each is documented in full in that report's version history.

Report	Version	Correction
Flattened	2.0	Divergent cases no longer described as not "correctly represented" in the API export; no ground truth was established
Flattened	3.0	The API's own `parentId` measured instead of assumed
Flattened	3.1	Recommendation of interface extraction limited to below the newly found cap
Flattened	3.2	Unresolvable mentions had been scored as disagreements; 66.1% corrected to 71.0%; two corpora withdrawn
Flattened	3.3	"Neither method is a superset" withdrawn; the figures showed the API export as a superset
Flattened	3.7	A third reconstruction method recorded as previously unknown
Flattened	3.8	Agreement figures recomputed under the documented rule; withdrawn rows supplied
Pitfalls	2.0	Inconsistent corpus counts (29 vs 33 videos) resolved; "pre-registered" replaced by "specified in advance"
Pitfalls	2.1	Cap no longer attributed to platform storage; attachment rule no longer asserted
Pitfalls	2.2	Reply total 156,730 obtained by subtraction; recounted as 238,700
Pitfalls	2.2	Approximate depth-3 agreement ~59%

Report	Version	Correction
		replaced by measured 54.2%
Pitfalls	2.3	Abstract figure 13,439 of 13,443 corrected to 13,435 of 13,439
Pitfalls	2.3	Description of the cap as a rendering decision removed
Pitfalls	2.4	Four reference entries wrong in title, year, venue or authors; one non-existent reference removed
Pitfalls	2.6	Closest prior statement of the cap (YouTube Data Tools) missed by the 2.5 search; added
Pitfalls	2.7	Statement that the tool gives no depth value corrected; its source gives two
Pitfalls	2.7	Published resolver rule found not to be the one behind the headline figures; all mention figures restated
Pitfalls	2.7	Share of the apparent handle-migration trend that is artefact: “roughly half” corrected to “about a third”
Reconstructed Time	1.1	Framing changed from discovery to measurement after yt-dlp sources were found
Reconstructed Time	1.1	Search for affected work no longer called systematic
Reconstructed Time	1.1	Structural recommendation limited to below the cap
Reconstructed Time	1.2	Missing version-history entry supplied; dependence of marker reconstruction on chronology added
Two Pointers	1.1	Address marker described as the speaker’s signal; it is mostly interface-inserted
Storm Names	2	Replies stated as included were never collected (reported in §8.2)
Earlier project	—	A temporal filter applied to reconstructed timestamps reported as a finding (reported in §6.4)
WtCDNS	1.1	“Maximum depth 3 in every case” corrected to “no comment deeper than depth 3” (§5.1)
WtCDNS	1.1	“Depth parameter unset in every collection” qualified for the sweep runs (§3.3)
WtCDNS	1.1	The two extractions of the main corpus identified; incomplete-delivery warnings disclosed (§12)
WtCDNS	1.2	Herring & Chae (2021) given their full weight as the conceptual precedent; thirteen further works on thread reconstruction, collection-tool variation and YouTube comment data added (§2)
WtCDNS	1.2	“The sweep rules out the extractor” narrowed to “the extractor’s depth setting does not explain the cap” (§5.2)
WtCDNS	1.2	Cap and timestamp behaviour stated as findings about the releases and serving conditions tested (§5.1, §12)
WtCDNS	1.2	Recomputation from the verification package distinguished from independent validation of the original structural assignments (§12, §16)
WtCDNS	1.3	Efstratiou (2025) described as testing the

Report	Version	Correction
		comment endpoints as well as search, not search only (§2); the coverage observation of §4 related to his finding

Changes in version 1.2. Literature added: Aumayr et al. (2011); Bassi et al. (2024, 2025, 2026); Dutta & Khan (2025); Fernández-Martínez (2026); Iakovlev (2026); Ksiazek et al. (2016); Liou (2026); Mints (2026); Tanskanen (2021); Thelwall et al. (2012); Weber et al. (2021). Statements narrowed: the novelty claims in the abstract and §1.2, the parameter sweep, the scope of the cap, and what the verification package shows. Unchanged: every figure and all six findings.

Changes in version 1.3. The description of Efstratiou (2025) in §2 is corrected: besides the search endpoint, his audit tests the comment endpoints and finds them stable, returning almost all comments for a queried video. The coverage paragraph of §4 notes that the one completed comparison here is consistent with that finding. Unchanged: every figure and all six findings.

Changes in version 1.4. Added §11.1 on what the findings mean for large text studies that do not analyse threads, and Appendix D with a fixed protocol for re-checking the premises every six months and a log of each check. The first check (tool source, 24 October 2026) is reported. The first page now states when the premises were last checked. Unchanged: every figure and all six findings.

Changes in version 1.5. The first re-check of the premises (Appendix D, check 1, 24 October 2026) is reported: all premises unchanged under yt-dlp 2026.10.21. The first page, the abstract and §12 state the result. The check script now shows progress while it runs; its measurements are unchanged. The scripts and the result file are deposited with the verification package. The abstract states the use for linguistic work. Unchanged: every figure and all six findings.

Changes in version 1.6. Literature added after a further external literature check (Consensus, 29 October 2026): Sen et al. (2021) and Olteanu et al. (2019) on error frameworks for digital trace data and Davidson et al. (2023) on platform-controlled access, with one sentence placing the six findings in that framework (§2, platform artefacts); Campos-Moinier (2026), a tutorial and tool for API-based collection, with one sentence relating Appendix A to such tools (§2, research software). §14 records the check. This is not a correction and is not counted in the register above. Unchanged: every figure and all six findings.

Changes in version 1.7. The first page gives a contact address and links to the two companion papers, *What the Corpus Can No Longer Find* (Sedlmair, 2026f) and *What the Corpus Still Holds* (Sedlmair, 2026g), which are also introduced in the opening section and listed in the references. §15 adds a note from the author. This is not a correction. Unchanged: every figure and all six findings.

Appendix D — Keeping the premises current

The cap, the timestamp behaviour and the API's flat model are properties of a tool release and of the way the platform served data at the time of collection. Both can change without notice; a new yt-dlp release (2026.10.21) appeared within two months of the original collections. The premises are therefore re-checked every six months under the protocol below — the first check on 24 October 2026, then in May and November — which was fixed on 24 October 2026 and is not changed after the fact. Each result is published as a new version of this paper, and the first page states when the premises were last checked. If a check is missed, that date shows it.

Part A — tool source. For the current yt-dlp release, the source of the five functions that assign parent, depth and timestamp (`_comment_entries`, `_get_comments`, `_extract_comment` and `_extract_comment_old` in `extractor/youtube/_video.py`; `_parse_time_text` in `extractor/youtube/_base.py`) is fingerprinted with

SHA-256 and compared with release 2026.08.19. The check also confirms that `max_comments` still reads five positions that default to unbounded, and that the source still annotates the subthread structure as four levels deep.

Part B — platform behaviour. With the current release, the three sweep videos of §5.2 are re-collected at `max_depth` 1, 2 and 9, and a fixed panel of five videos used in this paper (70vYj1KPyT4, JDy95_eNPzM, pxEWXGSIpAI, UPrkC1LdlLY, WV29R1M25n8) is re-collected with a ceiling of 20,000 comments each. On every file the checks of Appendix A are run, and four quantities are recorded: the maximum returned depth; marker–parent divergence at depths 2 and 3, pooled over the panel under the rule of Appendix B; and the ratio of distinct timestamps to comments.

Part C — API (where a key is available). For 70vYj1KPyT4, up to 500 threads with replies are retrieved through `commentThreads.list` and `comments.list`, and it is recorded whether any returned `parentId` points to something other than a top-level comment.

Decision lines. A premise counts as unchanged if its value falls on the same side of a fixed line as in the original collections: no depth above 3; divergence below 1% at depth 2 and above 25% at depth 3; fewer than one distinct timestamp per hundred comments in every panel file with at least 1,000 comments; no API parent below the top level. A value that crosses its line is reported as a change, the affected sections are marked on the first page, and the text is revised. A quantity that a run could not measure is reported as not tested, never as unchanged. A video that fails to download is listed, not dropped.

Implementation. The protocol is implemented in `premises_check.py`, deposited with the verification package together with the result file of each check (§16). It uses the Python standard library and yt-dlp, keeps raw files locally and writes a summary of counts only. It shows a running progress line during each collection. Its measurement code was tested on synthetic corpora with known depth and divergence, including one that exceeds the cap, and its source check on both releases and on a deliberately altered copy, which it reported as changed.

Result of check 1 (24 October 2026). Every premise is unchanged, and no download failed. *Tool:* the five functions are identical to release 2026.08.19. *Cap:* no file of the fourteen re-collections returned a reply below depth 3, and the sweep reproduced the table of §5.2 exactly: at `max_depth` 1, 2 and 9 the three videos returned depths 0, 1 and 3. Pooled over the panel (93,609 comments, 40,443 replies), the depth distribution is 53,166 / 20,630 / 7,689 / 12,124 at depths 0 to 3, and in three of the five panel videos the deepest level is larger than the one above it. *Divergence:* 2 of 7,085 resolved markers at depth 2 (0.03%) against 5,054 of 10,891 at depth 3 (46.4%). *Time:* between 2 and 38 distinct timestamps per panel file (ratio 0.0001 to 0.0019), and no label carried more than one value. *API:* across 500 threads of 70vYj1KPyT4, none of 1,231 replies had a parent other than a top-level comment. Four of the five panel videos reached the protocol's ceiling of 20,000 comments, as intended; the figures describe those first 20,000, not the full comment sections.

No.	Date	yt-dlp	A — tool source	B — platform behaviour	C — API
0	Aug–Sep 2026, original collections	2026.08.19	reference	no depth above 3; divergence 0.0002 at depth 2 and 0.458 at depth 3 (main corpus); timestamps reconstructed	flat: 43,896 of 43,896 replies under top-level comments
1	24 Oct 2026	2026.10.21	all five functions identical to 2026.08.19; five-position `max_comments`, default unbounded; four-level annotation present. The release	unchanged: no depth above 3; sweep 0 / 1 / 3; divergence 2 of 7,085 at depth 2 and 0.464 at depth 3 (panel); 2–38 distinct timestamps	unchanged: 0 of 1,231 replies below the top level

No.	Date	yt-dlp	A — tool source	B — platform behaviour	C — API
			changes default player clients, which these functions do not reference.	per file	
2	due May 2027	current			