NOISE-ROBUST DE-DUPLICATION AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Identifying near duplicates within large, noisy text corpora has a myriad of applications that range from de-duplicating training datasets, reducing privacy risk, and evaluating test set leakage, to identifying reproduced news articles and literature within large corpora. Across these diverse applications, the overwhelming majority of work relies on N-grams. Limited efforts have been made to evaluate how well N-gram methods perform, in part because it is unclear how one could create an unbiased evaluation dataset for a massive corpus. This study uses the unique timeliness of historical news wires to create a 27,210 document dataset, with 122,876 positive duplicate pairs, for studying noise-robust de-duplication. The time-sensitivity of news makes comprehensive hand labelling feasible - despite the massive overall size of the corpus - as duplicates occur within a narrow date range. The study then develops and evaluates a range of de-duplication methods: hashing and N-gram overlap (which predominate in the literature), a contrastively trained bi-encoder, and a "re-rank" style approach combining a bi- and cross-encoder. The neural approaches significantly outperform hashing and N-gram overlap. We show that the bi-encoder scales well, de-duplicating a 10 million article corpus on a single GPU card in a matter of hours. The public release of our `NEWS-COPY` de-duplication dataset will facilitate further research and applications.

1 INTRODUCTION

Robust identification of near-duplicate texts in large, noisy corpora is important for a variety of applications. Duplication in training data degrades model performance Lee et al. (2021) and can also raise serious privacy risks: Kandpal et al. (2022) show that there is a superlinear relationship between a sequence's count and the rate at which language models regenerate training sequences, with a sequence appearing 10 times in training data being a thousand times more likely to be regenerated than a sequence appearing only once. Additionally, the presence of test set leakage complicates evaluation of model performance, concerns that are elevated with large language models that have greater capacity to memorize training data or can consult an external database. In introducing the retrieval-enhanced transformer model, Borgeaud et al. (2022) conclude: "Further work is yet needed to better understand the role of test set leakage in the performance of LMs." Patterns of duplication are also themselves of interest, for studying the dissemination of reproduced content such as literature or news (Cordell, 2015; Smith et al., 2015; Vesanto et al., 2017).

In contrast to the literature on semantic textual similarity, where deep neural architectures predominate - e.g. Reimers & Gurevych (2019) - text de-duplication overwhelmingly uses N-gram methods. There have been few efforts to formally evaluate the adequacy of N-gram based de-duplication or to explore potential performance gains from neural text de-duplication. This study builds a large de-duplication dataset and develops neural methods for robust textual de-duplication that significantly outperform N-gram based methods and scale efficiently.

A major hurdle to overcome in systematically studying text de-duplication is the lack of data for an unbiased evaluation of different methods. Typically, there is no way to exhaustively identify all duplicates of a given example in a large corpus, complicating comparisons of recall.

To circumvent this challenge, we examine duplication in a unique context: historical news. Reproduction of news from news wires and syndicate services was widespread, forming over half the content of U.S. local newspapers. Media historian Julia Guarneri (2017) writes: "by the 1910s and

1920s, most of the articles that Americans read in their local papers had either been bought or sold on the national news market... This constructed a broadly understood American 'way of life' that would become a touchstone of U.S. domestic politics and international relations throughout the twentieth century." Because news is timely, reproduction happens within a narrow time window, and hence annotators can exhaustively identify all duplicates despite the massive overall size of the corpus. To build an unbiased evaluation sample, highly skilled human annotators manually reviewed every front page article from 973 newspapers on four randomly chosen days in 1930, 1955, and 1974 to create clusters of duplicated articles (including all singletons). Additional data, spanning the period from 1920 to 1977, were compiled for model training. The resulting public NEWS-COPY dataset - which contains 27,203 articles, comprising 122,876 positive duplicate pairs - aims to encourage further study of robust de-duplication.

In the absence of evaluation data, the literature has largely assumed that text de-duplication is sufficiently simple that neural methods are not required. However, noise - rampant in user-generated or OCR'ed texts - can lead near duplicate documents to have low N-gram similarity. Amongst duplicated pairs of articles in the NEWS-COPY test set, the average number of 3-grams (4-grams, 5-grams) shared between pairs of reproduced articles, normalized by the number of N-grams in the shorter article, is 56% (50%, 45%). 19% of duplicates have no 10-grams in common and 31% have no 15-grams in common, often as a result of minor text noise. Neural methods are plausibly more robust in the presence of noise, which is a common feature of large text corpora more generally.

Using the NEWS-COPY dataset, we examine different text de-duplication methods that vary along two key dimensions: whether or not the method is neural and computational cost. Drawing inspiration from work on semantic textual similarity and on retrieval, we develop two approaches for neural text de-duplication: a contrastively trained bi-encoder plus clustering method and a reranking style method, which uses a computationally cheap transformer bi-encoder to measure the pairwise similarity between all articles and then passes each article's nearest neighbors to a cross-encoder, at an additional computational cost. We also examine N-gram overlap and locally sensitive hashing, the latter of which is highly scalable. The neural methods significantly outperform the non-neural approaches. The Adjusted Rand Index (ARI) for the re-rank model is 93.7 and for the bi-encoder model is 91.5, versus 73.7 for LSH.

While the primary advantage of hashing - and a central motivation for its frequent usage - is its scalability, massive scale similarity search (Johnson et al., 2019) is sufficiently cheap on modern GPUs to make neural de-duplication highly scalable. We use our contrastively-trained bi-encoder and a single NVIDIA 40GB A6000 GPU card to de-duplicate a 10 million document, 19 GB corpus in 11 hours and 45 minutes. While this cost is already marginal in the context of working with large text corpora, it could be reduced significantly further by using a lighter weight language model, as the majority of the time cost is embedding the 10M articles.

The large performance gains of our highly scalable neural approach do come at the cost of needing data to train the supervised bi-encoder. The publicly available NEWS-COPY dataset and our neural de-duplication models can be applied off-the-shelf or provide a starting point for using active learning to obtain training data for novel de-duplication problems.

The rest of this paper is organized as follows: Section 2 provides an overview of the de-duplication literature, as well as related problems in semantic textual similarity and retrieval. Section 3 describes the NEWS-COPY dataset, and Section 4 develops neural de-duplication methods and outlines their non-neural comparisons. Section 5 evaluates the performance of different de-duplication methods, and Section 6 explores scaling to a 10 million article corpus. Finally, Section 7 concludes.

2 Literature

2.1 De-Duplication

Textual de-duplication is a fundamental task for curating the large text corpora that support the deep learning revolution. Lee et al. (2021) review the de-duplication literature, providing evidence that duplication in training datasets is widespread: e.g. Dodge et al. (2021) find up to 14.4% of test examples of various standard benchmarks verbatim in C4 and Bandy & Vincent (2021) document that the Books Corpus (Zhu et al., 2015) - used in training BERT (Devlin et al., 2018), GPT (Brown

et al., 2020), and other large language models - contains 4,255 unique books and 2,930 books that are exactly duplicated at least once. When training LMs that are either very large or can consult an external database, test set leakage becomes a particularly salient concern (Borgeaud et al., 2022).

Non-neural methods predominate in textual de-duplication (Leskovec et al., 2020). Borgeaud et al. (2022) compute 13-gram Jaccard similarity between train and test documents using MinHashing and remove all training documents with 0.8 similarity or higher to validation/test documents. Radford et al. (2019) use 8-gram overlaps for post-hoc identification of duplication between GPT-2's training data and evaluation datasets, and Brown et al. (2020) remove from the GPT-3 training data any example with a 13-gram overlap with an evaluation example, documenting widespread contamination. They also de-duplicate each component dataset of the training data using MinHashing with 10 hashes. This reduces the training corpus size by 10%, underscoring that duplication is widespread. Other de-duplication contexts include large datasets of medical notes (Shenoy et al., 2017) and scholarly articles (which can include updates) (Gyawali et al., 2020), both of which have been examined with locally sensitive hashing.

Identifying reproduced texts within historical newspapers is itself an application that has generated considerable interest. The Viral Texts Project (Cordell, 2015; Smith et al., 2015) uses N-gram comparisons to track the dissemination of reproduced literature in antebellum newspapers. Viral Texts utilizes the Chronicling America (Culpepper, 2007) OCR of historical U.S. newspaper scans, which does not recognize individual articles, headlines, captions, etc., typically reading a multi-column newspaper page with many different text objects as if it were a single column book. This leads to scrambled up texts. We first apply object detection methods to the document layouts (He et al., 2017; Shen et al., 2021) to extract structured texts of individual articles. These structured texts then allow us to capture performance gains from the language understanding of neural methods.

Vesanto et al. (2017) use NCBI BLAST, a software for comparing and aligning biological sequences, to quantify text reproduction in Finish newspapers from 1771 to 1910. They remove all characters besides the 23 most common letters from an uncased corpus of Finish newspapers, and then convert these to the alphabet of 23 amino acids recognized by BLAST. BLAST is used to make pairwise comparisons between all documents in the corpus, indicating which pairs have text overlap. They average the start and ending of passages that overlap by at least 80% to get consensus passages, create a graph connecting consensus passages, and use a community detection algorithm to remove false positives. We likewise find that community detection can help control false positives. To scale the problem, we use hashing - which avoids the need to convert texts into amino acid sequences - or a contrastively trained bi-encoder - which leverages the power of deep learning.

2.2 Semantic Textual Similarity

There are important parallels between semantic textual similarity (STS) and textual de-duplication. Notably, our bi-encoder method draws inspiration from Sentence BERT (S-BERT) Reimers & Gurevych (2019), and we use an S-BERT pre-trained bi-encoder as our base language model. S-BERT adds a pooling operation to BERT/RoBERTa embeddings - that takes the mean of all output vectors - to derive a fixed sized sentence embedding that can then be examined with clustering methods.

While STS tasks are sometimes framed in terms of duplicates - such as detecting duplicate Quora questions - *semantic* duplicates are different than *near textual* duplicates. In STS tasks, the aim is to compute how similar texts are in meaning even when their wordings are very different, whereas the textual de-duplication literature aims to detect texts that are worded similarly. We show that an off-the-shelf S-BERT STS model significantly underperforms our neural de-duplication methods - as expected given that S-BERT is not intended for textual de-duplication - although interestingly it performs in the same ballpark as N-gram methods. If our goal was solely to identify exact duplicates, neural methods would not be needed. However, the presence of significant noise makes textual de-duplication a more challenging problem than it might appear at first glance.

2.3 Retrieval

We draw inspiration for our reranking approach from the literature on open domain retrieval and question answering (Wang et al., 2018; Lin et al., 2018; Karpukhin et al., 2020; Thakur et al., 2021;

Wu et al., 2019). This literature avoids the infeasible quadratic cost of applying a cross-encoder to a massive corpus by first ranking documents with a bi-encoder (or with sparse methods). Each document representation needs to be computed only once and can be stored offline, for comparison to any query vector. The passages that are closest to a given query are then passed to a cross-encoder, which jointly embeds the passage and query in order to compute cross-attention between them. In our re-ranking model, instead of a passage encoder and a query encoder, there is a symmetric bi-encoder. Like Karpukhin et al. (2020), we use Facebook AI Similarity Search (FAISS) (Johnson et al., 2019) to select nearby document pairs to evaluate with cross-attention.

3 THE NEWS-COPY DATASET

3.1 REPRODUCTION IN NEWS

Reproduction is an important feature of news. News wire services distribute stories written by their own news bureaus and by member newspapers to member news outlets, whereas syndicates disseminate to their subscribers columns written by freelance journalists or purchased from newspapers. The nation's largest newspapers also ran syndicate services to redistribute their own stories. The main news wire services in the United States historically were the Associate Press (AP), the United Press (UP), and William Randolph Hearst's International News Service (INS), the latter two of which merged to form United Press International (UPI) in 1958. Angelucci et al. (2020) document that in the 1950s, about two thirds of U.S. daily newspapers subscribed to the AP, 45% to the UP, and 15% to the INS (papers could subscribe to more than one wire).

Editing could take place at multiple places along the news wire transmission chain. News wire articles were written by local members, by wire staff reporters, and by foreign news services. Wire staff verified and edited stories after receiving them from members, and then stories could be edited again by local wire bureaus, of which there were around 100 for the Associated Press. Finally, local newspapers could further abridge content to fit space requirements. This leads to a range of near-duplicates in the presence of abridgement. For evaluating N-gram methods, we define similarity such that abridgements and truncations are not punished.

3.2 DESCRIPTION OF THE NEWS-COPY DATASET

Table 1 summarizes the key features of the NEWS-COPY dataset. It consists of 27,203 articles, drawn from 973 newspapers between 1920 to 1977.[1] NEWS-COPY contains two types of data: data for training and four full day exhaustively labeled evaluation samples, constructed with two consecutive days of content 1930 and single days in 1955 and 1974, selected at random. The 1955 sample is a validation set used to select hyperparemters for both the N-gram and neural methods. 1930 and 1974 are pooled to form the test set and used only to produce the results shown in this paper. In the full day samples, there are far more negative than positive pairs, as is generally the case in de-duplication problems, whereas the training data contain a more balanced sample for model training.

3.3 PROCEDURE FOR BUILDING THE DATASET

To build NEWS-COPY, we first apply Layout Parser (Shen et al., 2021) with a custom-trained object detection model (He et al., 2017) to front page scans of off-copyright historical newspapers to identify individual article bounding boxes. The contents of article bounding boxes are OCR'ed with Tesseract. When component bounding boxes span multiple columns on the same page, the OCR'ed texts are associated into full articles using a rule-based association method that exploits the coordinates of headline and article bounding boxes. This pipeline extracts the structured article texts. Headlines were chosen by local newspapers - not wires - and as a result are rarely reproduced and not included in the dataset. Weather forecasts are removed by running a distil-RoBERTa classifier trained on 392 labeled articles (179 positive, 202 negative). We also hand-removed documents containing incorrectly merged article bounding boxes from different underlying source articles (as there was no single ground truth cluster to which these articles belonged), and news summaries, which

[1] A copyright law change effective January 1, 1978 resulted in nearly all newspapers from that date forward being under copyright by default.

	Positives Pairs	Negative Pairs	Reproduced Articles	Singleton Articles	Total Articles
Training Data					
Training	36,291	37,637	891	–	7,728
Validation	3,042	3,246	20	–	283
Full Day Evaluation					
Validation	28,547	12,409,031	447	2,162	4,988
Test	54,996	100,914,159	1,236	8,046	14,211
Full Dataset	122,876	113,364,073	2,594	10,208	27,203

Table 1: This table provides summary statistics from the NEWS-COPY dataset, decomposed into the training sample and the full day evaluation data.

summarize multiple news stories in a single article and hence also have no clear cluster with which they are associated.

Duplicates are defined as articles that came from the same underlying news wire or syndicate source, regardless of the degree of abridgement or OCR noise. Articles from different sources that contain the same quote are labeled as non-duplicated, as are articles about the same event from different news wire sources. Likewise, articles updated to reflect breaking news are labeled as different.

To construct the full-day samples, we first ran 5-gram overlap with a very conservative N-gram overlap threshold of 1% to create large candidate duplicate clusters. Highly trained student research assistants carefully reviewed these clusters, breaking false positive links. Congruence labeling between two annotators was used to ensure that annotators understood the definition of a duplicated article. Next, annotators reviewed each of the resulting clusters to merge together clusters as needed. Finally, annotators exhaustively reviewed every singleton article, associating them with article clusters as needed. Articles were sorted by byline (recognized with a custom-trained named entity recognition model) to facilitate this process. For building the training data, the approach was similar, which provides hard negatives. We did not review all singletons, as the aim was to produce labeled batches for constrastive training. About two thirds of the negative pairs in the training data are hard negatives, with the remaining third coming from randomly selected article pairs.

4 Model Architectures

4.1 The Bi-Encoder Model

We contrastively train a symmetric bi-encoder to learn similar representations for near duplicate articles and dissimilar representations for non-duplicated articles. We use an S-BERT MPNET model (Reimers & Gurevych, 2019; Song et al., 2020) contrastively trained on over a billion sentence pairs - drawn from STS datasets - as the base language model. The S-BERT architecture pools representations for up to the first 512 tokens in each article, using mean pooling, to construct a document level representation. Like Reimers & Gurevych (2019), we found when experimenting with vanilla RoBERTa embeddings - which also perform well on de-duplication - that mean pooling of each of the representations significantly outperforms using the [CLS] token to represent the document. S-BERT provides a speed-optimized implementation of this pooling strategy. We chose the MPNET S-BERT because it performs best overall on STS benchmarks.

We use S-BERT's online contrastive loss (Hadsell et al., 2006) implementation, with a 0.2 margin and cosine similarity distance. The learning rate is 2e-5 with 100% warm up and a batch size of 32. We use an AdamW optimizer, and the model is trained for 16 epochs.

The bi-encoder dense document representations can be clustered to identify duplicates. We use FAISS (Johnson et al., 2019) to compute all embeddings within a given distance range, a hyperparameter tuned on the full-day validation sample. This output is used to build a graph, where nodes are articles and edges connect articles within the threshold distance. Connected components can be extracted to define clusters - which is equivalent to single linkage clustering - or Louvain commu-

nity detection can be applied to the graph to control false positive edges that can merge otherwise disparate groups of articles.

4.2 THE RE-RANKING MODEL

While a cross-encoder can offer the most flexible, expressive comparisons between texts, it requires N^2 embeddings to compare N texts. In the 10M article corpus explored in Section 6, a cross-encoder would require 10^{14} embeddings, well beyond the feasibility of current technology.

To make the use of a cross-encoder feasible, we draw inspiration from the retrieval literature (Wang et al., 2018; Lin et al., 2018; Karpukhin et al., 2020; Thakur et al., 2021; Wu et al., 2019) by first ranking document similarity with a bi-encoder, and then passing only the most similarly ranked documents to a cross-encoder. This approach, while not as cheap as simply clustering the bi-encoder embeddings, can still scale to a significant extent. We test whether it offers additional performance gains over using a bi-encoder alone.

For the baseline re-ranking model, we choose a threshold for the bi-encoder outputs of 0.92, optimized using the one-day validation sample. We use RoBERTa-base (Liu et al., 2019) as the base language model, with a 2e-5 learning rate and an AdamW optimizer. It is trained for 5 epochs with 20% warmup and a batch size of 32.

4.3 N-GRAM METHODS

We explore two different N-gram methods, one which relies on full computation of N-gram overlaps as well as locality-sensitive hashing (LSH), or fingerprinting. For both methods, we first shingle each article into N-grams, where $N \in \{3, 4, 5, 10, 15\}$. Uncommon punctuation marks are removed from the text of the article during pre-processing.

While traditional N-gram approaches to de-duplication often rely on a computation of the Jaccard similarity between two documents, defined for two documents A, B as $\frac{|A \cap B|}{|A \cup B|}$, we modify this metric of similarity for our use case, instead defining the "overlap" between two documents as $\frac{|A \cap B|}{\min(|A|,|B|)}$. This has the desirable property that abridgements and truncations, common in `NEWS-COPY`, are not penalized. If A is a substring of B, the documents will have an overlap of 1; this is not the case with Jaccard similarity.

For the N-gram overlap method, we compute overlaps between each pair of articles, drawing an edge between two articles if the overlap is above some minimum overlap threshold. This hyperparameter, which determines the accuracy of N-gram de-duplication, is tuned on the full-day validation sample. We also choose N using the full-day validation sample, with $N = 3$ providing the best Adjusted Rand Index. Once all edges have been drawn, Louvain community detection can be applied to remove false positives that erroneously connect otherwise disjoint document clusters, a step that we show improves the performance of this method.

For LSH, we use Datasketch's MinHashLSH library, which is widely used in the literature. The two relevant hyperparameters are the number of hash functions used and the minimum number of hash collisions needed to designate two documents as duplicates. For the former, we choose 10, which predominates in the literature, balancing computational efficiency and accuracy. The latter is tuned on the full-day validation sample. All documents with at least the threshold number of hash collisions are defined as duplicates. As with the other methods, community detection can be run once the graph has been constructed.

5 MODEL EVALUATION

5.1 BASELINE RESULTS

Table 2 compares the accuracy of 4 baseline models on the full-day test samples: non-neural methods (N-gram overlap and hashing) and neural methods (bi-encoder and re-ranking). For both of these, we examine a scalable method (hashing and bi-encoder) and a method that aims to achieve higher accuracy at some computational cost (N-gram overlap and re-ranking). We do not evaluate the cross-

	Neural	Non-Neural
Most scalable	Bi-encoder (91.5)	LSH (73.7)
Less scalable	Re-ranking (**93.7**)	N-gram overlap (44.0)

Table 2: The numbers in parentheses are the Adjusted Rand Index for four different models - a bi-encoder, a "re-ranking" strategy that combines a bi- and cross-encoder, locally sensitive hashing (LSH), and N-gram overlap. Hyperparameters were chosen on the NEWS-COPY validation set, and all models were evaluated on the NEWS-COPY test set.

encoder alone, as the scale of the majority of de-duplication applications makes it computationally infeasible (i.e. 10^{14} embeddings would be required for our scaling exercise).

Table 2 reports the adjusted Rand index (ARI) (Hubert & Arabie, 1985).[2] The neural methods significantly outperform the N-gram based methods, increasing ARI from 73.7 to 91.5 when comparing the most scalable methods (LSH and bi-encoder) and from 44.0 to 93.7 when comparing the more computationally costly methods (N-gram overlap and re-ranking).

In contrast, the more computationally intensive methods offer little advantage over their more scalable counterparts. In fact, hashing significantly outperforms N-gram overlap on the test data. The thresholds to determine a duplicate (overlap/collisions) were chosen using the validation sample, where N-grams indeed outperform hashing. Defining what a duplicate is using a single threshold makes N-gram based methods brittle, even in contexts such as this one where distribution shift is likely modest. The optimal N-gram overlap threshold in the validation sample leads to quite poor performance in the test sample. The 're-ranking' strategy does offer modest gains over using a bi-encoder alone (ARI of 93.7 vs. 91.5), making it a compelling method for moderately size datasets where accuracy is paramount.

These results underscore the potential returns of applying deep neural models to de-duplication of massive text datasets. When neural false positives occur, it is typically in the context of articles that have some repeated content but do not meet the definition of duplicates used to create the NEWS-COPY dataset: i.e. the articles contain the same quote or one is an updated article that contains additional breaking news. False negatives are most likely to occur when the beginning of an article has been truncated due to an error in the data digitization pipeline. If desired, further improvements in accuracy could plausibly be achieved by adding more of these challenging edge cases to the training data.

With N-gram methods, errors are mechanical, occurring in articles where noise results in fewer N-grams in common or where distinct articles by chance have significant overlap. N-gram overlap and hashing control this tradeoff through a single threshold, whereas neural methods are highly flexible and can be finely tuned to the downstream task through curation of appropriate training data.

5.2 Ablations

Tables 3 and 4 explore several sets of ablations to the neural and non-neural methods, respectively. Community detection can detect false positive links between otherwise disparate clusters. Removing it makes little to no difference for the neural models but leads to a significant decline in performance for N-gram overlap and hashing. This underscores the greater robustness of the neural approaches, whereas the non-neural methods are more prone to false positives that link otherwise disparate groups of articles.

Next, we consider whether hierarchical agglomerative clustering (HAC), which does not scale well, improves the accuracy of the bi-encoder approach. There are modest gains, with ARI increasing from 91.5 to 92.5, which highlights the potential returns of using HAC for moderately-sized de-duplication problems.

We also consider off-the-shelf S-BERT (Reimers & Gurevych, 2019), a model trained to detect semantic textual similarity (STS). STS and de-duplication are distinct problems - despite STS being framed sometimes as the removal of (semantic) duplicates. As expected, the models designed for STS perform inadequately on textual de-duplication (ARI 70 for the bi-encoder and 69.3 for

[2]Evaluating with F1 leads to similar conclusions.

	Baseline	No commun. Detection	HAC Clustering	S-BERT STS	Loss Function SupCon	MNRL
Bi-encoder	91.5	91.5	**92.5**	70.0	87.2	84.5
Re-ranking	**93.7**	92.0	-	69.3	91.1	89.9

Table 3: The first column removes community detection, and the second column uses hierarchical agglomerative clustering for the bi-encoder. *S-BERT STS* uses off-the-shelf S-BERT semantic textual similarity models. *SupCon* uses a supervised contrastive loss function and *MNRL* uses a multiple negatives ranking loss function.

re-ranking), though are not that much worse than LSH despite not being trained for this task. Interestingly, this suggests that our pre-trained `NEWS-SIM` model could potentially outperform LSH for text de-duplication even on quite distinct corpora.

For the neural methods, we also explore different loss functions for contrastively training the bi-encoder - supervised contrastive (SupCon) loss (Khosla et al., 2020)[3] and multiple negatives ranking loss (MNRL) (Henderson et al., 2017).[4] The online contrastive loss outperforms both of these alternative losses in both neural approaches.

For the N-gram based methods, we chose the optimal $N = 3$ on the one-day validation sample. Table 4 varies N - with the overlap threshold/number of collisions for each N again chosen to maximize ARI in the one-day validation sample. For LSH, $N = 3$ remains optimal in the test set, whereas for N-gram overlap, it performs quite poorly, with $N = 5$ having an edge. This again demonstrates that N-gram methods can be brittle.

	Baseline	No commun. detection	4-grams	5-grams	10-grams	15-grams
Hashing	**73.7**	67.7	71.5	69.6	63.7	59.4
N-gram overlap	44.0	23.2	57.9	**59.2**	56.6	57.4

Table 4: Numbers report the Adjusted Rand Index. *No community detection* removes community detection. The other columns vary the N used to compute N-grams and include community detection.

6 Computational Efficiency

Hashing is often advocated because of its scalability, and neural methods for de-duplicating large text corpora likewise need to be highly scalable. We conduct experiments on a 19 GB, 10 million article corpus, created by applying the same object detection model used to curate `NEWS-COPY` to millions of front page newspaper page scans. These experiments use a 32-Core 3.50 GHz AMD RYZEN Threadripper Pro 3975WX and a single NVIDIA A6000 GPU, a very modest setup for working with large text corpora.

We scale the bi-encoder and LSH approaches to this corpus, reporting speeds in Table 5. The largest cost of scaling the bi-encoder, by a wide margin, is embedding the 10M articles, which takes 8:38:52 on a single GPU. This could be sped up significantly by deploying a smaller language model. However, since this cost is already fairly marginal in the context of working with massive text datasets, we do not explore this here.

FAISS (Johnson et al., 2019), with `IndexFlatIP`, is used to calculate pairwise exact distances between the embeddings.[5] These computations require just over 3 hours on a single GPU. This

[3]Trained for 32 epochs, 67% warm up, batch size 32, AdamW optimizer, 2e-5 learning rate

[4]Trained for 32 epochs, 0% warmup, batch size 16, AdamW optimizer, 2e-5 learning rate

[5]Because FAISS range search is not GPU-supported, we implement k nearest neighbor search, conservatively setting k to 900. Then distances are filtered to those below the optimal threshold found in our one day validation sample. In NEWS-COPY, articles are never reproduced more than 200 times, with the average reproduced article appearing 6.3 times, indicating that $k = 900$ is quite conservative. Similarity search could be sped up by choosing a smaller k.

	Embed Articles	Compute Similarity	Build Graph	Commun. Detect.	Total Time	Articles per Duplic. Cluster
Bi-Encoder	8:38:52 (GPU)	3:04:53 (GPU)	0:01:23 (GPU)	0:00:02 (GPU)	11:45:10 (GPU)	6.41
Hashing		3:39:05 (CPU)	0:00:55 (GPU)	0:00:08 (GPU)	3:40:08 (mostly CPU)	11.55

Table 5: This table reports computational efficiency in scaling the bi-encoder and LSH methods to a 10 million article corpus. Parentheses indicate whether the calculations were run on a CPU or a single NVIDIA A6000 GPU card. *Articles per duplicate cluster* reports the average number of times reproduced articles appear in the corpus from applying the method at scale.

produces a list of article pairs whose embeddings are within a threshold distance, using the optimal threshold selected in the NEWS-COPY full day validation sample. We build a graph - where each of the 10M documents is a node and edges connect documents whose embeddings are within the threshold distance. This takes 0:01:23 using `CuDF` and `CuGraph`. False positives are controlled by running Louvain community detection on the graph; alternatively, one could define clusters simply by extracting the connected components. Total run-time on the single GPU card is 11:45:10.

While neural methods are not expected to be faster than hashing, they compare reasonably. LSH requires 3:39:05 CPU hours for pre-processing, shingling, and hashing, and around 1 GPU minute to create the resulting graph and apply community detection. Commonly used hashing libraries run on a CPU, and hence this is where we implement LSH, with Datasketch's MinHashLSH. To effectively scale LSH, we slightly modify the previous architecture and break the hashes into bands comprised of rows of hashes. Each of these bands is hashed again and two articles that share an identical band hash are considered duplicates. The choice of bands and rows determines an S-curve, which gives the probability of being considered a duplicate given a certain Jaccard similarity. To generate our desired S-curve, we used 15 bands and 2 rows.

The bi-encoder method detects 810,080 distinct reproduced articles. The average duplicated article was reproduced 6.41 times, strikingly similar to what we document in the single day `NEWS-COPY` labeled data. This is expected, since most reproduced articles occur within the time window captured in the `NEWS-COPY` full day samples. The number of true negative pairs is massively greater in the 10M article corpus, and it is encouraging that the average number of times articles are estimated to be reproduced remains quite constant, rather than being blown up by false positives. In contrast, LSH detects only 486,460 distinct reproduced articles, estimating that each on average appears 11.55 times. This is significantly greater than in the labeled `NEWS-COPY` data, and likely indicates that false positives are significantly increasing the size of detected duplicated article groups.

7 Conclusions

De-duplication is important to a range of NLP tasks, from training large language models, controlling privacy risk, and controlling test set leakage, to detecting reproduced content in large text corpora for social science and humanities research. N-gram based methods predominate, and most contexts do not provide a way to assess how well they perform since duplicates in large corpora cannot typically be comprehensively identified. We exploit the timely nature of news to build `NEWS-COPY`, a large, human-annotated dataset that comprehensively identifies reproduced news articles.

`NEWS-COPY` is then used to evaluate neural and non-neural de-duplication. Neural methods - which utilize a contrastively trained bi-encoder - offer significant performance gains, and can be scaled on a single GPU card to de-duplicate a 10M article corpus in a matter of hours. These results suggest that neural text de-duplication is straightforward and well worth further examination in a range of contexts where hashing and other N-gram based methods currently predominate. The public release of `NEWS-COPY` and this study's codebase aims to facilitate further work on robust, scalable text de-duplication.

REFERENCES

Charles Angelucci, Julia Cagé, and Michael Sinkinson. Media competition and news diets. Technical report, National Bureau of Economic Research, 2020.

Jack Bandy and Nicholas Vincent. Addressing "documentation debt" in machine learning research: A retrospective datasheet for bookcorpus. *arXiv preprint arXiv:2105.05241*, 2021.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pp. 2206–2240. PMLR, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ryan Cordell. Reprinting, circulation, and the network author in antebellum newspapers. *American Literary History*, 27(3):417–445, 2015.

Jetta Culpepper. Chronicling america: Historic american newspapers. *Reference Reviews*, 2007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.

Julia Guarneri. *Newsprint Metropolis*. University of Chicago Press, 2017.

Bikash Gyawali, Lucas Anastasiou, and Petr Knoth. Deduplication of scholarly documents using locality sensitive hashing and word embeddings. 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. *arXiv preprint arXiv:2202.06539*, 2022.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1736–1745, 2018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. Layoutparser: A unified toolkit for deep learning based document image analysis. In *International Conference on Document Analysis and Recognition*, pp. 131–146. Springer, 2021.

Sanjeev Shenoy, Tsung-Ting Kuo, Rodney Gabriel, Julian McAuley, and Chun-Nan Hsu. Deduplication in a massive clinical note dataset. *arXiv preprint arXiv:1704.05617*, 2017.

David A Smith, Ryan Cordell, and Abby Mullen. Computational methods for uncovering reprinted texts in antebellum newspapers. *American Literary History*, 27(3):E1–E15, 2015.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33: 16857–16867, 2020.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*, 2021.

Aleksi Vesanto, Filip Ginter, Hannu Salmi, Asko Nivala, and Tapio Salakoski. A system for identifying and exploring text repetition in large historical document corpora. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pp. 330–333, 2017.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.