Deduplicating Training Data Makes Language Models Better

A Paper Review

Professor Georgina Cosma

March 2025

Full Reference:
Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., & Carlini, N. (2022). Deduplicating Training Data Makes Language Models Better. *arXiv:2107.06499*.

Introduction: Background & Motivation

- **Context:** NLP progress relies on increasingly large text corpora
  - Datasets have grown from gigabytes to terabytes
  - Examples: Common Crawl, C4, RealNews, Wiki-40B
  - From Chelba et al. (2013) to Brown et al. (2020)

- **Quality vs. Quantity Problem:**
  - Large-scale datasets sacrifice quality for size
  - Manual review/curation impossible at scale
  - Data quality directly impacts model behavior
  - Models reflect the biases present in training data (Bender et al., 2021)

- **Research Gap:**
  - Understanding dataset quality is a research challenge
  - Previous work focused on types of content, not duplication
  - Quantitative understanding needed (Dodge et al., 2021a)

Introduction: The Problem of Data Duplication

- **Key Discovery:** Duplication is pervasive in common NLP datasets
  - All four studied datasets contained duplicates:
    - C4 (Colossal Clean Crawled Corpus) - 350GB web text
    - Wiki-40B - Multi-lingual Wikipedia text
    - LM1B (One Billion Word Benchmark) - News commentary
    - RealNews - News domain subset of Common Crawl

- **Types of Problematic Duplication:**
  - **Near-duplicates:** Documents with minor variations
  - **Exact substring duplicates:** Common text chunks across documents
  - **Train-test contamination:** Same text in both training and validation

- **Scope of the Problem:**
  - Found a 61-word sequence repeated 61,036 times in C4 training data
  - Same sequence appears 61 times in the validation set
  - Over 1% of tokens generated by models are verbatim copies from training
  - Up to 4% of validation examples have duplicates in training data

Introduction: Deduplication Methods

- **Challenge:** Performing thorough deduplication at scale is difficult
  - Naive deduplication (exact matching) is insufficient
  - Need to handle 350GB+ datasets efficiently
  - Must capture different types of duplication patterns

- **Method 1: EXACTSUBSTR - deduplicating exact substrings**
  - Identifies verbatim strings that are repeated across documents
  - Uses suffix arrays for efficient substring matching
  - Linear-time algorithm with 8 bytes per token memory usage
  - Removes substrings that appear multiple times (threshold: 50 tokens)

- **Method 2: NEARDUP - approximate deduplication based on matching entire examples**
  - Identifies documents with high n-gram overlap
  - Uses MinHash algorithm (Broder, 1997) for approximate matching
  - Efficient hash-based technique for estimating document similarity
  - Particularly effective for templated content with small variations

What are Suffix Arrays? An Explanation for Non-Experts

- **What is a suffix array?**
  - A data structure that stores all suffixes of a text in sorted order
  - A "suffix" is any substring that extends to the end of text
  - Used by EXACTSUBSTR to efficiently find repeated substrings
- **Simple example:** For the word "banana" (positions 1-6)
  - All suffixes: "banana", "anana", "nana", "ana", "na", "a"
  - Sorted alphabetically (lexicographically-ordered): "a", "ana", "anana", "banana", "na", "nana"
  - Suffix array stores starting positions: [6, 4, 2, 1, 5, 3]
  - This means: "a" starts at position 6, "ana" at position 4, etc.
- **Why they're useful:**
  - Memory efficient: stores only positions, not actual suffix text
  - Makes substring search extremely fast
  - Similar suffixes appear next to each other in the array
  - Can quickly identify all repeated sequences
  - Can process terabytes of text in hours instead of weeks

MinHash Algorithm: A Simple Explanation

- **What is MinHash?**
  - A technique to quickly estimate similarity between documents
  - Used by NEARDUP to find near-duplicate documents
  - Avoids comparing every single word in every document pair
- **How it works:**
  - Convert documents into sets of "shingles" (n-grams)
  - Example: "The quick brown fox" $\rightarrow$ ["The quick", "quick brown", "brown fox"]
  - Apply multiple hash functions to each n-gram
  - Keep only the smallest hash values as document "signatures"
  - Similar documents will have similar signatures
- **Why it's powerful:**
  - Can approximate document similarity in constant time
  - Detects documents with small differences
  - Scales to billions of documents
  - Popular in search engines and large-scale web analysis

Dataset Analysis: Duplication Statistics

Dataset	% Train Examples with Dup in Train	% Valid Examples with Dup in Valid	% Valid with Dup in Train
C4	3.04%	1.59%	4.60%
RealNews	13.63%	1.25%	14.35%
LM1B	4.86%	0.07%	4.92%
Wiki-40B	0.39%	0.26%	0.72%

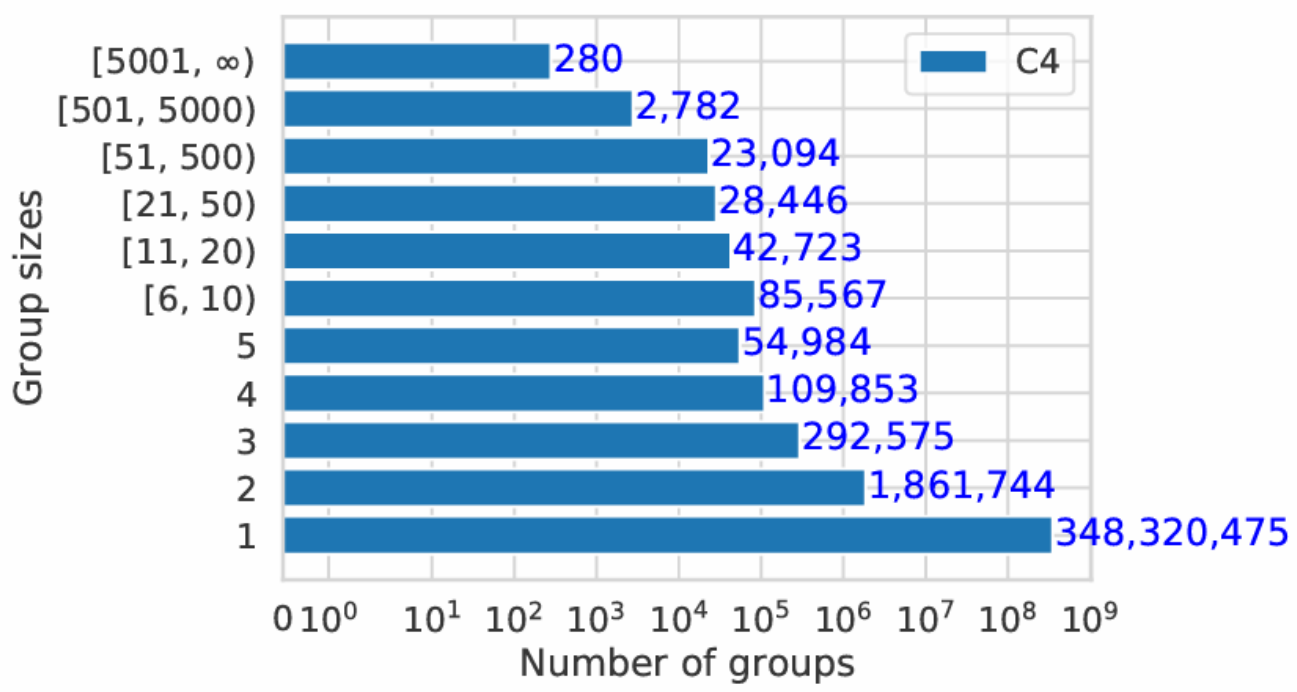
How to read this table: Higher percentages = more duplication. Shows what portion of each dataset contains duplicated content.

Dataset	% Train Tokens with Dup in Train	% Valid Tokens with Dup in Valid	% Valid with Dup in Train
C4	7.18%	0.75%	1.38%
RealNews	19.4%	2.61%	3.37%
LM1B	0.76%	0.016%	0.019%
Wiki-40B	2.76%	0.52%	0.67%

Top table: Near-duplicate documents (NEARDUP method) — **Bottom table:** Exact substring duplicates (EXACTSUBSTR method)

Key takeaway: RealNews has the most duplication (19.4% of tokens are duplicates). Even carefully curated Wiki-40B contains duplicates.

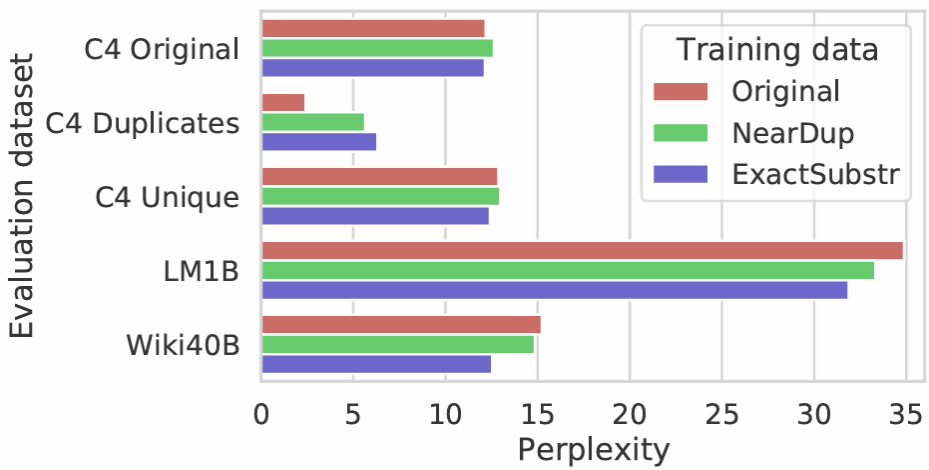
Near-Duplicate Cluster Analysis in C4



How to read this graph: X-axis = number of clusters, Y-axis = size of clusters (number of similar documents per group)

- Most common: clusters of 2 documents (1.8M pairs)
- Largest cluster: 250,933 similar documents!
- Total documents in near-duplicate clusters: 11M (3.04% of C4)
- 280 clusters contained over 5,000 examples each

Impact of Deduplication on Model Perplexity



How to read this graph: Lower bars = better model performance (perplexity is a measure of prediction quality; lower is better). Different colours show models trained on different datasets.

Key insights:

- **Perplexity** is the standard metric for language model quality - it measures how well the model predicts the next word in a sequence
- Models trained on deduplicated datasets perform equal or better than original models
- Particularly large improvement on Wiki-40B evaluation (almost 3 points)
- Deduplicated models show higher perplexity on duplicate validation examples (as expected)

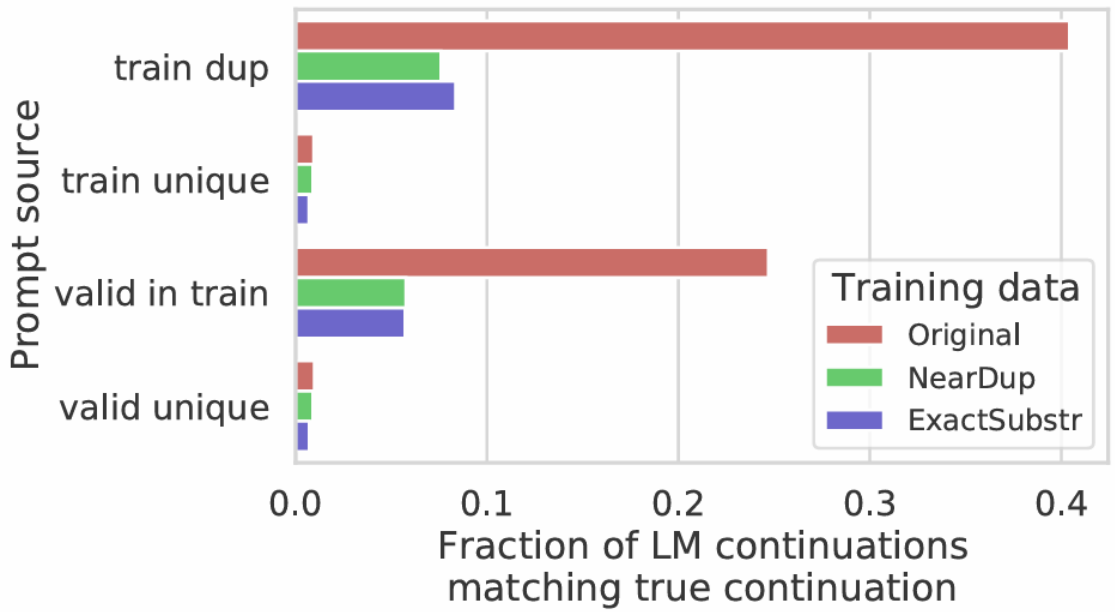
Reduction in Memorised Content Generation

Model	1 Epoch	2 Epochs
XL-ORIGINAL	1.926%	1.571%
XL-NEARDUP	0.189%	0.264%
XL-EXACTSUBSTR	0.138%	0.168%

How to read this table: Lower percentages = better (less memorisation). Shows what percentage of generated text was copied directly from training data.

- **Key finding:** Over 1% of tokens from models trained on original data are part of memorised sequences
- Deduplication reduces memorisation by approximately 10×
- Both deduplication methods show similar improvements
- Even after two epochs, memorisation stays low in deduplicated models

Memorisation When Using Prompts



How to read this graph: Higher bars = more copying. Different colours = different models. Shows how often models copy text when given various prompts.

Key insights:

- When prompted with duplicated content, original model reproduces ground truth over 40% of the time
- Deduplicated models show much less verbatim copying
- All models show higher matching when prompted with duplicated content
- Demonstrates how duplication biases models toward memorisation

Train-Test Contamination in Existing Models

Model	Dataset	Original	Duplicates	Unique
Transformer-XL	LM1B	21.77	10.11	23.58
GROVER-Base	RealNews	15.44	13.77	15.73
GROVER-XL	RealNews	9.15	7.68	9.45

How to read this table: Lower numbers = better performance (lower perplexity). "Duplicates" = validation examples with copies in training data. "Unique" = validation examples with no copies in training.

- **Major finding:** Existing models show much lower perplexity on validation examples with duplicates in training
- Transformer-XL's perplexity on duplicates is less than half its perplexity on unique examples. (Note Perplexity: how well the model predicts the next word in a sequence)
- All evaluated models show this pattern
- Suggests published model evaluations may be inflated due to contamination!!! Let's discuss this.

Key Benefits: A 4-Point Summary

1. **Reduced Memorisation (10× Improvement)**
   - Original data: 1.57% of generated text was memorised
   - Deduplicated data: Only 0.2% memorised
   - Result: More creative, less repetitive outputs

2. **Better Evaluation (More Accurate Testing)**
   - Removed overlap between training and validation sets
   - Prevents artificially inflated performance metrics
   - Result: We select models that truly generalise better

3. **Training Efficiency (Smaller Datasets)**
   - Deduplication shrinks datasets by up to 19%
   - Reduces training time, cost, and environmental impact
   - Result: More efficient model training

4. **Equal or Better Performance (No Downsides)**
   - No loss in model quality (perplexity stayed the same or improved)
   - All benefits come essentially "for free"
   - Result: A clear win with no observed disadvantages

Example Duplicates in Datasets

Dataset	Example	Near-Duplicate Example
Wiki-40B	`Hum Award for Most Impactful Character... In the list below, winners are listed first in the colored row, followed by the other nominees.`	`Hum Award for Best Actor in a Negative Role... In the list below, winners are listed first in the colored row, followed by the other nominees.`
LM1B	`I left for California in 1979 and tracked Cleveland 's changes on trips back to visit my sisters.`	`I left for California in 1979 , and tracked Cleveland 's changes on trips back to visit my sisters.`
C4	`Affordable and convenient holiday flights take off from your departure country, "Canada"... Book your Halifax (YHZ) - Basel (BSL) flight now...`	`Affordable and convenient holiday flights take off from your departure country, "USA"... Book your Maui Kahului (OGG) - Dubrovnik (DBV) flight now...`

What to notice: The highlighted text shows the parts that are identical between document pairs. These are typical examples of near-duplicates found by the NEARDUP algorithm.

- **Types of duplicates found:**
  - Wiki-40B: Template text with only the award name changed
  - LM1B: Almost identical sentences (just a comma difference)
  - C4: Template-based content with just location names changed
- These examples show how standard deduplication would miss these near-duplicates, as they're not 100% identical but contain very little unique information.

Method 1: EXACTSUBSTR Explained with an Example

- **Basic idea:**
  - Identifies and removes identical text chunks (≥ 50 tokens) chosen after statistical analysis
  - Common in copy-pasted content like legal disclaimers, advertisements
  - 50-token threshold: length at which identical sequences are extremely unlikely to occur naturally and almost always indicate copied content. Long enough to avoid false positives while still catching meaningful duplicates, and computationally manageable for processing massive datasets.

- **Real Example:**
  - News article 1: "Breaking news: The city council approved the budget proposal with funding for infrastructure, education, and community development..."
  - News article 2: "Local update: Yesterday, the council approved the budget proposal with funding for infrastructure, education, and community development..."
  - EXACTSUBSTR would identify and remove this duplicate passage from one article

Method 2: NEARDUP Explained with an Example

- **Basic idea:**
  - Find documents that are nearly identical but with small changes
  - Common with templated content, product descriptions, form letters
- **Real Example:**
  - Product 1: "The X500 laptop features a 15-inch display, 8GB RAM, and 256GB SSD storage. Perfect for students and professionals alike, with all-day battery life."
  - Product 2: "The X700 laptop features a 17-inch display, 16GB RAM, and 512GB SSD storage. Perfect for students and professionals alike, with all-day battery life."
  - Nearly identical template with just a few specific details changed
- **Why this matters:**
  - These documents wouldn't be caught by exact matching
  - But they add very little unique information to the dataset
  - Can cause models to overfit to specific templates
- NEARDUP catches structurally identical but have small differences. Like product listings that only vary by model number or price, they teach the model to memorize templates rather than understand language

What is Perplexity? A Simple Explanation

- **What is perplexity?**
  - A measure of how well a language model predicts text
  - Think of it as the model's level of "confusion" when predicting the next word. **Lower perplexity = better predictions**
  - Perplexity measures uncertainty or confusion. How "perplexed" or surprised the model is by the text. Lower values mean less confusion. A perfect model would have a perplexity of 1 (complete certainty)
- **Analogy:**
  - Imagine you're reading a book in a language you're learning
  - If you can easily predict the next word, you have low "perplexity"
  - If you're constantly surprised by the words that appear, you have high "perplexity"
- **What the paper found:**
  - Deduplication does **not** hurt perplexity. In some cases, it actually improved perplexity by up to 10%
  - Shows models can be more efficient when trained on deduplicated data
- The surprise finding was that removing duplicated text made models better at predicting new text, not worse

Impact on Memorisation: Simplified Explanation

- **What is memorisation?**
  - When a language model reproduces exact text from its training data
  - Like copying from a textbook instead of writing in your own words
- **Why it's a problem:**
  - Models should generalise, not just repeat (information) without analysing or comprehending it. Can expose private or sensitive information from training data. Gives a false impression of model capabilities
- **What the paper found:**
  - Models trained on originaldata:$>$1.5% of generated text was memorised
  - Models trained on deduplicated data:Appx. 0.2% memorised(10$\times$ less)
  - When prompted with duplicate content, the difference was more dramatic
  - Original model: 34% memorisation vs Deduplicated models: 3-5%
- This finding is really important for privacy and creativity. Models trained on deduplicated data are 10 times less likely to reproduce private information verbatim, and they produce more original, less copied content.

Train-Test Contamination: A Simple Explanation

- **What is train-test contamination?**
  - When the same text appears in both training and testing data
  - Like giving students exam questions they've already seen the answers to
- **Why it's a problem:**
  - Models can simply memorise answers instead of learning to generalise
  - Evaluation metrics become artificially inflated
  - We might select models that are good at memorising rather than understanding
- **What the paper found:**
  - Transformer-XL(a popular model):Perplexity on duplicate validation:10.11
  - Same model on unique validation: 23.58 (more than doubled!)
  - Similar patterns across other models (GROVER, etc.)
  - Deduplication ensures validation examples are truly unseen data
- This is maybe the most shocking finding-many published models look twice as good as they really are because of test contamination. It's like claiming a student is brilliant when they were actually given the test answers in advance

What the Methods Found

- **EXACTSUBSTR discovered:**
  - 7.18% of tokens in C4 were part of duplicate substrings
  - 19.4% of tokens in RealNews were duplicates
- **NEARDUP discovered:**
  - 3.04% of documents in C4 were near-duplicates
  - 13.63% of documents in RealNews were near-duplicates
  - Found one cluster with 250,933 similar documents!
- **Most striking example:**
  - A 61-word sequence about wedding design ideas
  - Repeated verbatim 61,036 times in C4 training dataset
  - Also appeared 61 times in the validation set
- These numbers should be shocking. Imagine if 1 in 5 pages in a textbook were duplicates of other pages. That's what's happening in RealNews. And the wedding design example shows how weird patterns can dominate these datasets.

Dataset Size Reduction After Deduplication

Dataset	Original Size	After NEARDUP	After EXACTSUBSTR
C4	177.3B tokens	173.7B tokens	165.4B tokens
RealNews	24.7B tokens	22.4B tokens	20.1B tokens
LM1B	1.0B tokens	0.94B tokens	0.90B tokens
Wiki-40B	2.25B tokens	2.24B tokens	2.19B tokens

How to read this table: Lower numbers in "After" columns = more duplicates removed. Shows dataset sizes before and after applying each deduplication method.

- **Most significant reductions:**
  - RealNews: **19%** smaller after both methods (4.6B tokens removed)
  - C4: **7%** smaller after EXACTSUBSTR (11.9B tokens removed)
- **Practical impact:** Smaller datasets mean faster, cheaper, and more environmentally friendly training
- Even with just basic deduplication, we can significantly reduce dataset size. RealNews is reduced by nearly one-fifth, showing how much redundant data it contained.

Two Complementary Methods: A Library Analogy

- **Imagine you're organising a library:**
  - EXACTSUBSTR is like finding books with identical chapters
  - You'd keep only one copy of each chapter to save space
  - Example: Legal disclaimers, standard introductions, etc.
- **Meanwhile...**
  - NEARDUP is like finding different editions of the same book
  - You'd keep only the most representative edition
  - Example: Nearly identical product descriptions with minor variations
- **Working together:**
  - EXACTSUBSTR catches identical passages across different documents
  - NEARDUP catches whole documents that are nearly identical
  - Together they provide comprehensive deduplication
  - Like removing both partial and complete duplicates from your library

Conclusion

- **Main message:** Dataset deduplication is important and beneficial
- **Practical outcomes:**
  - Authors released deduplicated datasets
  - Authors released deduplication tools
  - Future research should incorporate deduplication
- **Benefits summary:**
  - No negative impact on perplexity (sometimes improves it)
  - 10× reduction in memorised content
  - More reliable evaluation metrics
  - More efficient training (smaller datasets)
  - Negligible computational cost compared to training
- **Bottom line:** "Data deduplication offers significant advantages and no observed disadvantages"