Chuweb21D: A Deduped English Document Collection for Web Search Tasks

Zhumin Chu
Quan Cheng Laboratory,
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing, China
chuzm19@mails.tsinghua.edu.cn

Tetsuya Sakai
Waseda University
Tokyo, Japan
tetsuyasakai@acm.org

Qingyao Ai
Quan Cheng Laboratory,
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing, China
aiqy@tsinghua.edu.cn

Yiqun Liu*
Quan Cheng Laboratory,
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

As a traditional information retrieval task, ad hoc web search has long been an important part of IR research and evaluation tracks (e.g. TREC, NTCIR and CLEF). A crawled, large-scale web document collection is a central component to offline web search evaluation. Although there already exist several English document collections, such as the ClueWeb series, GOV2 and c4, a collection that satisfies properties of both strong timeliness and raw HTML formatting is still relatively scarce.

To better support the demands of nascent web search tasks, we have built and publicly released Chuweb21D, a large-scale deduped English document collection for web search tasks. The Chuweb21D collection is derived from Chuweb21, which we released in April 2021 as a target corpus for the NTCIR-16 WWW-4 Task. We applied two different deduping thresholds to obtain two versions of Chuweb21D, called Chuweb21D-60 and Chuweb21D-70; the former is used as the target corpus for the ongoing NTCIR-17 FairWeb-1 task. To gain an insight into the impact of deduping, we evaluate the runs submitted to the NTCIR-16 WWW-4 task using Chuweb21D, and compare the outcome with the official results that used the corpus before deduping.

CCS CONCEPTS

• **Information systems → Document filtering**.

KEYWORDS

Document Collection, Ad hoc Web Search, Document Duplication

*Corresponding author

SIGIR-AP '23, November 26–28, 2023, Beijing, China
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0408-6/23/11.
https://doi.org/10.1145/3624918.3625317

ACM Reference Format:
Zhumin Chu, Tetsuya Sakai, Qingyao Ai, and Yiqun Liu. 2023. Chuweb21D: A Deduped English Document Collection for Web Search Tasks . In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23), November 26–28, 2023, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3624918.3625317

1 INTRODUCTION

Ad hoc retrieval [29] is the fundamental task in information retrieval: given a static target document collection, it requires a system to return a ranked document list for each topic in the test topic set. In particular, ad hoc web search has long attracted much attention from research efforts and evaluation conferences such as TREC [29], NTCIR [26], and CLEF [11], in order to study and improve web search quality in offline environments. A crawled, large-scale web document collection is a core component in such evaluation tasks. Furthermore, in recent years, dense retrieval models are placing greater demands on the scale and quality of training document collections, resulting in more traffic of attention to large-scale document collections.

Although there already exist several popular document collections in the field of English web search, such as the ClueWeb series [5, 6, 22], GOV2 [9] and c4 [24] datasets, the large-scale timely raw-HTML English document collections are still scarce. Table 1 shows the basic information about some popular English document collections. GOV2, ClueWeb09 and ClueWeb12, as representatives of previous generation document collections, comprise HTML documents that were created over ten years ago. MS MARCO Documents and Tensorflow c4 datasets, as representatives of the new generation document collections, are frequently used for pre-training and finetuning in various dense retrieval models. To better serve the demands of model training, these two collections provide preprocessed plain text rather than raw HTML documents, and are not necessarily suitable as target corpora for web search tasks. ClueWeb22, as a recently released large-scale English document collection, features raw HTML of web pages and provides better support for ad-hoc web search tasks. However, obtaining ClueWeb22 requires a fee for hard disk shipping; we provide an alternative that can be downloaded online for free.

This paper introduces the Chuweb21D English web document collection, a deduped version of Chuweb21 that was used as the target corpus for the NTCIR-16 WWW-4 task [27, 28]. Our motivation for deduping Chuweb21 is based on our experience as the organisers of the NTCIR task: when conducting relevance assessments of the pooled documents for the task, we witnessed considerable amount of duplicate and near-duplicate web pages, which can potentially cause problems. More specifically, these duplicates will exhaust the limited evaluation resources (i.e. hinders the identification of relevant documents outside the pool files), and thereby may introduce evaluation bias (i.e., underestimate runs that return relevant documents that we failed to identify, and overestimate runs that return many duplicate relevant documents). For deduping, we employ the Simhash strategy [8, 19] with two different clustering thresholds, and release two versions of Chuweb21D; the smaller collection is used as the target corpus for the ongoing NTCIR-17 FairWeb-1 (Group-Fair Web Search) task [1]. To gain an insight into the impact of deduping, we evaluate the runs submitted to the NTCIR-16 WWW-4 task using Chuweb21D, and compare the outcome with the official results that used the corpus before deduping.

Our contributions can be summarised as follows.

- We publicly release the Chuweb21D collection, a large-scale deduped English document collection for web search tasks. This collection is already available online for free [2]. Anyone can download the data via disk, or contact us to obtain it through other means (SCP or hard disk shipping).
- We detail the construction process of the Chuweb21 and Chuweb21D collections, which will probably be a useful reference for other researchers working on web corpus construction.
- We clarify the impact of deduping on the results of the NTCIR-16 WWW-4 task, which relied on our corpus before deduping.

The remainder of this paper is organised as follows. we first introduce the related work (§2) on large-scale English document collections and document deduplication technologies, and then we introduce the construction procedures of Chuweb21 (§3) and Chuweb21D (§4) datasets in detail. Next, we compare the Chuweb21 and Chuweb21D collections (§5) in terms of statistics and WWW-4 systems ranking. Finally, we conclude this paper (§6).

2 RELATED WORK

2.1 Large-Scale English Document Collection

In Table 1, we have presented some general information about the existing large-scale English datasets. In this section, we will introduce them more specifically.

As a TREC test collection, the GOV2 dataset [9] was first applied in the TREC 2004 Terabyte Track [9]. As a crawl of the .gov websites, the majority of the web pages in the GOV2 collection belong to the .gov domain, thus naming it the GOV2 dataset.

Based on bing's search logs, Bajaj et al. [20] collected millions of questions and a large number of associated documents and passages, called the MS MARCO dataset. The MS MARCO dataset was originally designed for machine reading comprehension tasks. Its large scale of documents and human-annotated data makes the MS MARCO dataset frequently used for the pre-training and fine-tuning of dense retrieval models [21, 23].

In 2020, Raffel et al. [24] first presented the Colossal Clean Crawled Corpus (c4) dataset and utilized it to explore the limits of the transfer learning approach. Similar to our Chuweb21 dataset, the c4 dataset has extracted some of the web pages from the Common Crawl dataset (April 2019 block), and carried out data cleaning jobs based on it. Unfortunately, only the pre-processed text content of the documents is retained in the c4 dataset. The creators also integrated the c4 dataset into the TensorFlow package. Although these operations facilitate the pre-training of dense retrieval models, it is not user-friendly for ad hoc web search tasks. Another similar work is CC-News-En [17], which focuses on English news documents. CC-News-En was constructed based on a news-specific crawl announced by the Common Crawl Foundation in late 2016. After a series of data cleaning process, including language filtering and identifier augmenting, CC-News-En contains about 44 million English news documents crawled collected between September 2016 and March 2018.

Another popular set of English documents is the ClueWeb series of data collections [5, 6, 22], which were all created by the Lemur Project. Compared to other popular large-scale English document data collections, one of the major features of the ClueWeb series collections is the much larger scale of the data. Each version of the ClueWeb collection contains billion-level documents, especially the latest ClueWeb22 collection even containing 10 billion documents. Another advantage of the ClueWeb series collections is that they provide raw HTML data, which is of great value. However, they only support hard disk shipping, requiring users to pay a not-so-cheap fee to access the ClueWeb collections.

2.2 Document Deduplication Technology

To address the near-duplicate document problem, researchers have proposed several document de-duplication approaches. As a plain idea, the K-shingle algorithm [18] measures the similarity of documents by representing them as a set of K-shingles (i.e., all substrings of length K in the documents) and calculating the Jaccard similarity between the sets of K-shingles of each document pair. The current mainstream document de-duplication algorithms belong to Local Sensitive Hash (LSH) technology [15]. Unlike general hashing algorithms, LSH maps similar texts to neighboring hash codes as possible, while preserving the fundamental characteristics of the hashing algorithm. Typical LSH algorithms comprise the Minhash algorithm [3] and Simhash algorithm [8, 19]. The Minhash algorithm was first applied to the AltaVista search engine to remove duplicate documents. The Minhash algorithm designs a minimum hash function to transform the raw ultra-high-dimensional sparse vector generated by the K-shingle algorithm into a low-dimensional dense vector, which reduces the space complexity. In addition, it also indexes the transformed dense vectors by segments to narrow down the scope of potentially similar texts, so as to reduce the time

[1]http://sakailab.com/fairweb1/
[2]https://github.com/chuzhumin98/Chuweb21D

Table 1: Comparisons between some popular English document collections and our Chuweb21(B) datasets

Dataset Name	#Documents	Space	Collected time	Format
TREC GOV2 [9]	25M	80GB	Early 2004	raw html
MS MARCO Documents [20]	3.2M	22GB	Before January 2017	text (title; body)
Tensorflow c4 [24]	/	750GB	April 2019	text
CC-News-En [17]	44M	991GB	September 2016 ~ March 2018	raw html
ClueWeb09 [6]	1.04B	5TB	January 2009 ~ February 2009	raw html
ClueWeb12 [5]	733M	5.54TB	February 2012 ~ May 2012	raw html
ClueWeb22 [22]	10B	/	Before August 2022	raw html
Chuweb21 (ours)	82.5M	1.7TB	April 2021	raw html
Chuweb21D-60 (ours)	49.8M	1.2TB	April 2021	raw html
Chuweb21D-70 (ours)	57.9M	964GB	April 2021	raw html

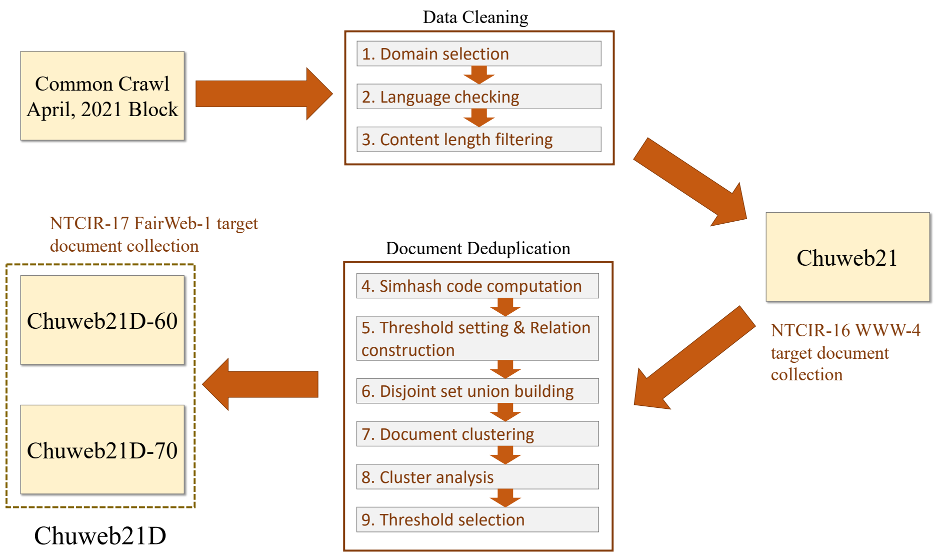

Common Crawl April, 2021 Block
Data Cleaning
1. Domain selection
2. Language checking
3. Content length filtering
Chuweb21
NTCIR-16 WWW-4 target document collection
Document Deduplication
4. Simhash code computation
5. Threshold setting & Relation construction
6. Disjoint set union building
7. Document clustering
8. Cluster analysis
9. Threshold selection
NTCIR-17 FairWeb-1 target document collection
Chuweb21D-60
Chuweb21D-70
Chuweb21D

Figure 1: The construction procedures of the Chuweb21 and Chuweb21D collections

complexity. In a recent work, Lee et al. [16] adopted the Minhash algorithm to deduplicate the C4 dataset and discovered a 61-word sentence that appeared over 60,000 times. To tackle the enormous document scale, the Google team further proposed the Simhash algorithm, which adopts keywords and their word frequencies as features for binarized hash calculation, and then accumulates all the hash values to obtain the final 64-bit or 128-bit hash value. The similarity between documents is represented by the Hamming distance between their hash codes.

Bernstein et al. [2] investigated the influence of redundant documents on search effectiveness. They introduced the "novelty principle", which states that a document – though relevant in isolation – should not be seen as relevant if it duplicates a document the user has already browsed. A subsequent study [13] evaluated this novelty principle on 11 benchmarks, observing that the overall system rankings do not change significantly in most cases. A recent work [12] adopted a combination of SimHash and canonical links to deduplicate and evaluate the impact on the ClueWebs and Common Crawls. Additionally, [12] suggested utilizing various deduplication strategies: for the corpus level, only very fast approaches like SimHash and canonical links are feasible; while for the submitted run files and the resulting pool, even expensive approaches are viable.

3 CONSTRUCTION OF CHUWEB21

Figure 1 shows the construction procedures of the Chuweb21 and Chuweb21D collections . As we know, the Chuweb21D collection is constructed based on our released Chuweb21 collection. So in this section, we will report the details for the construction of Chuweb21.

The Chuweb21 collection is constructed based on the 2021-17 (April, 2021) block of the Common Crawl dataset [3]. We choose Common Crawl as the cornerstone corpus mainly due to its timeliness and accessibility. As a non-profit project, Common Crawl has continuously released fresh data blocks since 2008 and is still ongoing today. For each block, its CCBot crawler continuously crawls web pages from the Internet over a period of time, resulting in a "snapshot" of the whole current collection of Internet documents. For example, our used 2021-17 block was crawled from 10th to

[3] https://commoncrawl.org/2021/04/april-2021-crawl-archive-now-available/

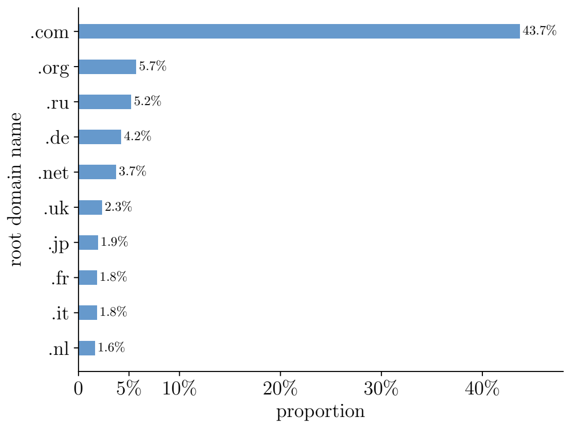


Figure 2: The proportion of the top-10 frequent root domains in terms of their numbers of web pages

23rd in April 2021, including about 3.1 billion web pages or 320 TiB of uncompressed content. Due to the limit of computation resources for downstream data cleaning and usage, we sample the web pages crawled within one day (2021-04-10 10:58:31 ∼ 2021-04-11 11:56:10) for the construction of Chuweb21. The original data of this part (denoted as original data in the subsequent content) contains 858, 616, 203 different web pages, occupying the space of 5.66 TiB.

We first conduct domain analysis for the original data. The original data contains 3, 402, 457 different domains. We group these web pages according to their root domains, Figure 2 shows the proportion of the top-10 frequent root domains in terms of their numbers of web pages. As the largest domain name, the .com domain accounts for half of the entire Internet data. Except for .com, the percentage of web pages under other domains does not exceed 6%. Among the top-10 frequent root domains, we find that except for .com, .org and .net, all other domains are region-specific, such as .de for Germany, .ru for Russia and .jp for Japan. To avoid the inclusion of much non-English content in the dataset, we strip out these regional domains and only retain the web pages under other popular domains (e.g. .com, .org and .net).

To ensure the quality of documents in the corpus, inspired by Raffel et al. [24], we apply the following restrictions to filter out useful web pages:

- The WARC-Type of the record must be "response" rather than "request".
- The main content of the document is presented in English. We remove the <script> and <style> tags and extract the body text of the web pages. Then, we use the "langdetect" [4] package to compute the probability that the body text of the document belongs to the English language. We exclude any pages with a probability less than 0.99.
- The character length of HTML content must be larger than 1, 000.

We deploy the foregoing data cleaning job on 4 servers, running 32 processes in parallel. The whole preprocessing job takes about two weeks. The final Chuweb21 collection contains 82, 451, 337 web pages (around 9.6% of the original data) or 1.69 TiB of compressed content. The complete Chuweb21 collection is split into 5, 118 subfiles with "warc.tar" format to facilitate the user in downloading and using.

[4]https://pypi.org/project/langdetect/

4 CONSTRUCTION OF CHUWEB21D

4.1 Motivations

The Chuweb21 collection was used as the evaluation dataset for the NTCIR-16 WWW-4 task, which was concluded in June 2022. As an ad hoc web search task, participants were required to design ranking models to generate retrieved ranking lists on the Chuweb21 collection for each topic in the given topic set. When we later carried out pooling document annotations, we found a number of duplicated document phenomena in the pooling set. These duplicate documents not only waste the limited computation and annotation resources, but also bias the evaluation results of the ranking systems. These issues motivate us to de-duplicate the Chuweb21 collection and release Chuweb21D.

4.2 Simhash code

We adopt the Simhash algorithm [8, 19] to de-duplicate the Chuweb21 collection. Unlike traditional hash algorithms (e.g. SHA-1 [4] and MD5 [25]), the simhash algorithm maps similar text to similar hash codes (i.e. a small Hamming distance). This property enables us to determine whether the contents of different documents are duplicated based on the simhash value with an appropriate discrimination threshold. Specifically, we adopt the "simhash" package [5] to compute 64-bit and 128-bit simhash codes for each HTML document in Chuweb21.

4.3 Duplicated documents clustering

In order to efficiently cluster duplicated documents, we use the disjoint set union [14] to organize the documents, reducing the time complexity to $O(M \log^* N)$. M and N represent the number of duplicate relations and documents, respectively. And $\log^*(x)$ is called Iterated logarithm function [10], which is even smaller than the function $\log(x)$. Using disjoint set union, we naturally decompose the document clustering task into the following three separated steps: (1) Relation construction, i.e., we need to generate a series of document pairs (d_1, d_2) to represent the duplicated relations between documents; (2) Disjoint set union building, i.e., we use the document pairs generated in the preceding step to construct a disjoint set union of documents; (3) Document clustering, i.e., we cluster documents based on the components of constructed disjoint set union.

Algorithm 1 shows the workflow for the relation construction step. Here we present a further elaboration for it. Notice that if we try to compute the whole set of document pair (d_1, d_2) to check whether they satisfy duplicated relation, the computation time complexity is $O(N^2)$, which is unaffordable. To improve the efficiency of relation construction, we propose different strategies under different conditions:

[5]https://github.com/1e0ng/simhash

Algorithm 1: Duplication Relation Construction

```
Input: The 64-bit and 128-bit simhash code $c_d$ and $\tilde{c}_d$ for
       each document $d$ in the Chuweb21 collection $\mathcal{D}$
Output: The duplication relation list, each item contains a
        document pair $(d_1, d_2)$
1  % parameter $\tilde{\tau}$ is only used for the 128-bit condition
2  RelationConstruct($\tau, \tilde{\tau}$) begin
3      Initialize the relation list $R = []$
4      if $\tilde{\tau}$ is None then
5          if $\tau \leq 3$ then
6              % Algorithm 2
7              RelationConstruct_64_less3($\tau, R$)
8          end
9          else
10             % Algorithm 3
11             RelationConstruct_64_great3($\tau, R$)
12         end
13     end
14     else
15         % Algorithm 4
16         RelationConstruct_128($\tau, \tilde{\tau}, R$)
17     end
18     return $R$
19 end
```

4.3.1 $\tau \leq 3$ with 64-bit simhash code. (Algorithm 2). τ denotes the threshold we use to determine whether documents are duplicated or not. When the Hamming distance between documents d_1 and d_2 is less than or equal to τ, we regard d_1 and d_2 as duplicated documents; otherwise, they are considered to be different documents. For each document d, we enumerate all the 64-bit simhash codes where we consider it as a duplicate with d, i.e., the Hamming distance to the simhash code of d does not exceed τ. Its time complexity for each document d is $O(\sum_{k=0}^{\tau} \binom{64}{k})$. Since we set $\tau \leq 3$, the total time complexity is upper-bounded to a tolerable overhead ($O(43745N)$).

4.3.2 $\tau > 3$ with 64-bit simhash code. (Algorithm 3). When $\tau > 3$, the brute-force enumeration approach is no longer feasible due to the overwhelming time cost. In practice, the value of τ is also not too large, because excessive tau values will make the rate of false positives much more uncontrollable. In our experiments, we bound τ up to 6 (we will present more details on the threshold selection in subsequent content).

Based on this restriction, we propose a randomized bucketing algorithm to accomplish the duplicated relation construction. Specifically, in each iteration, we randomly sample B indices from 64-bit codes to form the key of the bucket. Then, we bucket the documents by their keys. In this round of iteration, we only need to discriminate the duplication of document pairs within the same bucket, rather than all the document pairs. Due to the stochastic bucketing, the algorithm cannot guarantee that either duplicated document pair (d_1, d_2) is partitioned into the same bucket in every iteration. Instead, we can assure that d_1 and d_2 can be allocated to the same bucket in at least one iteration as long as we independently conduct enough iterations. More rigorously speaking, note that the probability that in each iteration d_1 and d_2 are allocated to the same bucket is $p_0 = \frac{\binom{64-h_{12}}{B}}{\binom{64}{B}} \geq \frac{\binom{64-\tau}{B}}{\binom{64}{B}}$, where h_{12} denotes the Hamming distance between the 64-bit simhash codes of d_1 and d_2. Thus, the probability that d_1 and d_2 are allocated to the same bucket in at least one iteration of the whole T iterations equals $p_T = 1 - (1 - p_0)^T$. In our experiments, we set B and T equal to 16 and 20, respectively. It is not hard to verify that $p_T > 97\%$ for any $\tau \leq 6$ under this setting.

In terms of efficiency, the randomized bucketing algorithm approximates the time complexity of $O(\frac{TN^2}{2^B})$ with a fairly uniform bucket partition. In real scenarios, of course, a uniform partition is extremely difficult to achieve, thus the actual time complexity is slightly higher than it, yet already affordable compared to $O(N^2)$.

4.3.3 $\tilde{\tau}$ with 128-bit simhash code based on the 64-bit τ. (Algorithm 4). Considering that 128-bit simhash codes lead to a greater computational overhead, we carry out the construction of 128-bit duplicated relations based on the relations extracted under 64-bit codes. Specifically, we first employ the 64-bit simhash codes to construct the candidate set of duplicated relations under the threshold τ. Afterward, we filter out those duplicated relations where the Hamming distance between the 128-bit codes of their document pairs is greater than $\tilde{\tau}$. Besides the high efficiency, another advantage of this algorithm is to reduce the risk of the high false positive rate due to the long tail effect (The duplicated relation still exists at a certain proportion when the Hamming distance of two document codes is relatively large, even up to 10 or more) of 128-bit code when $\tilde{\tau}$ is large. In our experiments, we set τ as 6, and $\tilde{\tau}$ not exceeding 15.

Based on the constructed duplicated relation, we organize the documents using a disjoint set union. In the disjoint set union, each group of duplicated documents would be organized on the same graph component. After completing the disjoint set union building, we can directly take each graph component as a duplicated document cluster.

4.4 Cluster Analysis & Threshold Selection

In section 4.3, we propose various strategies to cope with different threshold scenarios. In this section, we aim to pick the appropriate thresholds for the generation of the final data collection via the analysis of the clustering effect. The proportion of documents retained after deduping (i.e., documents retained), and the size of the largest duplicate-relation cluster (i.e., ratio of max cluster), as two critical indicators, play an important role when evaluating the performance of the thresholds. Table 2 and Figure 3 show the variation of these two indicators with threshold τ or $\tilde{\tau}$ under 64-bit and 128-bit simhash code settings, respectively.

In Table 2, we find a sharp rise in the scale of the largest cluster between the threshold τ equals to 3 and 4. This phenomenon is in line with our expectations. By sampling and examining some document pairs, we observed that when the Hamming distance between documents is not greater than 3, especially less than or equal to 2, most of the document pairs share the same content. In contrast, there are a fair amount of discrepancies, as well as some

Table 2: The comparison of both collection scale and the number of documents in the largest cluster for data collections constructed under different 64-bit thresholds τ

Threshold	#docs	| max cluster |	remained ratio	| max cluster |/total
original	82.5M	1	100.00%	0.00%
$\tau = 0$	73.0M	21.5K	88.55%	0.03%
$\tau = 1$	65.6M	43.5K	79.60%	0.05%
$\tau = 2$	57.9M	64.7K	70.18%	0.08%
$\tau = 3$	49.8M	82.1K	60.40%	0.10%
$\tau = 4$	37.3M	27.1M	45.21%	32.82%
$\tau = 5$	27.7M	43.1M	33.58%	52.29%
$\tau = 6$	18.3M	58.2M	22.10%	70.56%

Algorithm 2: Duplication Relation Construction for 64-bit $\tau \leq 3$ Condition

```
Input: The 64-bit simhash code $c_d$ for each document $d$ in
       the Chuweb21 collection $\mathcal{D}$, current relation list $R$
RelationConstruct_64_less3($\tau, R$) begin
    Construct a dictionary $S$, its key is each appeared 64-bit
      code $c$, the value of key $c$ is a document set $\{d|c_d = c\}$
    Initialize the valid operation scheme list $P = []$
    for $k \in 0...\tau$ do
        Enumerate all possible $k$-modified of 64-bit
          operations, append it to list $P$
    end
    % Each operation $p \in P$ can be acted on any 64-bit
      encode $c$ to generate a unique encode $c \oplus p$
    for $d \in \mathcal{D}, p \in P$ do
        $c' = c_d \oplus p$
        if $c' \in S$ then
            for $d' \in S[c']$ do
                $R$.append(($d, d'$))
            end
        end
    end
end
```

Algorithm 3: Duplication Relation Construction for 64-bit $\tau > 3$ Condition

```
Input: The 64-bit simhash code $c_d$ for each document $d$ in
       the Chuweb21 collection $\mathcal{D}$, current relation list $R$
RelationConstruct_64_great3($\tau, R$) begin
    Initialize hyper-parameter $B = 16, T = 20$
    for $t \in 1...T$ do
        Randomly sample $B$ indices from $Range(64)$ to form
          list $I_t$
        Initialize bucket $G = dict()$
        for $d \in \mathcal{D}$ do
            if $c_d[I_t] \notin G$ then
                Initialize $G[c_d[I_t]] = []$
            end
            $G[c_d[I_t]]$.append($d$)
        end
        for $b \in G$ do
            for $d_1, d_2 \in G[b]$ do
                if $HammingDistance(c_{d_1}, c_{d_2}) \leq \tau$ then
                    $R$.append(($d_1, d_2$))
                end
            end
        end
    end
end
```

consistent cases, when the Hamming distance ranges from 4 to 6. These redundant relations connect different clusters of duplicated documents together and form a mega-cluster, e.g. one third of the whole document set at $\tau = 4$! The formation of mega-cluster prevents the usage of data collection directly constructed under the threshold of $\tau \geq 4$. Setting τ to 2 or 3 becomes the favored option at 64-bit.

As for the 128-bit condition, Figure 3 shows the performance of different thresholds $\tilde{\tau}$ based on 64-bit $\tau = 6$. As the threshold $\tilde{\tau}$ increases, the retained candidate set of relations gradually increases, resulting in a larger scale of the largest cluster and a diminishing proportion of remained documents after de-duplication. $\tilde{\tau} = 10$, as a decent threshold for balancing the document retention ratio (61.17%) and the largest cluster scale (0.96 million documents), is a reasonable option for generating data collections. However, due to its document scale being too close to the version of $\tau = 3$ at 64-bit (61.17% v.s. 60.40%), this scheme is eventually discarded. Hence, we finally choose to construct the data collections using $\tau = 2$ and 3 under 64-bit, and name them by their document retention rate, i.e. ChuwebD-70 ($\tau = 2$) and ChuwebD-60 ($\tau = 3$).

5 CHUWEB21 V.S. CHUWEB21D

In this section, we conduct a series of analyses on the Chuweb21 and Chuweb21D datasets, to observe the impact of the de-duplication procedure.

5.1 Statistical Analysis

Table 3 shows the statistics of Chuweb21 and Chuweb21D collections. As we can observe, the de-duplication operation significantly reduces the document scale and storage space, which facilitates

Algorithm 4: Duplication Relation Construction for 128-bit Condition

```
  Input: The 64-bit and 128-bit simhash code $c_d$ and $\tilde{c}_d$ for
         each document $d$ in the Chuweb21 collection $\mathcal{D}$,
         current relation list $R$
1 RelationConstruct_128($\tau, \tilde{\tau}, R$) begin
2     Initialize the candidate relation list $R_0$ = []
3     RelationConstruct_64_great3($\tau, R_0$) % record candidate
        relations based on threshold $\tau$ with 64-bit
4     for $(d_1, d_2) \in R_0$ do
5         if HammingDistance($\tilde{c}_{d_1}, \tilde{c}_{d_2}$) $\leq \tilde{\tau}$ then
6             $R$.append($(d_1, d_2)$)
7         end
8     end
9 end
```

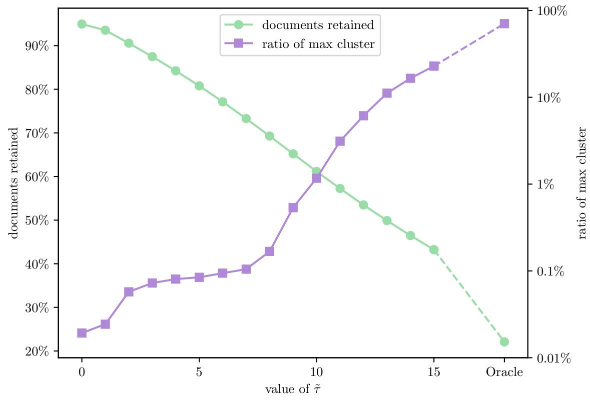


Figure 3: The proportion of remained de-duplicated documents and that of the largest cluster vary with different 128-bit thresholds $\tilde{\tau}$ based on the 64-bit τ = 6. "Oracle" denotes the result of the candidate relation set under τ = 6, i.e., set $\tilde{\tau}$ to the maximum value 128.

Table 3: The basic statistical information of Chuweb21 and Chuweb21D collections

dataset	Chuweb21	Chuweb21D-70	Chuweb21D-60
#Documents	82.5M	57.9M	49.8M
Space	1.7TB	1.2TB	964GB
Remained docs ratio	100.00%	70.18%	60.40%
#Pooling set	10,333	9,145	8,864
Remained labeling ratio	100.00%	88.50%	85.78%

more efficient use of the Chuweb21D collection afterward. In addition, Table 3 also shows the savings of annotation resources in the Chuweb21D collection compared to Chuweb21, when considering the annotation of the pooling document set under WWW-4 task. We find that the proportion of annotations saved is much smaller than the proportion of removed duplicate documents. We assume this is because the test topics of WWW-4 are relatively specific and infrequent, while duplication is more likely to occur along with popular (i.e., high clicks) document content.

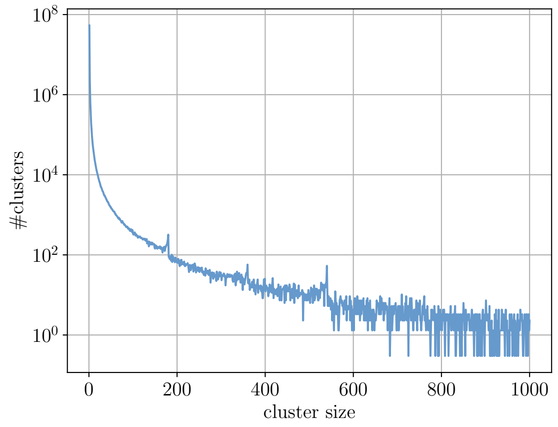
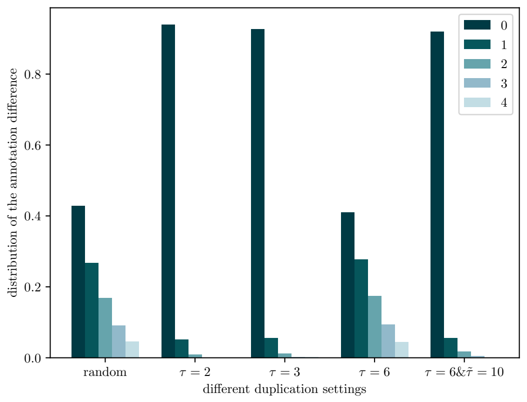
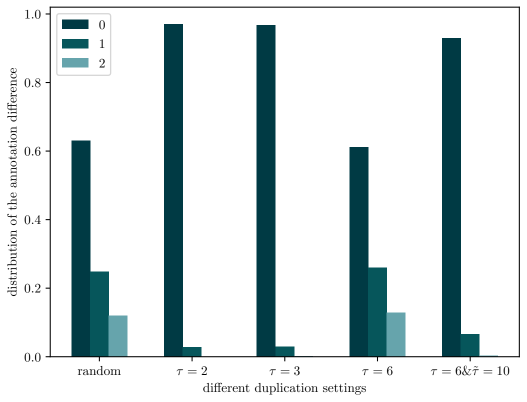


Figure 4: The population distribution of clusters with different duplication cluster scales in Chuweb21D-70 collection



(a) bronze assessment



(b) gold assessment

Figure 5: The distribution of the absolute difference in relevance annotations between the document pairs with duplicated relations in the pooling document set of the WWW-4 task. "random" denotes treating all document pairs as duplicate relations.

Figure 4 shows the population distribution of duplication clusters with different scales in Chuweb21D-70 collection. The population distribution under Chuweb21D-60 is also quite similar to Figure 4. We find that the majority of documents (92.57% of clusters, 65.03% of documents) in Chuweb21D-70 do not have duplicate documents with them. The number of clusters decays exponentially as the restricted cluster scale grows, with only 0.62% of duplicate clusters exceeding the size of 10 documents.

To further validate the reasonability of the Chuweb21D-70 ($\tau = 2$) and Chuweb21D-60 ($\tau = 3$) collections, we evaluate the annotation agreement between the documents identified as duplicates on the annotation results of WWW-4 task. Specifically, we adopt Eq. 1 to characterize the distribution of the differences in the annotation of duplicate documents, where $\mathcal{R}_{\text{dup}}$ and $I_A(x)$ denote the whole duplicated relation set and the indicator function [6].

$$P(\Delta rel = k) \propto \sum_{(d_1,d_2)\in\mathcal{R}_{\text{dup}}} I_{\{k\}}(|rel_{d_1} - rel_{d_2}|) \tag{1}$$

Figure 5 shows the difference distribution of different collections under both bronze (5-grade setting) and gold assessments (3-grade setting) released by WWW-4 task organizers. Gold assessments are those obtained from the topic creator, while bronze assessments were those obtained from those who are neither topic creators nor topic experts [1].[7]. The "random" tag denotes the condition that $\mathcal{R}_{\text{dup}}$ contains all the document pairs, perceived as the borderline distribution. The setting $\tau = 6$ delivers a poor performance as shown in Figure 5. Owing to its giant cluster, the distribution of $\tau = 6$ is not obviously distinguished from the random condition. In contrast, when we add the filtering step with 128-bit simhash code (i.e., $\tilde{\tau} = 10$) to $\tau = 6$, the constructed duplicate clusters become internally highly consistent. This phenomenon reveals that the adoption of the 128-bit code for double checking indeed assists in improving the effectiveness of de-duplication by eliminating many undesirable duplicate relations when τ takes a comparatively large value. The results in Figure 5 also reveal the validity of the final selected Chuweb21D-60 and Chuweb21D-70 collections. In these two collections, more than 92% of the document pairs being considered duplicates share the same relevance label, which serves as a signal that most of the duplicate-perceived document pairs contain similar content.

5.2 Systems Ranking Comparison

In this section, we would like to investigate the effect of de-duplication on the evaluation results of the retrieval system submitted to the WWW-4 task. Specifically, we aim to answer the following two research questions: (1) Does the performance difference among retrieval systems become more significant after de-duplication or vice versa? (2) Is the leaderboard of system performance consistent with the original version after de-duplication?

To evaluate the document ranking results for each retrieval system on the duplicated collections, we adopt the following procedures:

(1) For each duplicate cluster under the ranking list of a topic, only retain the highest-ranked document in the run and replace it with the representative document in this cluster. For example, if the original ranking result is $a_1b_2c_2c_1d_1b_1$, where a, b, c and d are four duplicate clusters, and the document with subscript 1 is the representative document of this cluster, then the ranking result after de-duplication is $a_1b_1c_1d_1$.
(2) For each duplicate cluster under the ranking list of a topic, only retain one document's relevance record in *qrels*. In this record, the document is the representative document of the cluster and the relevance score is the maximum of all the raw relevance records of the cluster. For example, if the relevance records of cluster a in the original *qrels* contain a_1(L2), a_2(L3) and a_3(L1), then the only retained record after de-duplication is a_1(L3).
(3) Using the NTCIREVAL tool [8] (the tool used for the official evaluation of the WWW-4 task), conduct system evaluation with the duplicated versions of runs and *qrels*. The evaluation metrics we apply are also identical to the official results, which include MSnDCG@10, Q-Measure@10, nERR@10 and iRBU@10.

It is worth noting that this experiment only approximates the situation where Chuweb21 is replaced by Chuweb21D for the NTCIR task, because the runs submitted to the task most probably rely on corpus statistics of Chuweb21 before deduping, such as idf and the corpus size.

To answer RQ1, we conduct the Randomised Tukey HSD test [7] for each run pair under both Chuweb21 and Chuweb21D collections. Table 4 shows the experimental results of Randomised Tukey HSD test. We can observe that the de-duplicated Chuweb21D collection achieves a competitive result with Chuweb21 in terms of the significance of system variation. Especially the Chuweb21D-70 collection significantly outperforms the Chuweb21 set on MSnDCG@10 and Q-Measure@10 metrics under bronze assessment. Under the nERR@10 and iRBU@10 metrics, the Chuweb21 collection is significantly superior to the Chuweb21D collection in terms of the system variability significance. Table 4 also shows that the metrics nERR and iRBU are considerably less discriminative (i.e., higher mean p-value) compared to MSnDCG and Q-Measure.

We use Kendall's τ to evaluate the consistency of system performance ranking across different data collections. Table 5 shows the evaluation results. Results show a high agreement in the ranking of system performance over the Chuweb21 and Chuweb21D collections: The tau values between them surpass 0.9 or even 0.95 under most of the evaluation schemes. This phenomenon indicates that we can achieve similar evaluation quality with less annotations on Chuweb21D than Chuweb21.

5.3 Case Studies

To better illustrate the effects of de-duplication, we conduct several case studies, as shown in Table 6. Table 6 presents both successful cases of de-duplication (#2 and #5) and failed situations (#1 and #3). We attribute the failure of some cases to the limited representation capacity of 64-bit hash codes. After all, terms and their

[6]https://en.wikipedia.org/wiki/Indicator_function
[7]The WWW-4 bronze assessment have been bug-fixed after NTCIR-16: we use the bug-fixed version available from http://sakailab.com/www234corrected/
[8]http://research.nii.ac.jp/ntcir/tools/ntcireval-en.html
[9]https://github.com/chuzhumin98/Chuweb21D/tree/main/materials/cases

Table 4: The Randomised Tukey HSD test results ($B = 5,000$ trials) for the WWW-4 runs evaluated on Chuweb21 and Chuweb21D collections. Each item shows two values a / b%, where a denotes the mean p-value of the whole run pair set, and b denotes the percentage of run pairs where the p-value $\leq 5\%$. † indicates that the p-value of this collection is significantly lower than the p-value of the worst dataset under this metric, and †† indicates that the p-value of this dataset is significantly lower than both of the other two datasets (p-value < 0.05).

assessment	collections	MSnDCG@10	Q-Measure@10	nERR@10	iRBU@10
gold	Chuweb21	**0.1962** / 53.59%	0.2013 / 53.59%	**0.2311**† / 35.95%	**0.2689**†† / **32.03%**
	Chuweb21D-70	0.1974 / **54.90%**	0.1968 / **56.86%**	0.2380† / **36.60%**	0.2875† / 28.10%
	Chuweb21D-60	0.1977 / 52.94%	**0.1930** / 56.21%	0.2449 / 35.29%	0.3065 / 24.84%
bronze	Chuweb21	0.1777 / 49.02%	0.1880 / 48.37%	**0.2665**†† / **31.37%**	**0.3036**†† / **30.07%**
	Chuweb21D-70	**0.1659**† / **52.94%**	**0.1737**†† / **50.33%**	0.2813† / 30.07%	0.3422† / 28.10%
	Chuweb21D-60	0.1694 / 51.63%	0.1851 / 49.02%	0.2950 / 30.07%	0.3483 / 26.80%

Table 5: The system ranking similarity between the evaluation results of Chuweb21 and Chuweb21D collection, evaluated in terms of Kendall's τ with 95% confidence intervals ($n = 18$ runs). "orig", "D60" and "D70" denote the Chuweb21, Chuweb21D-60 and Chuweb21D-70 collections respectively.

assessment	collection pair	MSnDCG@10	Q-Measure@10	nERR@10	iRBU@10
gold	orig v.s. D70	0.961 [0.923, 0.980]	0.961 [0.923, 0.980]	1.000 [1.000, 1.000]	0.856 [0.732, 0.925]
	orig v.s. D60	0.922 [0.849, 0.960]	0.961 [0.923, 0.980]	1.000 [1.000, 1.000]	0.895 [0.801, 0.946]
	D70 v.s. D60	0.961 [0.923, 0.980]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.935 [0.873, 0.967]
bronze	orig v.s. D70	0.961 [0.923, 0.980]	0.974 [0.948, 0.987]	1.000 [1.000, 1.000]	0.817 [0.665, 0.904]
	orig v.s. D60	0.948 [0.898, 0.973]	0.961 [0.923, 0.980]	0.961 [0.923, 0.980]	0.817 [0.665, 0.904]
	D70 v.s. D60	0.987 [0.974, 0.993]	0.987 [0.974, 0.993]	0.961 [0.923, 0.980]	1.000 [1.000, 1.000]

Table 6: Case studies on Chuweb21D collections. Each case contains a pair of documents that are treated as duplicates by the Chuweb21D-60 collection, for the purpose of demonstrating the success and failure of the de-duplication operation. The web pages mentioned in these cases can be viewed and downloaded on the Chuweb21D homepage [9]. "uid" denotes the unique identifier of the document in the dataset.

case id	uid of d_1	uid of d_2	duplicate in D60	duplicate in D70	remark
#1	17c1f32d-05ce-4be2-9359-a540429bf85c	d8caf0a8-32c3-4dbf-8646-4000ae1ee4ff	√	√	d_1 and d_2 present different content. d_1 reviews Yolo, an anonymous question-answering application; while d_2 comments on the failures of the game Half-Life: Alyx.
#2	7015a4d3-083d-4a82-900a-64537a48ab37	f5394d6b-6abe-4989-bfce-dc9d5fc91d09	√	√	d_1 and d_2 share the same content. Both of them introduce Czechia from various aspects, including history, geography, humanities and et al.
#3	b7f79d6a-9f43-4055-b43b-5dcce140f25d	c85f4ab2-cf55-4146-bf37-a71012c72bbc	√		d_1 and d_2 show completely different information. d_1 illustrates Brian Lewis' statements about catalytic effects of CYGTT 2021; while d_2 introduced some interesting communities on the dark web.
#4	99f3b6c5-4c2c-4cb3-b62c-73e5ee1715e8	32467779-4141-49dc-b61a-72554327be9d	√		Both d_1 and d_2 are affiliated with the same anti-nuclear website. They share the same right-column content, which contains the latest anti-nuclear statements and news of current month; while they present different anti-nuclear news on the left side.
#5	6b0826a0-7d8d-4f29-a93a-f103f42d73a0	9284845a-44f9-43de-8105-fcf515a5ac2a	√		Both d_1 and d_2 comprise multiple Comic Con-related blogs. Although the arrangement of these blogs seems slightly different, most of the materials are the same between the two documents.

frequencies create high-dimensional representation vectors, making it inevitable to have hash conflicts in some cases. Moreover, #4 in Table 6, as a more interesting case, presents a semi-duplicate situation: the right columns of documents share the same content, whereas the left parts differ. Under case #4, collections with different tightness of deduplication yield various outcomes.

6 CONCLUSION

In this paper, we release a novel de-duplicated English document collection for web search tasks (Chuweb21D). The Chuweb21D data collection is constructed based on the Chuweb21 collection, which was adopted as the target corpus for the NTCIR-16 WWW-4 task. We de-duplicate the Chuweb21 collection, and publicly release two versions of the Chuweb21D collections with different deduping thresholds, named Chuweb21D-60 and Chuweb21D-70; the former is used as the target corpus for the ongoing NTCIR-17 FairWeb-1 (Group-Fair Web Search) task. In this paper, we present the construction procedure of Chuweb21 and Chuweb21D data collections in detail, especially the analysis and selection under different deduplication thresholds. We also explore the impact of deduping on the results of the NTCIR-16 WWW-4 task. The experimental results show that the de-duplicated Chuweb21D collection possesses a competitive system variance discrimination ability compared to the Chuweb21 collection, and achieves a high agreement with the Chuweb21 collection in terms of system performance ranking. We hope that the presented details and findings could enlighten follow-up studies on the construction and usage of web search collections.

ACKNOWLEDGMENTS

We are grateful to the other task organizers of the WWW-4 and FairWeb-1 tasks, especially Ian Soboroff, for the discussion on the construction of Chuweb21D. We also thank Ruizhe Zhang for his suggestions on the design of the de-duplication strategy in case of $\tau > 3$. Moreover, this work is supported by Quan Cheng Laboratory (Grant No. QCLZD202301)

REFERENCES

[1] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P de Vries, and Emine Yilmaz. 2008. Relevance assessment: are judges exchangeable and does it matter. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 667–674.

[2] Yaniv Bernstein and Justin Zobel. 2005. Redundant documents and search effectiveness. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. 736–743.

[3] Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE, 21–29.

[4] James H Burrows. 1995. *Secure hash standard*. Technical Report. Department of Commerce Washington DC.

[5] Jamie Callan. 2012. The lemur project and its clueweb12 dataset. In *Invited talk at the SIGIR 2012 Workshop on Open-Source Information Retrieval*.

[6] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. The clueweb09 dataset, 2009. *URL http://boston. lti. cs. cmu. edu/Data/clueweb09* (2009).

[7] Benjamin A Carterette. 2012. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Transactions on Information Systems (TOIS)* 30, 1 (2012), 1–34.

[8] Moses S Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. 380–388.

[9] Charles LA Clarke, Nick Craswell, and Ian Soboroff. 2004. Overview of the TREC 2004 Terabyte Track.. In *TREC*, Vol. 4. 74.

[10] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms*. MIT press.

[11] Nicola Ferro and Carol Peters. 2019. *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*. Vol. 41. Springer.

[12] Maik Fröbe, Janek Bevendorff, Lukas Gienapp, Michael Völske, Benno Stein, Martin Potthast, and Matthias Hagen. 2021. CopyCat: Near-Duplicates within and between the ClueWeb and the Common Crawl. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2398–2404.

[13] Maik Fröbe, Jan Philipp Bittner, Martin Potthast, and Matthias Hagen. 2020. The effect of content-equivalent near-duplicates on the evaluation of search engines. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*. Springer, 12–19.

[14] Zvi Galil and Giuseppe F Italiano. 1991. Data structures and algorithms for disjoint set union problems. *ACM Computing Surveys (CSUR)* 23, 3 (1991), 319–344.

[15] Omid Jafari, Preeti Maurya, Parth Nagarkar, Khandker Mushfiqul Islam, and Chidambaram Crushev. 2021. A survey on locality sensitive hashing algorithms and their applications. *arXiv preprint arXiv:2102.08942* (2021).

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

[17] Joel Mackenzie, Rodger Benham, Matthias Petri, Johanne R Trippas, J Shane Culpepper, and Alistair Moffat. 2020. CC-News-En: A large English news corpus. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3077–3084.

[18] Mehdi Ebady Manaa and Ghufran Abdulameer. 2018. Web documents similarity using k-shingle tokens and minhash technique. *Journal of Engineering and Applied Sciences* 13, 6 (2018), 1499–1505.

[19] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. 2007. Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web*. 141–150.

[20] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[21] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[22] Arnold Overwijk, Chenyan Xiong, and Jamie Callan. 2022. ClueWeb22: 10 Billion Web Documents with Rich Information. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3360–3362.

[23] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* (2020).

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints* (2019). arXiv:1910.10683

[25] Ronald Rivest. 1992. *The MD5 message-digest algorithm*. Technical Report.

[26] Tetsuya Sakai, Douglas W Oard, and Noriko Kando. [n. d.]. *Evaluating Information Retrieval and Access Tasks: NTCIR's Legacy of Research Impact*. Springer Nature.

[27] Tetsuya Sakai, Sijie Tao, Zhumin Chu, Maria Maistro, Yujing Li, Nuo Chen, Nicola Ferro, Junjie Wang, Ian Soborof, and Yiqun Liu. 2022. Overview of the NTCIR-16 We Want Web with CENTRE (WWW-4) Task. *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies* (2022).

[28] Tetsuya Sakai, Sijie Tao, Maria Maistro, Zhumin Chu, Yujing Li, Nuo Chen, Nicola Ferro, Junjie Wang, Ian Soboroff, and Yiqun Liu. 2022. Corrected Evaluation Results of the NTCIR WWW-2, WWW-3, and WWW-4 English Subtasks. *arXiv preprint arXiv:2210.10266* (2022).

[29] Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*. Vol. 63. MIT press Cambridge.