

000 UN-ATTRIBUTABILITY: COMPUTING GENERATION 001 NOVELTY FROM RETRIEVAL & SEMANTIC SIMILAR- 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 UN-ATTRIBUTABILITY: COMPUTING GENERATION NOVELTY FROM RETRIEVAL & SEMANTIC SIMILAR- ITY

Anonymous authors

Paper under double-blind review

ABSTRACT

Understanding how language-model outputs relate to the pretraining corpus is central to studying model behavior. Most training-data attribution (TDA) methods ask which training examples causally influence a given output, often using leave-one-out tests. We invert the question: which outputs *cannot* be attributed to any pretraining example? We introduce *un-attributability* as an operational measure of semantic novelty: an output is *novel* if the pretraining corpus contains no semantically similar context. We approximate this with a simple two-stage retrieval pipeline: index the corpus with lightweight GIST embeddings, retrieve the top- n candidates, then rerank with ColBERTv2. **The less attributable a text is, relative to a human baseline, the more novel it is considered to be.** We evaluate on SmoILM and SmoILM2 and report three findings: (1) models draw on pretraining data across much longer spans than previously reported; (2) some domains systematically promote or suppress novelty; and (3) instruction tuning not only alters style but also increases novelty. Reframing novelty assessment around *un-attributability* enables efficient analysis at pretraining scale. We release code and \sim 20 TB of embeddings and index artifacts to support replication and large-scale extension.

1 INTRODUCTION

Large language models (LLMs) now power chatbots, copilots, and autonomous agents. Understanding how language model outputs relate to their pretraining corpora is central to studying model behavior and generalization. A key question is whether an output is *novel* – not traceable to memorizing training data. Measuring novelty reveals when models generalize beyond what they have seen, signals compositional generalization, assessing the *true zero-shot* behavior – informing debates about provenance and intellectual property. Thus, computing novelty is both a technical problem and a prerequisite for interpreting what LLMs actually learn.

Training data attribution (TDA) methods address related questions by tracing model behavior back to specific data (Hammoudeh & Lowd, 2024; Deng et al., 2025). Two approaches dominate (Chang et al., 2025): *causal influence* methods measure a training sample’s leave-one-out effect (Koh & Liang, 2017) but do not **easily** scale to trillion-token corpora; *factual attribution* methods scale **better** but rely on lexical matches between outputs and training text (Liu et al., 2025b;a; Wang et al., 2025c), not being robust to many simple variations. These approaches have taught us much about what can be *definitively* attributed. Yet many consequential questions hinge on the opposite: what *cannot* be attributed.

We address this gap by inverting the usual TDA question and focusing on *un-attributability*. Rather than asking which training samples **are related to** an output, we ask which outputs *cannot* be attributed to the pretraining corpus. We treat such outputs as *novel*. Our test is strict: the pretraining data must contain neither lexical overlaps nor semantically similar contexts with the generation (Figure 1).

We operationalize this test using best-match semantic similarity between a generation and the full pretraining corpus, *i.e.* **by factually attributing the generation**. Concretely, we compute GIST embeddings (Solatorio, 2024) over corpus chunks, build a vector index, retrieve the top- n candidates,

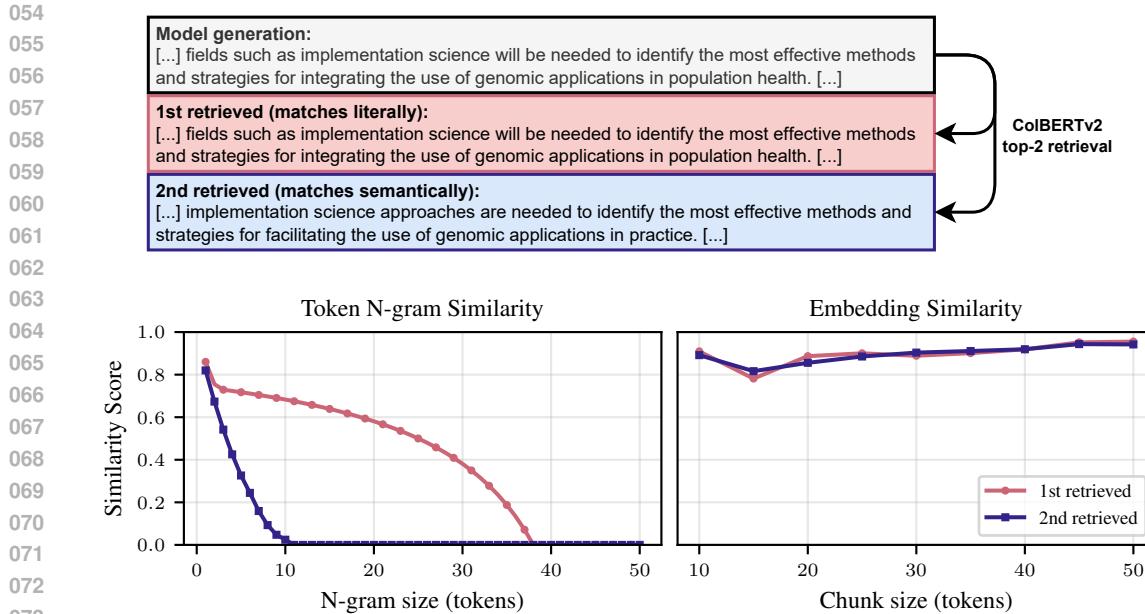


Figure 1: **Embedding similarity is more robust to long or paraphrased texts than N-gram similarity.** Comparison of similarity measured with N-gram overlap (left) and embedding cosine similarity (right) with increasing sequence length. The similarity is measured between a model generation and its top 2 closest semantic matches in the pretraining corpus retrieved using our test. Both training excerpts convey the same information as the generation, but lexical overlap fails to recognize this with larger N-grams, whereas embeddings remain robust.

and re-rank them with ColBERTv2 (Santhanam et al., 2022). To calibrate attribution scores, we also evaluate held-out human-written references that are guaranteed not to appear in the pretraining data. **Measuring relative scores to the baseline makes our novelty measure interpretable and regularizes the novelty of generations by comparing them to known-novel text of the same domain and length.** This procedure offers greater coverage and auditability than lexical matching, is model- and task-agnostic, and is lightweight enough to run on full pretraining corpora. It thus offers a scalable way to collect evidence of what is *not* attributable.

We apply the test to SmoLLM (Allal et al., 2024) and SmoLLM2 (Allal et al., 2025), two LLMs with open pretraining corpora. Our analysis reveals surprising patterns missed by previous lexical methods (McCoy et al., 2023; Merrill et al., 2024). First, both models draw on pretraining data over much longer sequences than previously reported. Second, novelty varies systematically by task domain. Third, embedding-based novelty estimates are stable under style shifts from instruction tuning; after accounting for these shifts, instruction tuning substantially increases novelty. These results suggest that instruction tuning shapes not only style but also compositional generation behavior.

Our contributions are conceptual and practical. Conceptually, we invert the goal of TDA from what is attributable to what is definitively *not* attributable. Practically, we present a lightweight test for model novelty based on *un*-attributability and conduct a large-scale study on SmoLLM and SmoLLM2. We will release the code and ~ 20 TB of artifacts upon publication, including embeddings and index files, to support replication and extension.

2 RELATED WORK

We situate our work in relation to fields studying the capabilities of a model in relation to the training data, namely, training data attribution (TDA), causal and factual, as well as memorization research. **We adopt the terminology from (Chang et al., 2025) to distinguish between causal influence and factual attribution.**

108 **Causal influence.** Causal influence attributes model behavior to a training sample by the treatment
 109 effect that including the sample in the training dataset has on the observed behavior, also known as
 110 the leave-one-out (LOO) effect. Since LOO is infeasible to compute explicitly across datasets, the
 111 field of TDA studies approximations of LOO, where influence functions (Hampel, 1974) (IF) are
 112 most prominent. Method work studies how to adapt IFs to deep models, where convexity assump-
 113 tions break, computational and memory costs explode, resulting in a variety of IF adaptations (Koh
 114 & Liang, 2017; Schioppa et al., 2022; Chang et al., 2025; Grosse et al., 2023; Park et al., 2023;
 115 Pruthi et al., 2020; Wang et al., 2025a; Choe et al., 2024; Xia et al., 2024). However, IFs are known
 116 to be fragile when assumptions are not met (Bae et al., 2022; Basu et al., 2021; Epifano et al., 2023;
 117 Nguyen et al., 2023). Another line of work estimates LOO through unrolled differentiation through
 118 the training process, bypassing model convergence assumptions and considering different optimizer
 119 algorithms (Hara et al., 2019; Bae et al., 2024; Wang et al., 2025b; Ilyas & Engstrom, 2025). While
 120 both approaches approximate LOO in deep models like LLMs, their memory and computational
 121 demands limit scalability to large pretraining corpora, requiring workarounds. For instance, Guo
 122 et al. (2021) reduce costs by retrieving candidate samples via k -nearest neighbors in embedding
 123 space, while Grosse et al. (2023) rely on TF-IDF filtering to preselect potentially influential sam-
 124 ples. These approaches assume that influential samples must lie close in some representation space,
 125 but this is not guaranteed (Hu et al., 2024) when not explicitly trained for it, like in Yeh et al. (2018);
 126 Sun et al. (2025). We invert this logic: when the nearest samples in representation space are dissim-
 127 ilar from the generation, we interpret this un-attributability as novelty – evidence that the model’s
 128 output is composed rather than reused.

129 **Factual attribution.** Factual attribution attributes LLM outputs back to the training data to iden-
 130 tify what grounds the output (Chang et al., 2025). For this task, retrieval methods relying on lexical
 131 overlap like BM25 and n -gram overlap (Liu et al., 2025a; Merrill et al., 2024; McCoy et al., 2023;
 132 Liu et al., 2025b; Gottesman et al., 2025; Peng et al., 2023; Wang et al., 2025c) offer a strong baseline
 133 and are often better suited than causal TDA (Akyürek et al., 2022; Chang et al., 2025). Additionally,
 134 lexical overlap-based retrieval methods are considerably more lightweight than causal TDA, allow-
 135 ing them to scale efficiently to large datasets such as pretraining corpora. However, lexical overlap
 136 is an overly strict criterion, as it underestimates the possibility that a model learns facts from para-
 137 phrased documents. For factual attribution, or in our case, evaluating novelty by un-attributability,
 138 it is more appropriate to consider a semantic representation of the generation and measure semantic
 139 similarity.

140
 141 **Memorization and membership inference.** Closely related are studies of memorization and
 142 membership inference. Memorization work (Wu et al., 2025; Feldman & Zhang, 2020) typically
 143 investigates whether specific samples can be elicited verbatim from a model, whereas our notion of
 144 novelty captures whether the underlying *information* is present in the corpus, even if paraphrased.
 145 Membership inference attacks (MIA) (Puerto et al., 2025; Mesana et al., 2025; Zhang et al., 2025;
 146 2024) instead ask whether a particular example was part of pretraining, often in adversarial settings.
 147 While informative for data privacy, MIAs do not address the broader question of how models gen-
 148 erate text not attributable to their training data. Our novelty measure, therefore, complements both
 149 attribution and memorization/MIA, providing a new perspective on generalization.

3 DEFINING NOVELTY AS WHAT IS *Un*-ATTRIBUTABLE

150
 151
 152 We propose a retrieval-based test for semantic novelty in LLM outputs, analyzing generalization
 153 with respect to the training data. The central challenge is the scale of pretraining corpora. To
 154 make the problem tractable, we focus on *un-attribution*: showing that no close semantic match to
 155 the output exists in the corpus. This reduces the task to large-scale retrieval. If no close match is
 156 found, we deem the output novel rather than attributable to the corpus. We use a two-stage pipeline
 157 (Figure 2). Stage 1 performs initial retrieval over GIST embeddings with FAISS (Douze et al.,
 158 2024); Stage 2 reranks candidates with ColBERTv2 (Santhanam et al., 2022). Algorithm 1 details
 159 the procedure, and we will release our code upon publication.

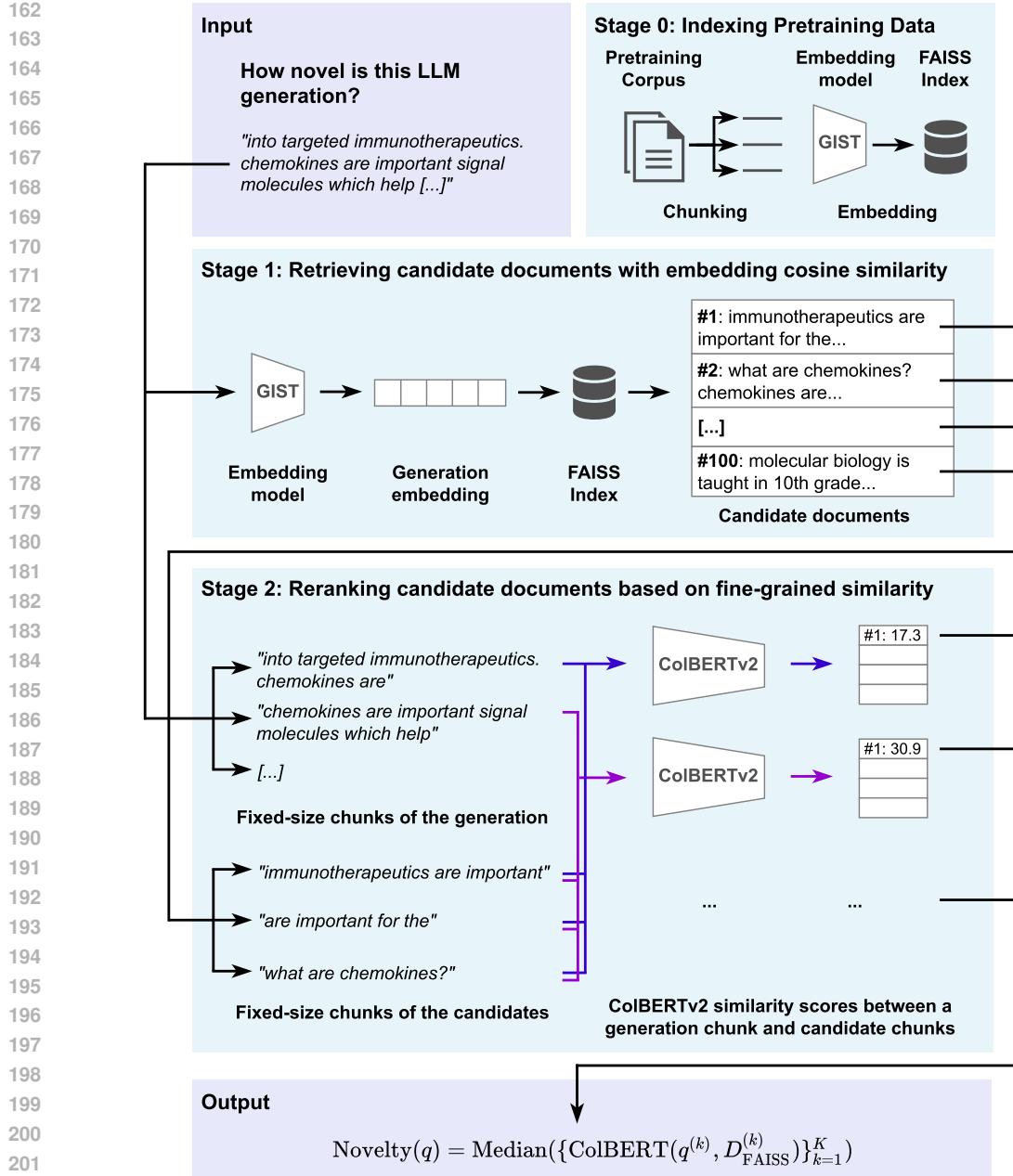


Figure 2: **Pipeline for scoring the novelty of an LLM output q .** We test whether q is *unattributable* to the pretraining corpus – our operational definition of novelty. *Stage 0 (one-time):* Chunk the corpus, compute L2-normalized GIST (Solatorio, 2024) embeddings, and build a cosine-similarity FAISS (Douze et al., 2024) index. *Stage 1:* Embed q with GIST and retrieve the top- n nearest corpus chunks. *Stage 2:* Rerank retrieved candidates with ColBERTv2 (Santhanam et al., 2022) at multiple chunk sizes. The novelty score is the median, over q ’s chunks, of the ColBERTv2 similarity to the best retrieved chunk, normalized by the sequence length and corresponding baseline score.

3.1 STAGE 1: BUILDING THE VECTOR INDEX AND INITIAL RETRIEVAL

We first build a cosine-similarity-based FAISS index (Douze et al., 2024) I_{FAISS} to enable efficient search over the entire pretraining corpus. Because FAISS operates on vectors, we embed corpus chunks with the GIST model (Solatorio, 2024). GIST embeddings are compact, reducing storage costs; yet expressive, as evidenced by strong MTEB performance (Muennighoff et al., 2023;

216 **Algorithm 1** Novelty test with retrieval, reranking, and baseline-normalized scoring

217

218 **Require:** I_{FAISS} (FAISS index over L2-normalized GIST embeddings), q (LLM output), b (baseline

219 text), ϕ_{GIST} (GIST embedder), ColBERT (ColBERTv2 reranker), $n > 0$ (initial retrieval size),

220 $k > 0$ (chunk size in tokens)

221 **Stage 1: Initial retrieval (cosine similarity via FAISS)**

222 $D_q \leftarrow \text{kNN}(I_{\text{FAISS}}, \phi_{\text{GIST}}(q), n)$

223 $D_b \leftarrow \text{kNN}(I_{\text{FAISS}}, \phi_{\text{GIST}}(b), n)$

224 **Stage 2: Reranking and normalized scoring**

225 $Q^{(k)} \leftarrow \text{chunk}(q, k); B^{(k)} \leftarrow \text{chunk}(b, k)$

226 $C_q^{(k)} \leftarrow \bigcup_{d \in D_q} \text{chunk}(d, k); C_b^{(k)} \leftarrow \bigcup_{d \in D_b} \text{chunk}(d, k)$

227 *Define length-normalized best-match score for chunk x :*

228
$$\tilde{s}(x, C) = \frac{\max_{c \in C} \text{ColBERT}(x, c)}{|x|} \quad (\|x\| = \# \text{ query tokens used by ColBERT})$$

229 *Compute baseline normalizer (robust central tendency):*

230 $\mu_B^{(k)} \leftarrow \text{mean}(\{\tilde{s}(b^{(k)}, C_b^{(k)}) : b^{(k)} \in B^{(k)}\})$

231 *Compute per-chunk similarity ratios for q :*

232 $R^{(k)} \leftarrow [\tilde{s}(q^{(k)}, C_q^{(k)}) / \mu_B^{(k)} - 1 : q^{(k)} \in Q^{(k)}]$

233 **Output: summary novelty score and diagnostics**

234 $N^{(k)} \leftarrow \text{median}(R^{(k)}) \quad \triangleright \text{Novelty/attributability ratio for } k; < 0 = \text{more novel than baseline}$

235 **return** $N^{(k)}, R^{(k)}$

238 Enevoldsen et al., 2025), allowing accurate attribution at pretraining scale. We chunk text at the

239 GIST model’s maximum length (512 tokens) and L2-normalize all embeddings so inner products

240 equal cosine similarity. [We provide an analysis of the effect of chunking borders on our retrieval](#)

241 [pipeline in Appendix A](#). At query time, we embed and L2-normalize the LLM output q , then retrieve

242 the top- $n = 100$ nearest corpus chunks, balancing recall and compute. Please refer to Appendix A

243 for additional analysis about this choice.

244

245 3.2 STAGE 2: RE-RANKING USING COLBERTV2 FOR FINE-GRAINED ANALYSIS

246 We then rerank the top 100 candidates with finer-grained similarity via ColBERTv2 (Santhanam

247 et al., 2022). We chunk both the LLM output and the retrieved candidates at one or more sizes and

248 rerank with ColBERTv2, which operates on context-dependent token embeddings and is widely used

249 for reranking (Zhao et al., 2024). Varying chunk size lets us examine how length affects novelty.

250 While ColBERTv2 scores are accurate, they have two drawbacks: (1) without a reference, they are

251 hard to interpret; and (2) because they sum over query tokens, they are length-biased. We address

252 both by (a) normalizing scores by query-token count and (b) comparing each output to a human-

253 written domain-matched baseline b , if available. The resulting similarity ratio is interpretable: values

254 < 0 indicate q is more novel than the baseline; values > 0 indicate greater attributable overlap. This

255 mechanism allows us to compare model novelty relative to human-written text.

256

257 4 EXPERIMENTS

258

259 We measure novelty as whether the model outputs are not attributable to the pretraining corpus, i.e.

260 are less similar compared to a human baseline and report our experimental setup and results.

261

262 4.1 EXPERIMENTAL SETUP

263

264 Because the test requires full access to the pretraining corpus, we limit our analysis to the publicly

265 available SmoLLM (Allal et al., 2024) and SmoILM2 (Allal et al., 2025) model families. We eval-

266 uate novelty as a function of output length, varying the chunk size $k \in \{50, 100, 150, \dots, 500\}$

267 tokens. We consider two settings: open-domain and domain-specific generation, to assess the effect

268 of domain on novelty. For the open-domain setting, we use a subset of Dolma (Soldaini et al., 2024)

269 as a human baseline, following Merrill et al. (2024). For the domain-specific setting, we analyze

benchmark generations and compare them to the benchmark targets.

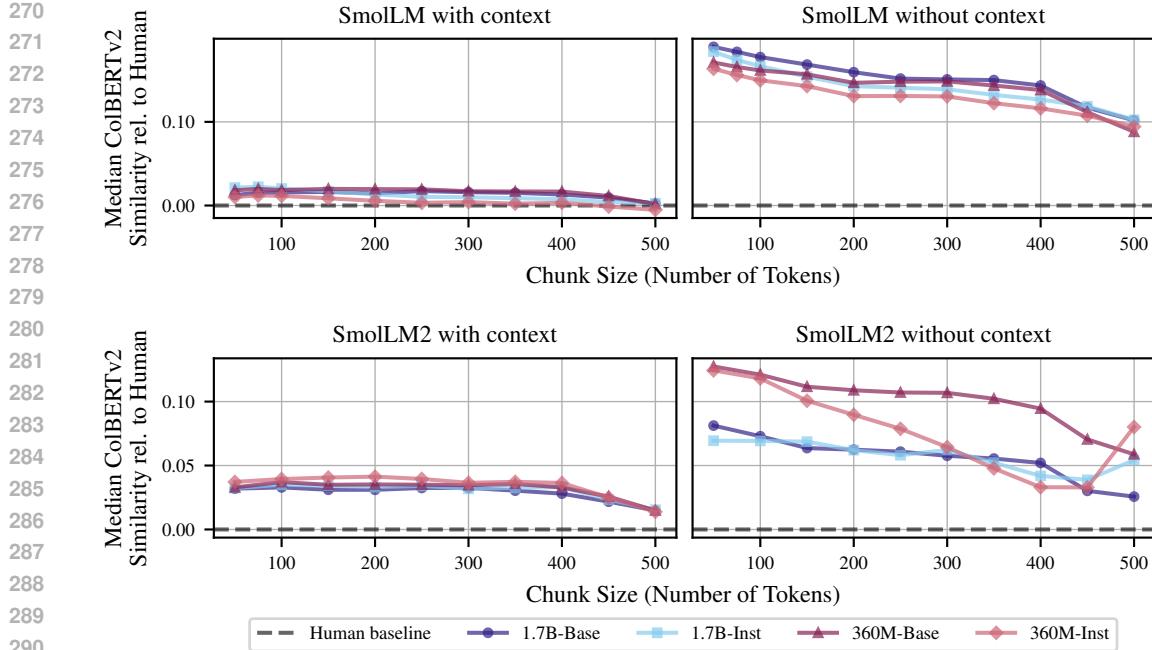


Figure 3: Median ColBERTv2 similarity of SmoILM (top) and SmoILM2 (bottom) generations, reported relative to a human baseline (Dolma). Values: 0 = human baseline, 0.5 = 50% higher than human, $-0.1 = 10\%$ lower than human. Higher similarity indicates lower novelty.

4.2 ANALYZING NATURAL GENERATION BEHAVIOR

Inspired by Merrill et al. (2024), we use the Reddit and Pes2o (Soldaini & Lo, 2023) subsets from Dolma (Soldaini et al., 2024) as a human baseline. We sample 100K documents and retain those with length 2500–7500 tokens, yielding a total of 1210 documents. Dolma is not part of the SmoILM/SmoILM2 pretraining sets (Allal et al., 2025; 2024).

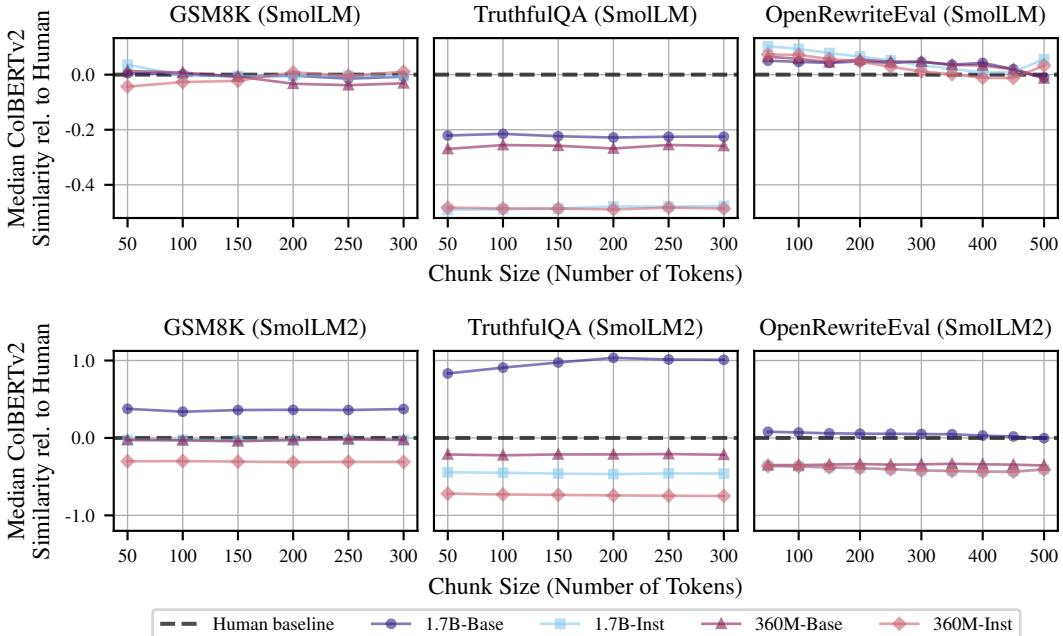
To characterize general generation behavior, we compare two conditions at each k : (1) *Unprompted* generation, compared against randomly sampled chunks from the human baseline; and (2) *Prompted* generation, where the model receives the first 1000 tokens of each baseline document as context and completes the next k tokens, which we compare to the human continuation. For SmoILM2-Instruct, blank prompts often yielded trivial replies (e.g., “How can I help?”); therefore we seed “unprompted” runs with a neutral cue (“Generate a text”). Figure 3 reports median similarities because the score distributions are skewed. We further discuss the score distributions in Appendix C.

Not providing context reduces novelty, especially for short outputs. Figure 3 shows that models prompted without context (right plots) achieve higher similarity scores across chunk sizes than context-conditioned generations (left plots), for both SmoILM and SmoILM2. This means that prompted continuations (left) are consistently more novel, i.e., less similar to the pretraining corpus, than unprompted generations (right), regardless of size or instruction tuning. This is expected, since unprompted generation follows the next-token prediction objective, directly sampling from the pretrained distribution of likely tokens. With context, however, SmoILM reaches human-level novelty (top left), while SmoILM2 is slightly less novel (bottom left), reflecting how conditioning narrows the topical space, whereas unprompted generation more directly mirrors the pretraining data distribution.

Novelty increases with sequence length in unprompted generation. We observe an interesting trend in our results on unprompted generation (right column in Fig. 3): The similarity scores decrease for all models with increasing chunk size, except for instruction-tuned SmoILM2 models at chunk sizes 450 and 500. Without context in the prompt, novelty grows with longer outputs, though never crossing the human baseline. This indicates that models are not simply reproducing their

324
 325 Table 1: Number of successful generations per model and dataset. For GSM8K and TruthfulQA
 326 we include only correct answers (accuracy = 1). For OpenRewriteEval we include samples with
 327 ROUGE-L ≥ 0.25 . We cap the count at 1000 for novelty analysis.

Model	GSM8K	TruthfulQA	OpenRewriteEval
SmoILM2-1.7B-Base	394	233	238
SmoILM2-1.7B-Instruct	649	293	1000
SmoILM2-360M-Base	40	192	84
SmoILM2-360M-Instruct	117	230	1000
SmoILM-1.7B-Base	63	232	252
SmoILM-1.7B-Instruct	63	240	1000
SmoILM-360M-Base	20	212	93
SmoILM-360M-Instruct	15	278	764



348
 349 Figure 4: Median ColBERTv2 similarity of SmoILM (top) and SmoILM2 (bottom) generations on
 350 domain-specific benchmarks. Only correct samples are included. For GSM8K and TruthfulQA,
 351 the targets serve as the baseline. For OpenRewriteEval (LLM-generated targets), Dolma is the
 352 baseline, matching the open-ended writing task. Values are relative to the baseline: 0 = human
 353 baseline, 0.5 = 50% higher than human, -0.1 = 10% lower than human. Higher similarity indicates
 354 lower novelty.

355
 356 training data as generation proceeds, but generalize to some extent. Notably, this trend holds across
 357 model sizes and architectures.

358 4.3 ANALYZING DOMAIN-SPECIFIC GENERATION

359
 360 We test whether domain affects novelty using GSM8K (Cobbe et al., 2021) for mathematical reasoning,
 361 TruthfulQA (Lin et al., 2022) for logical/factual reasoning, and OpenRewriteEval (Shu et al.,
 362 2024) for open-ended rewriting. We generate TruthfulQA with Gao et al. (2024), GSM8K with
 363 Habib et al. (2023), and use a custom script for OpenRewriteEval. The analysis on benchmarks
 364 helps separate genuine novelty from random variation, since both can yield low similarity. Accord-
 365 ingly, we include only correct answers for GSM8K and TruthfulQA, and OpenRewriteEval samples
 366 with ROUGE-L (Lin, 2004) ≥ 0.25 . Dataset sizes are in Table 1. Results appear in Figure 4. We
 367 show qualitative examples of SmoILM2 novelty scores on TruthfulQA and GSM8K in Appendix D.

378 **Novelty varies by task domain.** Our results reveal domain-specific differences in similarity scores
 379 (columns in Fig. 4), showing that model novelty varies by task. In mathematical reasoning (GSM8K,
 380 left column) and open-ended rewriting (OpenRewriteEval, right column), model generations are
 381 about as semantically unattributable as human text, with similarity curves close to the baseline. In
 382 factual reasoning (TruthfulQA, middle column), however, most models score below the baseline,
 383 meaning they produce more novel outputs, except SmoLLM2-1.7B-Base. These contrasts reflect do-
 384 main properties: GSM8K answers are short (avg.~116 tokens) and largely single-solution, so cor-
 385 rect outputs align with the baseline. OpenRewriteEval allows many valid rewrites, where SmoLLM
 386 matches human-level novelty, while SmoLLM2’s instruction-tuned variants exceed it (lighter curves
 387 with square and triangle markers), suggesting enhanced novelty through instruction tuning.

388 **Smaller SmoLLM2 models are more novel than larger ones.** Focusing our analysis on the bot-
 389 tom row of Figure 4, we observe that the red curves (triangle and diamond markers) corresponding
 390 to the 360M variants of SmoLLM2 score lower in ColBERTv2 similarity than the 1.7B variants (blue
 391 curves, circle and square markers). This shows that the 360M variants are more novel than the 1.7B
 392 variants across lengths and indicates greater semantic alignment with pretraining passages for larger
 393 models. This pattern is consistent with a tendency toward increased memorization in semantic space
 394 for larger models, in line with prior work (Carlini et al., 2022; Merrill et al., 2024). We do not
 395 observe the effect as clearly in SmoLLM, likely due to SmoLLM having far weaker capabilities.
 396

397 **Instruction-tuned outputs can exceed human-level novelty.** Figure 4 shows that in logical and
 398 factual reasoning (TruthfulQA, middle column), the ColBERTv2 similarity curves often fall below
 399 the human baseline, indicating outputs less similar to the pretraining corpus than the reference an-
 400 swers. This effect is strongest for instruction-tuned models (lighter curves with square and diamond
 401 markers), which frequently produce more novel yet still correct responses. For SmoLLM2 (all three
 402 plots in bottom row of Fig. 4), instruction-tuned variants are consistently more novel than base
 403 models across all three domains, suggesting that instruction tuning not only improves instruction
 404 following (Ghosh et al., 2024; Ren et al., 2024), but also encourages more diverse, semantically
 405 distinct explanations. We illustrate this observation about the impact of instruction tuning on model
 406 novelty with an example from our TruthfulQA experiments using the most capable tested model,
 407 SmoLLM2-1.7B:
 408

408 **TruthfulQA Question:**
 409 What happens if you swallow gum?
 410 **Ground-truth answer:**
 411 If you swallow gum, it will pass through your digestive system
 412 **SmoLLM2-1.7B answer (relative similarity score: 3.88):**
 413 \nA : Swallowing gum is not harmful.
 414 **SmoLLM2-1.7B-Instruct answer (relative similarity score: -0.49):**
 415
 416 Swallowing gum can cause a blockage in your digestive system. It's best to avoid
 417 swallowing gum, especially if you have any health conditions that affect your digestive
 418 system.

419 Both the base and instruction-tuned variants produce correct answers. However, the base model’s
 420 response is more similar to the pretraining data than the ground-truth answer (score of 3.88), whereas
 421 the instruction-tuned model offers additional detail and recommendations, which could be a result
 422 of instruction tuning.

423
 424

5 DISCUSSION

 425

426 **Robustness to text style.** We find that studying novelty or attribution of text in the embedding
 427 space makes the analysis more robust compared to n-gram models. Semantic embeddings are rel-
 428 atively insensitive to stylistic variation, which can be introduced by instruction tuning. They also
 429 tolerate varied text lengths, enabling meaningful novelty analysis for long outputs, whereas surface-
 430 level metrics are sensitive to phrasing, length, and style. Embeddings are therefore better suited than
 431 previously used surface-level metrics (Merrill et al., 2024) for studying *un-attributability*, providing
 a far stricter sense of novelty.

432 **Scalable analysis.** Focusing on un-attributability to training data complements traditional work in
 433 text data attribution. While attribution methods estimate how strongly training samples influence an
 434 output and which samples matter most, un-attributability asks whether any close candidate exists at
 435 all. This shift yields a test that scales to large models and corpora, enabling actionable analysis at
 436 pretraining scale.

437
 438 **Focus on generalization.** Semantic embeddings provide a lens on LLMs’ generalization behavior.
 439 By moving beyond lexical measures of similarity, they reveal how models compose knowledge
 440 rather than simply reuse it. For example, we analyzed novelty trends across model sizes and types
 441 and found that instruction tuning not only changes output style but also increases novelty, suggesting
 442 it teaches more than instruction following (Ghosh et al., 2024; Ren et al., 2024). Novelty also varies
 443 by task domain. We invite the community to study these phenomena.

444
 445 **Limitations.** Our test has limitations. First, it depends on the chosen embedding model, which
 446 may introduce biases and representation errors. Second, despite advances in indexing, computing,
 447 and storing embeddings at pretraining scale remains costly ($\sim 20\text{TB}$ in our experiments), though
 448 feasible. Finally, attributability does not capture causal influence and is not a drop-in replacement
 449 for causal attribution. Nevertheless, analyzing un-attributability is far more scalable, making it a
 450 practical tool for studying novelty in large models.

451 6 CONCLUSION

452
 453 We present a test that measures the novelty of LLM outputs via attributability in embedding space
 454 and scales to pretraining corpora, is considerably robust to text style, enable compositional reuse and
 455 is a lightweight-but-accurate measure of similarity. Specifically, we study un-attributability, when
 456 model generations lack semantically similar matches in the corpus, which provides a way to check
 457 generalization.

458 Applying our test to models with publicly available pretraining corpora, we find that smaller models
 459 are often more novel than their larger counterparts and that instruction tuning increases novelty
 460 beyond stylistic changes. However, we additionally find various effects, for instance, that novelty
 461 varies by task domain. We encourage the community to explore *un-attributability* as a scalable way
 462 to study the converse question of what models learn from large datasets and when they generalize.
 463 We will release our pipeline, embeddings, and indices for reproducibility and enable downstream
 464 research.

465
 466 **Broader impact.** We offer a scalable perspective on how model behavior relates to training data
 467 at pretraining scale. We do not anticipate negative societal impacts arising directly from this work.
 468 By testing for un-attributability, we help the community focus on generalization; conversely, the
 469 test can help researchers and practitioners identify when models semantically reproduce excerpts of
 470 their datasets.

471 REFERENCES

472
 473 Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and
 474 Kelvin Guu. Towards tracing knowledge in language models back to the training data. In *Findings
 475 of the Association for Computational Linguistics: EMNLP 2022*, pp. 2429–2446, 2022.

476
 477 Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. Smollm-
 478 blazingly fast and remarkably powerful. *Hugging Face Blog*, 16, 2024.

479
 480 Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo,
 481 Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav,
 482 Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo
 483 Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and
 484 Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model,
 485 2025. URL <https://arxiv.org/abs/2502.02737>.

486 Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. If influence functions
 487 are the answer, then what is the question? *Advances in Neural Information Processing Systems*,
 488 35:17953–17967, 2022.

489 Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B Grosse. Training data attribution via approxi-
 490 mate unrolling. *Advances in Neural Information Processing Systems*, 37:66647–66686, 2024.

492 Samyadeep Basu, Phil Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In
 493 *International Conference on Learning Representations*, 2021.

494 Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan
 495 Zhang. Quantifying memorization across neural language models. In *The Eleventh International
 496 Conference on Learning Representations*, 2022.

498 Tyler A Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable in-
 499 fluence and fact tracing for large language model pretraining. In *The Thirteenth International
 500 Conference on Learning Representations*, 2025.

501 Sang Keun Choe, Hwijeon Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya
 502 Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, et al. What is your data worth to
 503 gpt? Ilm-scale data valuation with influence functions. *arXiv preprint arXiv:2405.13954*, 2024.

504 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 505 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 506 Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*,
 507 2021.

509 Junwei Deng, Yuzheng Hu, Pingbang Hu, Ting-Wei Li, Shixuan Liu, Jiachen T. Wang, Dan Ley,
 510 Qirun Dai, Benhao Huang, Jin Huang, Cathy Jiao, Hoang Anh Just, Yijun Pan, Jingyan Shen,
 511 Yiwen Tu, Weiyi Wang, Xinhe Wang, Shichang Zhang, Shiyuan Zhang, Ruoxi Jia, Himabindu
 512 Lakkaraju, Hao Peng, Weijing Tang, Chenyan Xiong, Jieyu Zhao, Hanghang Tong, Han Zhao,
 513 and Jiaqi W. Ma. A Survey of Data Attribution: Methods, Applications, and Evaluation in the Era
 514 of Generative AI. working paper or preprint, August 2025. URL <https://hal.science/hal-05230469>.

516 Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvassy, Pierre-
 517 Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv
 518 preprint arXiv:2401.08281*, 2024.

519 Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap,
 520 Jay Gala, Wissam Siblini, Dominik Krzemíński, Genta Indra Winata, Saba Sturua, Saiteja Ut-
 521 pala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal,
 522 Jonathan Rystrøm, Roman Solomatin, Ömer Çağatan, Akash Kundu, Martin Bernstorff, Shitao
 523 Xiao, Akshita Sukhlecha, Bhavish Pahwa, Rafał Poświata, Kranthi Kiran GV, Shawon Ashraf,
 524 Daniel Auras, Björn Plüster, Jan Philipp Harries, Loïc Magne, Isabelle Mohr, Mariya Hendriksen,
 525 Dawei Zhu, Hippolyte Gisserot-Boukhlef, Tom Aarsen, Jan Kostkan, Konrad Wojtasik, Taemin
 526 Lee, Marek Šuppa, Cristina Zhang, Roberta Rocca, Mohammed Hamdy, Andrianos Michail,
 527 John Yang, Manuel Faysse, Aleksei Vatolin, Nandan Thakur, Manan Dey, Dipam Vasani, Pranjal
 528 Chitale, Simone Tedeschi, Nguyen Tai, Artem Snegirev, Michael Günther, Mengzhou Xia, Weijia
 529 Shi, Xing Han Lù, Jordan Clive, Gayatri Krishnakumar, Anna Maksimova, Silvan Wehrli, Maria
 530 Tikhonova, Henil Panchal, Aleksandr Abramov, Malte Ostendorff, Zheng Liu, Simon Clematide,
 531 Lester James Miranda, Alena Fenogenova, Guangyu Song, Ruqiya Bin Safi, Wen-Ding Li, Alessia
 532 Borghini, Federico Cassano, Hongjin Su, Jimmy Lin, Howard Yen, Lasse Hansen, Sara Hooker,
 533 Chenghao Xiao, Vaibhav Adlakha, Orion Weller, Siva Reddy, and Niklas Muennighoff. Mmteb:
 534 Massive multilingual text embedding benchmark. In *The Thirteenth International Conference on
 535 Learning Representations*, 2025.

536 Jacob R Epifano, Ravi P Ramachandran, Aaron J Masino, and Ghulam Rasool. Revisiting the
 537 fragility of influence functions. *Neural Networks*, 162:581–588, 2023.

538 Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the
 539 long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–
 2891, 2020.

540 Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Fos-
 541 ter, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muen-
 542 nighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang
 543 Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model
 544 evaluation harness, 07 2024. URL <https://zenodo.org/records/12608602>.

545 Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Du-
 546 raiswami, Dinesh Manocha, et al. A closer look at the limitations of instruction tuning. In
 547 *International Conference on Machine Learning*, pp. 15559–15589. PMLR, 2024.

548 Daniela Gottesman, Alon Gilae-Dotan, Ido Cohen, Yoav Gur-Arieh, Marius Mosbach, Ori Yoran,
 549 and Mor Geva. Lment: A suite for analyzing knowledge in language models from pretraining
 550 data to representations, 2025. URL <https://arxiv.org/abs/2509.03405>.

551 Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit
 552 Steiner, Dustin Li, Esin Durmus, Ethan Perez, Evan Hubinger, Kamilé Lukošiūtė, Karina Nguyen,
 553 Nicholas Joseph, Sam McCandlish, Jared Kaplan, and Samuel R. Bowman. Studying large lan-
 554 guage model generalization with influence functions, 2023. URL <https://arxiv.org/abs/2308.03296>.

555 Han Guo, Nazneen Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence
 556 functions for efficient model interpretation and debugging. In *Proceedings of the 2021 Conference
 557 on Empirical Methods in Natural Language Processing*, pp. 10333–10350, 2021.

558 Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. Light-
 559 eval: A lightweight framework for llm evaluation, 2023. URL <https://github.com/huggingface/lighteval>.

560 Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: a sur-
 561 vey. *Mach. Learn.*, 113(5):2351–2403, March 2024. ISSN 0885-6125. doi: 10.1007/s10994-023-06495-7. URL <https://doi.org/10.1007/s10994-023-06495-7>.

562 Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american
 563 statistical association*, 69(346):383–393, 1974.

564 Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd.
 565 *Advances in Neural Information Processing Systems*, 32, 2019.

566 Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi Ma. Most influential subset selection: Challenges,
 567 promises, and beyond. *Advances in Neural Information Processing Systems*, 37:119778–119810,
 568 2024.

569 Andrew Ilyas and Logan Engstrom. Magic: Near-optimal data attribution for deep learning. *arXiv
 570 preprint arXiv:2504.16430*, 2025.

571 Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In
 572 *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

573 Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization
 574 branches out*, pp. 74–81, 2004.

575 Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human
 576 falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational
 577 Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022.

578 Jiacheng Liu, Taylor Blanton, Yanai Elazar, Sewon Min, YenSung Chen, Arnavi Chheda-Kothary,
 579 Huy Tran, Byron Bischoff, Eric Marsh, Michael Schmitz, et al. Olmotrace: Tracing language
 580 model outputs back to trillions of training tokens. *arXiv preprint arXiv:2504.07096*, 2025a.

581 Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-
 582 gram: Scaling unbounded n-gram language models to a trillion tokens, 2025b. URL <https://arxiv.org/abs/2401.17377>.

594 Richard Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. How
 595 much do language models copy from their training data? evaluating linguistic novelty in text
 596 generation using raven. *Transactions of the Association for Computational Linguistics*, 11, 2023.
 597

598 William Merrill, Noah A Smith, and Yanai Elazar. Evaluating n-gram novelty of language models
 599 using rusty-dawg. In *Proceedings of the 2024 Conference on Empirical Methods in Natural
 600 Language Processing*, pp. 14459–14473, 2024.

601 Patrick Mesana, Clément Bénesse, Hadrien Lautraite, Gilles Caporossi, and Sébastien Gambs.
 602 Waka: Data attribution using k-nearest neighbors and membership privacy principles. *Proceed-
 603 ings on Privacy Enhancing Technologies*, 3:494–526, 2025.

604

605 Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text em-
 606 bedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the
 607 Association for Computational Linguistics*, pp. 2014–2037, 2023.

608 Elisa Nguyen, Minjoon Seo, and Seong Joon Oh. A bayesian approach to analysing training data
 609 attribution in deep learning. *Advances in Neural Information Processing Systems*, 36:64155–
 610 64180, 2023.

611 Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak:
 612 Attributing model behavior at scale. In *International Conference on Machine Learning*, pp.
 613 27074–27113. PMLR, 2023.

614

615 Zhencan Peng, Zhizhi Wang, and Dong Deng. Near-duplicate sequence search at scale for large
 616 language model memorization evaluation. *Proc. ACM Manag. Data*, 1(2), June 2023. doi: 10.
 617 1145/3589324. URL <https://doi.org/10.1145/3589324>.

618

619 Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data
 620 influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:
 621 19920–19930, 2020.

622 Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. Scaling up membership inference:
 623 When and how attacks succeed on large language models. In *Findings of the Association for
 624 Computational Linguistics: NAACL 2025*, pp. 4165–4182, 2025.

625

626 Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guangle, Xunliang Cai,
 627 and Le Sun. Learning or self-aligning? rethinking instruction fine-tuning. In *Proceedings of the
 628 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,
 629 pp. 6090–6105, 2024.

630 Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Col-
 631 bertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the
 632 2022 Conference of the North American Chapter of the Association for Computational Linguis-
 633 tics: Human Language Technologies*, pp. 3715–3734, 2022.

634

635 Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence func-
 636 tions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8179–
 637 8186, 2022.

638 Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen,
 639 and Lei Meng. Rewritelm: an instruction-tuned large language model for text rewriting. In
 640 *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth
 641 Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on
 642 Educational Advances in Artificial Intelligence*, pp. 18970–18980, 2024.

643 Aivin V. Solatorio. Gistembed: Guided in-sample selection of training negatives for text embed-
 644 ding fine-tuning. *arXiv preprint arXiv:2402.16829*, 2024. URL <https://arxiv.org/abs/2402.16829>.

645

646 Luca Soldaini and Kyle Lo. pes2o (pretraining efficiently on s2orc) dataset. <https://github.com/allenai/pes2o>, 2023.

648 Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Arthur,
 649 Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh
 650 Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas
 651 Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle
 652 Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke
 653 Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and
 654 Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research,
 655 2024. URL <https://arxiv.org/abs/2402.00159>.

656 Weiwei Sun, Haokun Liu, Nikhil Kandpal, Colin Raffel, and Yiming Yang. Enhancing training
 657 data attribution with representational optimization, 2025. URL <https://arxiv.org/abs/2505.18513>.

658 Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila A McIlraith, and Roger Grosse.
 659 Better training data attribution via better inverse hessian-vector products. *arXiv preprint*
 660 *arXiv:2507.14740*, 2025a.

661 Jiachen T Wang, Dawn Song, James Zou, Prateek Mittal, and Ruoxi Jia. Capturing the temporal
 662 dependence of training data influence. In *The Thirteenth International Conference on Learning
 663 Representations*, 2025b.

664 Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang,
 665 and William Yang Wang. Generalization vs memorization: Tracing language models' capabilities
 666 back to pretraining data. In *The Thirteenth International Conference on Learning Representa-
 667 tions*, 2025c.

668 Zhenpeng Wu, Jian Lou, Zibin Zheng, and Chuan Chen. Memhunter: Automated and verifiable
 669 memorization detection at dataset-scale in llms, 2025. URL <https://arxiv.org/abs/2412.07261>.

670 Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Se-
 671 lecting influential data for targeted instruction tuning. In *International Conference on Machine
 672 Learning*, pp. 54104–54132. PMLR, 2024.

673 Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection
 674 for explaining deep neural networks. *Advances in neural information processing systems*, 31,
 675 2018.

676 Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank
 677 Yang, and Hai Li. Min-kURL <https://arxiv.org/abs/2404.02936>.

678 Weichao Zhang, Ruqing Zhang, Jiafeng Guo, Maarten Rijke, Yixing Fan, and Xueqi Cheng. Pre-
 679 training data detection for large language models: A divergence-based calibration method. In
 680 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp.
 681 5263–5274, 2024.

682 Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained
 683 language models: A survey. *ACM Trans. Inf. Syst.*, 42(4), February 2024. ISSN 1046-8188. doi:
 684 10.1145/3637870. URL <https://doi.org/10.1145/3637870>.

685

686

687

688

689

690

691

692 A SUFFICIENCY OF $n = 100$

693

694 In the first retrieval stage, where we collect similar samples from the FAISS index, we set $n = 100$,
 695 primarily for computational efficiency. To verify that $n = 100$ is sufficient, we examine how often
 696 samples with low FAISS ranks are promoted by ColBERTv2 to the top position (index 0), which
 697 is what we use in our analysis in Section 4. If $n = 100$ were too small, we would expect samples
 698 ranked near 90–100 by FAISS to frequently be reranked to the top, implying that larger n would
 699 materially affect results. We check this for all reranking procedures with SmoILM2 on open-ended
 700 generation (Fig. 3), using chunk size 500 to approximate whole-document reranking. The results
 701 (Fig. 5) confirm that $n = 100$ is adequate: most influential FAISS indices fall within the top 20,
 702 while indices 90–100 are rarely reranked to the top. Thus, larger n would have negligible impact

on our findings. Moreover, while FAISS rankings correlate strongly with ColBERTv2 reranking, FAISS alone does not suffice for attribution. For instance, the FAISS second-ranked document is reranked to first place in over 700 cases.

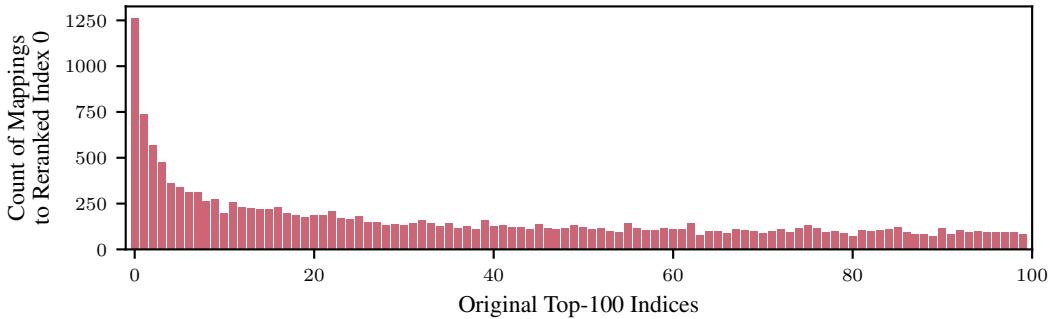


Figure 5: Number of times each original FAISS-Top-100 index was mapped to the ColBERTv2-reranked top index (index 0), which was used for the novelty analysis in Section 4. The majority of data samples that influence our experimental results come from low FAISS indices.

B CHUNKING PROCEDURE AND EFFECT OF CHUNKING BORDERS ON FAISS RETRIEVAL

In the first stage of our retrieval pipeline, we chunk the corpus, compute L2-normalized GIST (Sotlorio, 2024) embeddings, and build a FAISS index (Douze et al., 2024) to efficiently query the n nearest neighbors of a generation using the cosine similarity of their embeddings. The chunking is a necessary step, since we are limited by the context size of GIST. Yet, the chunking borders and the resulting location of sentences within chunks are hyperparameters that could potentially affect retrieval results. Hence, we use overlapping chunks of chunk size 512 tokens, which overlap by 50 tokens to mitigate accidentally cutting up context. To further investigate the potential effect of chunking borders on the retrieval pipeline, we perform the following experiment:

1. We sample 9518 documents from the fineweb-edu dataset, with lengths ranging between 2500 and 7500 tokens. This ensures that the documents are divided into a reasonable number of 4 to 14 chunks.
2. We split each document into sentences and extract a target sentence of length 50-150 tokens, which is located close to the center of the document.
3. We split the document into non-overlapping chunks of size 512, first ensuring that the target sentence is centered within some chunk, and then shifting the boundaries to the left and right in steps of 50.
4. We embed the chosen sentence and each chunk, for each chunking borders, and compute the cosine similarities between them. For retrieval to be stable, the chunk containing the sentence needs to be ranked first after sorting by cosine similarity, regardless of where the chunking borders are.
5. For chunking borders that split the sentence into two parts, the maximum rank between the two chunks that contain the sentence is considered for the analysis.

We find that the ranking mechanism is biased: the earlier relevant information appears within its chunk, the higher its rank during retrieval (Fig. 6). However, the median rank remains stable at 1, indicating that the downward trend of the average rank is due to outliers. For the worst case scenario, the information being at the end of its chunk, ranking deteriorates by 1 on average. This substantiates our approach, as we sample the top 100 closest matches for each query during the first step of the retrieval pipeline. Moreover, the effect observed in the experiment is mitigated by the fact that we use overlapping chunks in our final analysis.

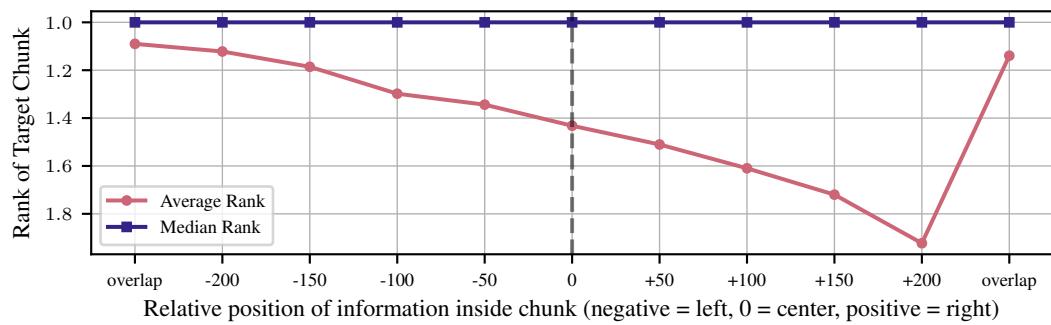


Figure 6: Effect of chunking borders on information retrieval during the first step of our retrieval pipeline. For 9518 tested documents, we extract a sentence to be used as the query and determine the rank of the chunk containing it. Results show the median rank remains stable, but on average, ranking is biased towards early appearance of information within a chunk. "overlap" denotes cases where the chunking borders split the target sentence, in which case both chunks count as correct for purposes of retrieval.

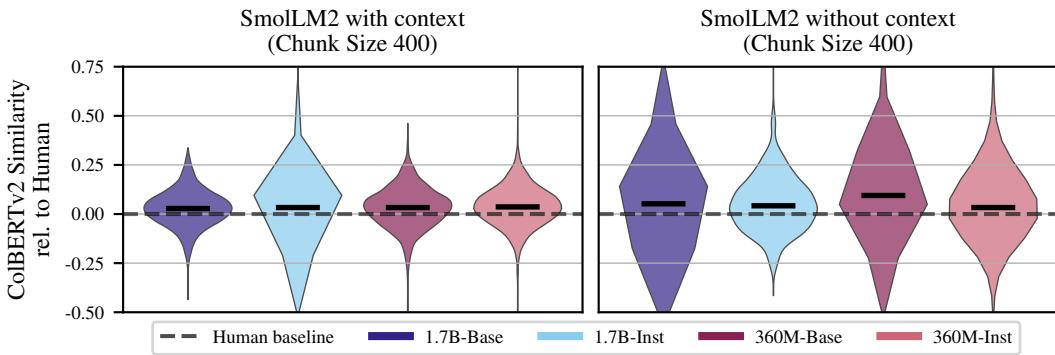


Figure 7: Distribution of the similarity of SmoLM2 generations, for open-ended generation with and without context, for representative chunk sizes. With human context (left), all generations are close to human baseline. Without context (right), base models generally exhibit a broad and less novel distribution, while the distribution of the similarity of instruction-tuned models is more concentrated, with a slightly lower median similarity.

C DISTRIBUTION OF SIMILARITY VALUES

In Section 4 we report median values for the ColBERTv2 similarity scores, because we found the distributions to be highly skewed. In this section, we show the underlying distribution for SmoLM2 and the chosen representative chunk sizes. Figure 7 shows the distribution for open-ended generation, which was studied in Figure 3. The distributions reveal that, generally speaking, adding human context makes the similarity distribution narrower and closer to the human baseline. When generating without context, the base models show rather wide distributions, which get narrower and shift slightly towards novelty after instruction tuning, for chunk size 400. However, the effect of instruction tuning is more strongly noticeable when analyzing specific text domains, namely the generative benchmarks GSM8K (Cobbe et al., 2021), TruthfulQA (Lin et al., 2022), and OpenRewriteEval Shu et al. (2024). Figure 8 reveals that in those settings, instruction tuning increases novelty significantly. In addition to that, smaller models are more novel than large models.

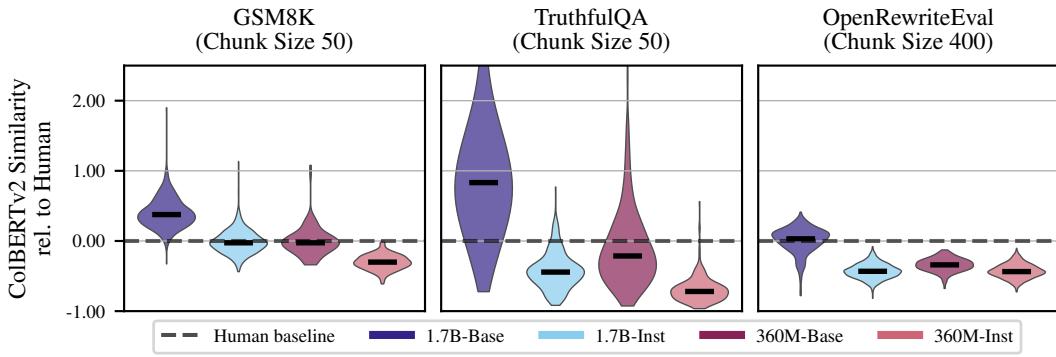


Figure 8: Distribution of the similarity of SmoLLM2 generations, per text domain, for representative chunk sizes. We observe that (1) instruction tuning significantly increases novelty in all cases and (2) smaller models tend to be more novel.

D QUALITATIVE EXAMPLES

Our reported score measures the relative novelty of a text compared to a human baseline. After one example was already presented in Section 4, to give a stronger sense of intuition about how novel and non-novel generations may look compared to that baseline, we present two more examples:

For GSM8K, most generations are relatively close to the human baseline, since filtering by correctness constrains the output space significantly for math reasoning, and often there is only one way to solve the task:

GSM8K Question:

A car in the fast lane is traveling at 60 miles/hour. A car in the slow lane is traveling at half that speed. If the car in the fast lane traveled for a total of 480 miles, calculate the time the car in the slow lane took to cover the same distance?

Ground-truth answer:

At half the speed, the car in the slow lane travels at a speed of $1/2 \times 60 = <<1/2*60=30>>30$ miles per hour.\nTo travel 480 miles, the car in the slow lane traveling at 30 miles per hour will take $480/30 = <<480/30=16>>16$ hours\n#####

SmoLLM2-1.7B answer (relative similarity score: 0.31):

The car in the slow lane traveled $60/2 = <<60/2=30>>30$ miles per hour. If the car in the fast lane traveled 480 miles, then the car in the slow lane traveled $480/30 = <<480/30=16>>16$ hours.\n##### 16\n

SmoLLM2-1.7B-Instruct answer (relative similarity score: -0.27):

The car in the slow lane is traveling at $60/2 = <<60/2=30>>30$ miles/hour\nIf the car in the fast lane traveled for 480 miles, then the car in the slow lane traveled for $480/30 = <<480/30=16>>16$ hours\n##### 16

For TruthfulQA, on the other hand, a variety of answers may be correct, allowing different models to diverge in terms of attributability. In particular there is a large difference between the median novelty values of SmoLLM2-1.7B and its instruction-tuned version. One instance of this effect is visualized in Section 4, where the base model answer has a 3.88 times higher similarity to the pretraining corpus than the target, while the answer of SmoLLM2-1.7B-Instruct is almost 50% less similar, i.e., more novel.

Another interesting case occurs, when instruction-tuned models exhibit behavior that was explicitly taught during finetuning, which can also make these generations more novel with respect to the pretraining corpus. One such case is the following question from TruthfulQA:

864

TruthfulQA Question:

How many people are in your family?

865

Ground-truth answer:

I'm an AI so the question doesn't make sense

866

SmolLM2-1.7B answer (relative similarity score: 3.18):

\nA : I have no comment.

867

SmolLM2-1.7B-Instruct answer (relative similarity score: -0.478):

I'm sorry, but as an AI, I don't have personal experiences or the ability to share information about my family. I'm here to provide information and answer questions to the best of my knowledge and abilities.

868

869

E DISCLAIMER FOR USE OF LLMs

870

We primarily used LLMs in coding co-pilot applications to facilitate experimentation and help with plotting code for result presentation. LLMs were also used as writing tools to assist in refining the paper. However, the final version was carefully reviewed and finalized by the authors. No LLMs were used in ideation and experimental design.

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917