Unlearning in RAG Systems

Ashmit Janjirkar GroupBrain Talks

25 October 2026

Evaluating retrieval-side unlearning: index fidelity and residual answerability

Why would AI need to forget?

Privacy

People can ask for their personal data to be erased, e.g. under the GDPR's "right to be forgotten".

Ownership

Licensed or copyrighted documents may have to be withdrawn when access ends.

Correctness

Outdated, wrong or harmful records should stop shaping answers.

Machine unlearning

Removing the effect of specific data from a trained system, ideally without rebuilding it from scratch.

How a RAG system answers

01

Question

"Who led Fascist Italy?"

02

Retriever

Embeds the question as a vector

03

Index

Finds the k nearest stored records

04

LLM

Reads question + records

05

Answer

"Benito Mussolini."

ℹ️ Knowledge sits in an editable store, so deletion looks easy: remove the record. This talk asks what that actually achieves.

Knowledge lives in two places

The index

- Documents added at deployment
- Can be edited record by record
- Deleting a record is cheap

The model's weights

- Everything learned in pre-training
- Fixed once the model is trained
- Changing it means retraining or fine-tuning

⚠️ Deleting from the index never touches the weights. If the model already knows a fact, it can still answer.

Three meanings of "unlearning"

Exact

- System indistinguishable from one rebuilt without the data
- Strongest guarantee
- Costly unless the method is designed for it

Approximate

- Edit the trained model to end up close to the rebuilt one
- e.g. fine-tuning against the data
- Cheaper; hard to verify how close

Behavioural

- Make the system stop producing certain outputs
- e.g. refusing or steering answers
- Works at the interface; says nothing about stored state



Exactness is defined relative to a reference: the system you would have built without the data.

A short history of machine unlearning

2015 — **Cao & Yang**

First 'machine unlearning': summation-form learners, so a point's contribution can be subtracted

2019 — **Ginart et al.**

Deletion defined relative to the learning algorithm; deletion-efficient (quantised) k-means

2020–21 — **Guo et al. · Golatkar et al. · SISA**

Certified removal (Newton step + noise); selective forgetting in deep nets; exact unlearning by sharding

2023–24 — **LLM unlearning**

Gradient ascent (Jang et al.), 'Who's Harry Potter?' (Eldan & Russinovich), NPO (Zhang et al.), TOFU benchmark (Maini et al.)

2024–26 — **Retrieval & in-context**

RAG-based unlearning (Wang et al.); exact ICL unlearning (Muresanu et al.); retrieval vs parametric leakage (Raeesi et al.); this work

Formal definitions

Exact unlearning

- $U(A(D), D, F) =^{d} A(D \setminus F)$
- Equality in distribution: the output is indistinguishable from retraining on $D \setminus F$
- Achieved by design: SISA, quantised k-means (Ginart et al. 2019; Bourtoule et al. 2021)

(ε, δ)-approximate unlearning

- For every set of models S:
- $P[U(A(D), D, F) \in S] \leq e^{\varepsilon} \cdot P[A(D \setminus F) \in S] + \delta$ (and symmetrically)
- Same shape as differential privacy; certified removal (Guo et al. 2020) adds calibrated noise to reach it

ℹ️ Both definitions compare against a reference, $A(D \setminus F)$. A low answer rate on F is a behavioural check, not either of these.

SISA: retrain only the part that saw the data

01

Sharded

Split D into n disjoint shards $D_1 \dots D_n$, one model each

02

Isolated

A point $z \in D_i$ only influences model θ_i

03

Sliced

Each shard trained in S slices, checkpoint after each

04

Aggregated

Prediction = majority vote over $\theta_1 \dots \theta_n$

✅ Delete z: retrain one shard from z's slice. Expected cost ≈ $(|D|/n)\cdot(S+1)/(2S)$ instead of $|D|$, and the result is exact (Bourtoule et al., 2021).

Deletion-efficient clustering (Ginart et al., 2019)

Quantised k-means

- Centroid of a cluster with m points: $c = (1/m)\ \Sigma\ x_i$
- Snap to a lattice of spacing ε with random phase θ: $Q(c) = \varepsilon \cdot \mathrm{round}((c - \theta)/\varepsilon) + \theta$
- Deleting x moves c by $\Delta c = (c - x)/(m - 1)$: small when m is large
- If Q(c) is unchanged, the index is already in the right state: deletion is O(1) work

How often does it fail?

- Usual statement: $P(Q(c) \text{ changes}) = O(1/(\varepsilon m))$
- But all d coordinates must stay in their cells. For unit vectors in d dims:
- $P(\text{change}) \approx 1 - (1 - \sqrt{(2/(\pi d))} / (\varepsilon m))^d$
- At d = 384 this is ~14× higher than the naive bound; E3 measures 12.8% vs 14.4% predicted

Approximate unlearning: editing the weights

Certified removal

Guo et al. 2020. One Newton step for convex losses: $\theta^- = \theta^* + H^{-1}\nabla\ell(z; \theta^*)$, plus training noise to hide the residual.

Gradient ascent

Jang et al. 2023. Raise the loss on the forget set D_f: $\theta \leftarrow \theta + \eta\nabla_\theta L(\theta; D_f)$. Utility collapses if run too long.

Gradient difference

Keep the retain set D_r: $\min_\theta L(\theta; D_r) - \lambda L(\theta; D_f)$. A standard baseline on TOFU.

NPO

Zhang et al. 2024. Bounded loss: $(2/\beta) \cdot E_{D_f} \log(1 + (\pi_\theta/\pi_{ref})^\beta)$. More stable than gradient ascent.

ℹ️ All four change θ. In our RAG setting θ never changes, so none of them can be what removes an answer.

Unlearning via retrieval and context

Wang et al. (2024)

RAG-based unlearning: add constructed knowledge to the knowledge base that steers the model away from the target answer. No weight changes.

Muresanu et al. (2025)

Exact removal of in-context examples with a deletion-friendly selection procedure, counting the extra cost paid at every query.

Dynamic indices

Systems like FreshDiskANN (Singh et al., 2021) support live inserts and deletes while keeping search quality high.

ℹ️ Judge these methods through the interface they control, not by whether the bare model can still answer elsewhere.

> *A common shortcut: delete the record, ask the question again, and call it success if the answer disappears.*

— The gap

That single answer rate mixes up what changed in the index, what the prompt told the model, and what the model already knew.

One request, three questions

1 · Index fidelity

- Is the edited index the same as a rebuilt one?
- Compare retrieved record IDs after deletion vs. rebuild
- Measured in E3

2 · Behaviour

- Does the system stop giving the answer?
- Depends on the prompt and what users may ask; check retained questions too
- Measured in E1, E2, E3

3 · Training

- Is the system like one never trained on the data?
- Needs a retrained model or a proof; index deletion can't reach it
- Not measured

⚠ Passing one test says nothing about the other two.

Two reference systems

$R_idx = (\theta , B(C \setminus F))$

- Same model, index rebuilt without F
- Prop. 1: if the deleted index has the same distribution as $B(C \setminus F)$, the pipeline's output law equals R_idx on every allowed request x
- Holds even if $F \cap D \neq \varnothing$: the model may still answer from memory

$R_train = (A(D \setminus F) , B(C \setminus F))$

- Model also retrained without F
- Prop. 2: if a request x_0 bypasses retrieval and $TV(P_\theta(\cdot|x_0), P_A(D \setminus F)(\cdot|x_0)) = \delta > 0$
- then any index-only method is at least δ away from R_train



A residual answer is not proof that deletion failed, and a refusal is not proof that the data is gone. We never estimate δ: there is no retrained model.

Data, models and metrics

Part	What we used
Data	50 Wikipedia articles → 2,506 QA pairs (Llama-3.3-70B); 25 established · 25 recent topics
Models	Llama-3.1-8B-Instruct, 4-bit NF4, greedy; all-MiniLM-L6-v2 embeddings (d = 384)
Answer match	$h(y, a) = 1$ if all words of a appear in y ($\|a\| \leq 2$ words), else if a is a substring of y
Retrieval overlap	Jaccard $J(S_B, S_C) = \|S_B \cap S_C\| / \|S_B \cup S_C\|$ over the top-2 record IDs
Divergence	First-token $KL(p_B(\cdot\|q) \,\|\, p_C(\cdot\|q))$, before generated histories differ
Uncertainty	5 seeds · 250 trials; paired contrasts; 95% CIs from 5,000 topic-level bootstrap resamples

⚠ "Recent" is a proxy, not proof the model never saw it: some recent topics have older history.

Three experiments

E1 · Answers without retrieval | E2 · Deletion vs. prompt |
E3 · Deletion vs. rebuild

E1: the model answers many questions on its own

30.7%

Established topics answered with no retrieval

- 369 / 1,206 · 95% CI 26.5–34.8%

6.4%

Recent topics answered with no retrieval

- 76 / 1,177 · 95% CI 4.3–8.8%

24.2 pts

Gap between groups

- 95% CI 19.5–28.9

ℹ️ No retrieval, so deleting a record cannot change these answers. Per topic: established 8.0% (Industrial Revolution) to 52.4% (Sahara Desert); recent 0% to 22.0%. Not evidence of training membership.

E2: crossing index state with prompt

Index states (each under both prompts)

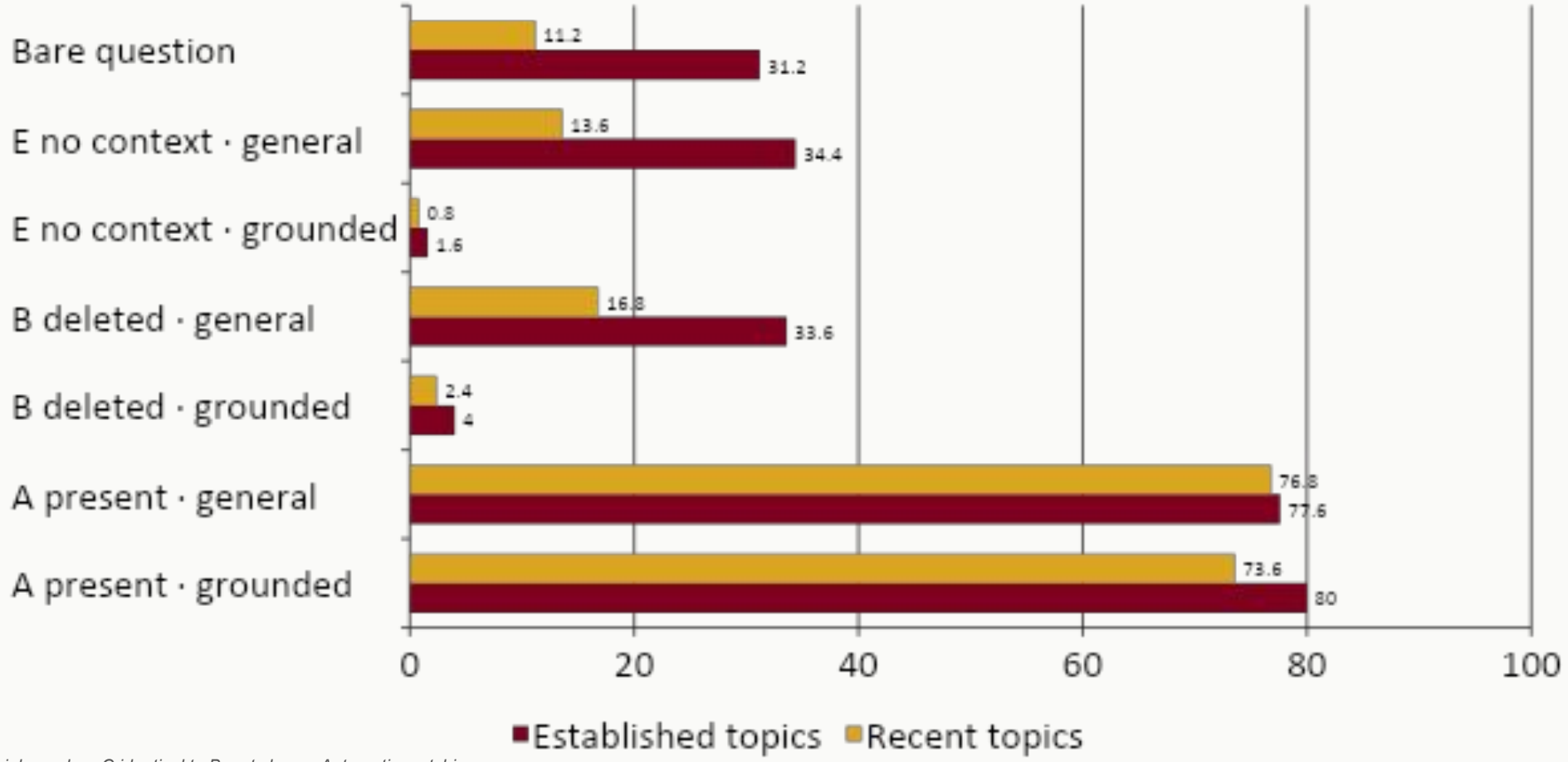
- A: target present
- B: target deleted
- C: index rebuilt without target
- E: no context
- Bare: question only, no template

Prompts

- Grounded: "Answer using only the context provided. If the answer is not in the context, say 'I don't know'."
- General: "Here is some context that may or may not be relevant. Answer the question."

ℹ️ Each index: target + 10 same-topic records + 390 others = 401 records, exact search, top 2. Plus a retained-question check. 250 trials, 3,750 answers.

E2: answer-match rate by condition (%)



Source: 125 trials per bar; C identical to B, not shown. Automatic matching.

E2: deletion and the prompt both matter

+29.6

Δ prompt, deleted context, established

+14.4

Δ prompt, deleted context, recent

≈ 0

h(B, general) − h(E, general)

⚠ Deletion effect h(A) − h(B) under the general prompt is still 44 / 60 points, and retained questions are unchanged in all 250 trials. A low rate under a strict prompt mixes the deletion and the instruction.

E2: where do leftover answers come from?

Source of match	Established	Recent
Model's own knowledge	32	8
Kept context records	7	6
False positive	0	7
Ambiguous	3	0
Knowledge-only rate	**25.6%**	**6.4%**

⚠ All 8 'recent' knowledge answers are older facts (0 of 125 used a post-cutoff fact). Labels are AI-assisted and still need independent human review.

E3: fast local deletion vs. a full rebuild

01

A · start

401 records, k-means with k = 2, centres snapped to a grid (spacing 0.5)

02

B · delete

Mark the record inactive, recompute only its cluster's centre, snap it again

03

C · rebuild

Cluster the 400 kept records from scratch, same random seed

04

C-alt

Same rebuild, different seed: the noise floor between two valid rebuilds

ℹ️ Search: nearest cluster centre, return the 2 closest records; also compared with exact search. Answers generated only after all index checks pass.

E3: deleted and rebuilt indices often disagree

36%

Trials with J(S_B, S_C) < 1

- 90 of 250

0.70

Mean J(S_B, S_C)

- vs 0.57 for J(S_Calt, S_C)

12.8%

Deletions where Q(c) changed

- Predicted 14.4% (dimension-aware)

ℹ️ Of the 90 disagreements, B is closer to exact search in 34, C in 32, with 24 ties: B is not identical to the rebuild, but not worse at search either.

E3: same answer rates, different indices

Condition	Established	Recent
A · target present	65.6%	56.0%
B · local delete	5.6%	1.6%
C · rebuild	4.0%	1.6%
Question only	28.8%	10.4%

⚠ B and C look interchangeable by answer rate, yet retrieve different records in 36% of trials. Local delete: 0.197 ms vs 8.035 ms rebuild (median, 401 records).

Results at a glance: E1, E2 and E3

Experiment	Measure	Established	Recent	What it shows
E1 No retrieval (2,383 Qs)	Answer match, bare question	**30.7% [26.5, 34.8]**	**6.4% [4.3, 8.8]**	Model answers on its own; index deletion cannot touch these
E2 Flat index, 4 states × 2 prompts (250 trials, 3,750 answers)	A · target present, grounded	80.0%	73.6%	Target retrieved in 99.2% of trials
	B · deleted, grounded (abstain)	4.0% (83.2%)	2.4% (95.2%)	Strict prompt + deletion looks like forgetting
	B · deleted, general	**33.6%**	**16.8%**	Same context, permissive prompt: answers return
	E · no context, general	34.4%	13.6%	Deleted ≈ no retrieval (Δ −0.8 / +3.2, n.s.)
	Prompt effect Δ = B general − B grounded	**+29.6 [22.4, 37.5]**	**+14.4 [6.8, 21.6]**	Instruction removes most of what the model knows
	Knowledge-only residual (hit review)	25.6%	6.4%	0 of 125 recent answers used a post-cutoff fact
E3 Quantised k-means, local delete vs rebuild (250 trials)	Answer match A → B → C, grounded	65.6 → 5.6 → 4.0%	56.0 → 1.6 → 1.6%	Deleted ≈ rebuilt by answer rate
	…when target was retrieved (187 / 250)	85.1 → 5.3 → 3.2%	75.3 → 2.2 → 2.2%	Deletion removes record-supplied answers
	Retrieved sets differ (B vs C)	**36% of trials**	**J(B,C) 0.70 vs J(C_alt,C) 0.57**	Not identical to rebuild; within rebuild-to-rebuild spread
	Median update time	0.197 ms delete	8.035 ms rebuild	≈41× faster at 401 records (cached embeddings)

Brackets: 95% topic-bootstrap CIs (5,000 resamples). Abstain = "I don't know"-style refusals. E2 used a 64-token limit; E3 answers use the grounded prompt (50 tokens). Hit-review labels are AI-assisted and pending human review. Llama-3.1-8B-Instruct throughout.

Three lessons

01

Record ≠ fact

Deleting a record is not forgetting a fact: the model answers ~31% of established questions with no retrieval (E1).

02

Prompt matters

A suppression rate also measures the prompt: 4.0% strict vs 33.6% general on the same deleted context (E2).

03

Rate ≠ index

Equal answer rates don't mean an equal index: 5.6% vs 4.0%, yet different records in 36% of trials (E3).

Questions?

Judge retrieval-side unlearning against a declared reference and a declared interface, and measure the index and the answers separately.