To appear in "Datenbank-Spektrum: Special Issue on Data and Repeatability"

Measuring and Facilitating Data Repeatability in Web Science

Julian Risch · Ralf Krestel

Received: 6th February 2019 / Accepted: 9th May 2019

Abstract Accessible and reusable datasets are a necessity to accomplish repeatable research. This requirement poses a problem particularly for web science, since scraped data comes in various formats and can change due to the dynamic character of the web. Further, usage of web data is typically restricted by copyright-protection or privacy regulations, which hinder publication of datasets.

To alleviate these problems and reach what we define as "partial data repeatability", we present a process that consists of multiple components. Researchers need to distribute only a scraper and not the data itself to comply with legal limitations. If a dataset is re-scraped for repeatability after some time, the integrity of different versions can be checked based on fingerprints. Moreover, fingerprints are sufficient to identify what parts of the data have changed and how much.

We evaluate an implementation of this process with a dataset of 250 million online comments collected from five different news discussion platforms. We re-scraped the dataset after pausing for one year and show that less than ten percent of the data has actually changed. These experiments demonstrate that providing a scraper and fingerprints enables recreating a dataset and supports the repeatability of web science experiments.

Keywords Repeatability · Web Science · Web Scraping · Fingerprinting

Julian Risch and Ralf Krestel
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam
E-mail: firstname.lastname@hpi.de

1 Repeatability in the Context of Large Datasets in the Web

Repeatability plays an important role in modern sciences and is an essential criterion for good research. If an experiment is repeatable, other researchers can double-check and compare its results. Successful repetition provides further evidence and thereby builds trust and credibility. If an experiment is not repeatable that does not necessarily mean falsification. For example, prior work might have failed to describe an experiment setup in enough detail and still its findings could be correct. However, other researchers can hardly rely on or build on these findings. To this end, repeatability is the key to scientific progress as it allows researchers to rely on prior work and thus move on to novel tasks.

SIGMOD[1] and pVLDB[2] recently intensified their efforts to encourage repeatability and reproducibility in the database community. To accomplish this ambitious goal, a first step is to ensure availability. All information necessary to re-create an experiment, which comprises software, datasets, experiment setups, and steps to render result graphs[3], need to be published. We define *data repeatability* as the availability of data in a way that enables to re-run an experiment. Currently there are two ways to achieve this goal:

1. The dataset itself can be provided.
2. A way to generate the dataset can be described.

Regarding the first, there are online data repositories specialized on storing research datasets, such as Mendeley[4]. However, a quick survey of such repositories shows

[1] `http://db-reproducibility.seas.harvard.edu/`

[2] `https://vldb-repro.com/`

[3] `https://vldb-repro.com/\#process`

[4] `https://data.mendeley.com/`

that only a small minority of researchers uses them. And again, legal restrictions might exclude this option completely. Regarding the second possibility, e.g. benchmarks of database algorithms typically fall in this class, datasets are generated according to pre-defined probability distributions. Popular benchmark data generators are dbtesma[5] and dbgen.[6] We propose to steer a middle course and provide a tool that generates the dataset by scraping web data. The difference is that the dataset itself is not distributed and the process to generate the data involves neither randomness nor synthetic data.

In this paper, we focus on the data aspect of repeatability in the field of web science. Experiments that deal with web data are especially hard to repeat for several reasons:

1. Web scrapers collect data from different sources in various formats, which need to be integrated and pre-processed in the same way;
2. Web data is not static and thus scraping at different points in time can lead to different results;
3. Web data is usually copyright-protected or its use is restricted by privacy regulations, which hinders publication of datasets even if it is only for research and not for commercial purposes.

We deal with these issues by introducing a process to measure and facilitate repeatability. This process consists of two components: a scraping component and a fingerprinting component. If researchers are unable to publish their dataset, we suggest that they instead publish implementations of these two components: a process to re-create the data in the form of a scraping tool and a process to create and compare fingerprints of the data.

The first component is to comply with legal restrictions or ethical concerns that hinder the publication of the dataset. For example, imagine that a dataset scraped from a social network contains personal data. If researchers published such data, affected persons could not remove their records from the dataset and neither could the original platform provider. Even if the user removed his or her data from the platform, it would remain in the published dataset. In contrast to that, our proposed approach ensures that the data can be edited by the user or the platform provider. At the extreme, the provider could prevent the usage of scrapers on the platform or take the data offline.

At first glance this might seem as a major disadvantage for research. But this cost is necessary to allow users and platform providers to retain control of their data. In fact, for researchers it comes with the advantage that datasets can be deleted locally after an experiment. There is no need to keep the data stored in a de-centralized way. The only place to store the data is the original provider. We assume that typically only slight changes are made and thus ensure what we call *partial data repeatability.*

The second component is to confirm this assumption: the component checks whether parts of the web data have changed. To this end, fingerprints are taken after initially collecting the data and after each rescraping. A comparison of the fingerprints serves as an integrity check without having to compare the actual data. Moreover, a similarity function defined on the fingerprints allows to measure the extent of changes. For example, this similarity can estimate the number of changed words in a text document. The fingerprints further allow to identify which subset of the data has changed and which remains unchanged. Thereby, an experiment can be repeated on an unchanged subset for better comparability.

We implement the proposed process and apply it to the field of online comment analysis to show its practical feasibility. The analysis of how people discuss only relies on the manifestation of such discussions at the web pages of online platforms, such as Twitter, Facebook, Reddit, or discussion sections of news platforms. Web scientists depend on web pages and their content, which they crawl, analyze, and use in experiments to evaluate their approaches. If the data can be freely distributed, there is no need for our approach. However, the terms of use of these platforms often prohibit to distribute the scraped dataset. As a consequence, datasets used in the field of online comment analysis are rarely published and experiments difficult to repeat. Our experiments demonstrate that a web dataset can be rescraped to repeat an experiment even after a year. Thus we conclude that the users' option to delete data and the researchers' desire for data repeatability are not necessarily mutually exclusive, but can exist in parallel.

2 Related Work

We begin with a definition of repeatability and reproducibility and summarize how they are currently handled in computer science. Strengths and weaknesses of different attempts to improve the current state are then compared. Further, we focus on repeatability in our exemplary research field of user comment analysis. A brief overview of fingerprinting techniques concludes this section.

[5] https://sourceforge.net/projects/dbtesma/

[6] https://github.com/electrum/tpch-dbgen

2.1 Repeatability and Reproducibility

According to Cohen et al. replicability and repeatability interchangeably describe the ability to recreate an experiment exactly as reported and come to the same results [6]. In contrast to that, reproducibility describes the ability to come to the same conclusions, findings or values as reported even if a different method is used. Reproducibility considers not only algorithms and their implementation in code, but also theorems and their proofs, and datasets [20].

While repeatability and reproducibility seem to be a foundation of science, it is by far not the standard in todays computer science research. Out of 601 papers from ACM conferences and journals, only one third provides source code [7]. Similarly, two independent studies on IEEE Transactions on Image Processing find that only one third of the papers make datasets available online [14,23]. Several tools attempt to improve repeatability by creating self-contained packages for experiments [12,13,19]. As an extension to these approaches, "Reprozip" adds support for VMs and Docker containers [5]. Thereby, created packages can be distributed to other operating systems.

Pedersen suggests to plan for releasing software from the start of a research project [18]. More open source software in machine learning would allow researchers to build on each others tools [22]. We transfer this idea to data and suggest to plan from the start for releasing datasets or a way to obtain them. Only if datasets can be accessed and modified, the research community can enrich existing datasets, connect them, and build them together. Vitek and Kalibera go one step further and question any experiments on unpublished (proprietary) datasets [24]. According to them, researchers can learn something from others' experiment results only if they can inspect and understand the dataset. And even if the data is available, Drummond emphasizes that it is important to document also the way it has been collected [9]. For example, this documentation helps to reveal sampling bias or worse "purposive sampling". The latter refers to samples that are not chosen randomly but by judgment of researchers.

Blockeel and Vanschoren present "experiment databases" and how to construct them [3]. The idea is to store all information about experimental setups in one online repository, which can be queried by other researchers. The vision of a Linked Open Data graph of related experiments goes into a similar direction [17]. As a first step towards this vision for the field of web science, we monitor to what extent datasets in the web remain unchanged. Blanco et al. propose a standardized evaluation framework for the semantic search domain [2]. This framework comprises standard datasets, queries, and metrics. The authors find that even crowdsourced relevance judgments are repeatable in their experiment. Godbole et al. study re-usability for research on text mining, for example on entity extraction [10]. They focus on dictionary-based approaches and bring forward best practices to make dictionaries re-usable across datasets, such as a service-oriented modular approach. To the best of our knowledge, no previous work studies repeatability of experiments on web data and its inherent dynamics.

2.2 Comment Analysis as an Exemplary Task in Web Science

Enormous datasets of online discussions are essential to comment analysis research. But besides a few positive exceptions, repeatability is unsupported. Some comment datasets are published and enable repeatability, such as the "Yahoo News Annotated Comments Corpus" (522k unlabeled and 10k labeled comments) [16], the "One Million Posts Corpus" (1M unlabeled and 12k labeled comments) [21], and a collection of Wikipedia discussion pages (100k human-labeled and 63M machine-labeled comments) [25]. For the latter, published annotation instructions even make the data collection process repeatable.

This process is prone to a sampling bias, especially for datasets of offensive comments. Data is not collected as representative samples of the platform, because of the strong class imbalance. Typically, less than ten percent of online comments are offensive. However, the sampling aims for a more balanced class distribution to improve the training of machine learning classifiers. An example for sampling bias are 25k tweets collected by Davidson et al. [8]. They sampled tweets based on a hate speech lexicon to collect offensive comments. Comments that do not match with the lexicon are less likely to be included than others, so this dataset is not representative. Unfortunately, most comment datasets are not published at all, for example because of copyright and privacy concerns. The non-repeatable fallback option is to re-implement related work approaches and compare with them on private datasets.

2.3 Scraping and Fingerprinting

In this paper, we distinguish web crawling from web scraping. We use crawling to describe progressively following links, while we use scraping for extracting information from (a pre-defined set of) web pages. For both, crawlers and scrapers, it is essential to know whether

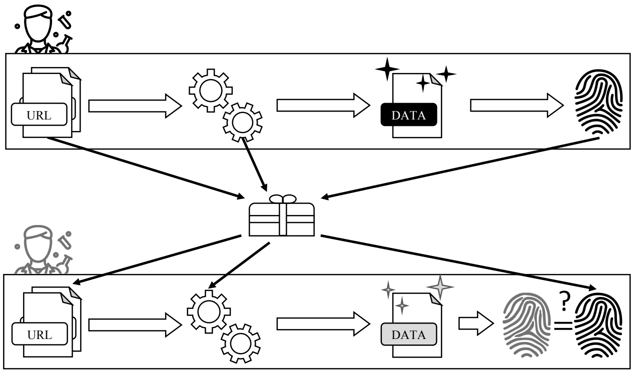


Fig. 1 A researcher (top) defines a set of URLs to collect a dataset and calculates its fingerprints. URLs, scraper, and fingerprints are provided to a second researcher (bottom), who re-scrapes the URLs and obtains a different version of the dataset. The fingerprints of these versions are then compared.

the web content of interest has changed since the last visit. If it is unchanged, there is need to update, for example, a search engine's index. In general, hash functions are used to detect content changes and they can also be applied to unstructured text data. A popular function is Charikar's locality-sensitive simhash function [4], which has been applied for web crawling [15]. In contrast to cryptographic hash functions, similar input texts are hashed to similar hash values. We use this property to estimate the number of changed words based on the difference of two hashes. If hashes are used to compare larger amounts of data, they are also called fingerprints and the underlying technique is fingerprinting. While we focus on unstructured data in this paper, fingerprinting is also used for structured data in relational databases. An overview of fingerprinting techniques for relational databases can be found in a paper by Halder et al. [11].

3 Approach

Figure 1 visualizes the proposed process, which consists of a scraping component and a fingerprinting component. Their combination allows to re-scrape a dataset, estimate the extent of changes compared to the original version, and identify a data subset that remained unchanged. Thereby, it can be estimated whether the re-scraped data is sufficient to re-run experiments in a comparable way. If so, the experiment can claim *data repeatability*.

3.1 Scraping Component

This component implements a way to extract a dataset from the web. To accomplish repeatability, web content that needs to be obtained requires an identifier. In the world wide web, the most typical identifier are Uniform Resource Identifiers (URIs) and if the location is specified Uniform Resource Locators (URLs). Therefore, the scraper needs to be accompanied by a list of URLs to collect data from. The scraping process can be run in parallel. For example, the list of URLs can be separated into smaller lists for multiple scraper instances.

The implementation of this component can be accomplished in different ways. The most naive way is to scrape every web page in the specified list and extract the desired content. Some websites provide an application programming interface (API), which can be used instead of actual web pages. APIs reduce necessary data transfer but often times also limit the number of API calls per day. Collecting 250 million items with a rate limit of 1000 calls per day would take more than 685 years. In rare cases, access to web content might be limited based on geolocation. Thus, the scraper is required to be used through this location, for example with a proxy server.

Once data has been fetched from different sources, it needs to be integrated into a common data structure. Unifying various data formats and boilerplate removal are the challenging tasks for this step. It ensures that further processing of the data, for example in experiments, works on a well defined basis. Further, it can normalize different data formats on the same platform if they change over time. To connect this component with the second one, every unit that can be scraped independently should be accompanied not only with a unique identifier but also with a fingerprint. The identifier is necessary so that a re-scraped version's fingerprint can be matched and compared to the initial version's fingerprint.

3.2 Fingerprinting Component

Fingerprinting methods and especially locality sensitive hash functions have properties that come in handy for detecting content changes, for example in web crawling [15]. One of these properties is desirable also in our scenario: small content changes result in small fingerprint changes. This property is not guaranteed vice versa because of potential hash collisions. There is a small chance that two records with the same fingerprint are very different content-wise. However, a 64-bit fingerprint has a range of 2^{64} values and thus the probability of collisions when hashing 250 million records ($\approx 2^{28}$) is rather small. Therefore, in practice, similar fingerprints are assumed to mirror similar content.

The fingerprinting component checks whether a re-scraped dataset differs from the original version. If so, this component measures the difference and identifies the largest unchanged subset of records. An implementation needs to define a fingerprinting function and a distance function. The challenge is to find functions that allow to distinguish slight changes that do not hinder repeatability from drastic changes that prevent repeatability.

There are several reasons why a re-scraped dataset might differ from the original one.

1. Parts of the web page have changed, for example, content has been added;
2. The full web page has changed, for example, it has been deleted or moved to a different URL;
3. The API or source code of the website has changed and thus the scraping tool does not work anymore.

The fingerprinting function ϕ maps arbitrary web content $x \in W$ to a fingerprint $y \in 0,1^n$, where W is the domain of web content, and n is the number of bits used for the fingerprint. The distance of two fingerprints y_1 and y_2 is defined as their Hamming distance (number of differing bits) and thus is a natural number in the interval $[0, n]$. The similarity function SIM maps pairs of web content $x_1, x_2 \in W$ to real numbers between 0 and 1:

$$\phi : W \mapsto \{0,1\}^n$$
$$\text{SIM} : W \times W \mapsto [0,1]$$
$$\text{Hamming distance} : \{0,1\}^n \times \{0,1\}^n \mapsto [0,n]$$

An example for SIM is the edit-distance for texts (mapped to real numbers between 0 and 1). ϕ is called a locality-sensitive hash function corresponding to the similarity function SIM. Similar input (according to SIM) is mapped to similar fingerprints (according to Hamming distance). There is a trade-off between granularity of discoverable data changes and memory consumption. On the one hand, if one fingerprint represents multiple units, the granularity of discoverable data changes gets worse. On the other hand, memory consumption to store the fingerprints decreases.

For text data, the fingerprinting component takes a text as input and tokenizes it. The tokenized text is then converted to k-shingles, where k denotes the number of words of each shingle. Shingles are all possible consecutive subsequences of k tokens. A different name for the same concept is word n-grams. A hash function (which does not need to be locality-sensitive) is applied to each shingle. Typically, the $md5$ hash function is used. From the sequence of hashes a fingerprint is taken. We use fingerprints of 64-bit length. This fingerprint can be compared to others based on their Hamming distance. Figure 2 exemplifies this procedure.

Table 1 Statistics for the one-year-old datasets

News Platform	Comments	Articles	Users
Daily Mail	129,732,977	1,414,258	1,764,557
The Guardian	61,491,774	625,690	1,213,555
Fox News	52,224,398	49,266	465,954
The Independent	5,598,425	171,052	211,114
Russia Today	687,436	65,384	49,333

4 Experiments

We implement the process to show its practical feasibility for two different kinds of data: user comments and news articles. To this end, we crawled a large dataset of user comments one year ago and now re-scrape it based on the same URLs. For the two versions of the dataset we measure the exact number of changed comments (ground truth). The difference of fingerprints serves as an estimation of this number. Our implementation is published online[7].

4.1 Dataset

The collection of user comments from discussion pages of five English-language online news platforms contains 250 million comments in total (Table 1).[8] We integrate all data according to a unified model into one large dataset. To this end, we propose a unified data model for online news discussions as visualized in Figure 3. A comment is represented with a comment id (primary key), the user id of its author (foreign key), the id of the referenced news article (foreign key), the comment text itself, its timestamp, and if existent, the number of upvotes and downvotes. If the comment is a reply to another comment, the parent's comment id is referenced (foreign key). An article is composed of its id (primary key) and its article URL, which typically contains the article category, such as politics, sports, etc., and the article title. Users are represented with their id (primary key) and their user name. No other user-specific information, in particular no demographic information, is stored.

4.2 User Comments

First, we analyze what kind of changes can occur at online discussions. Section 3.2 lists abstract reasons why

[7] `https://hpi.de/naumann/projects/repeatability/text-mining.html`

[8] Links to these platforms are `https://www.dailymail.co.uk`, `https://www.theguardian.com`, `https://www.foxnews.com`, `https://www.independent.co.uk`, `https://www.rt.com`.

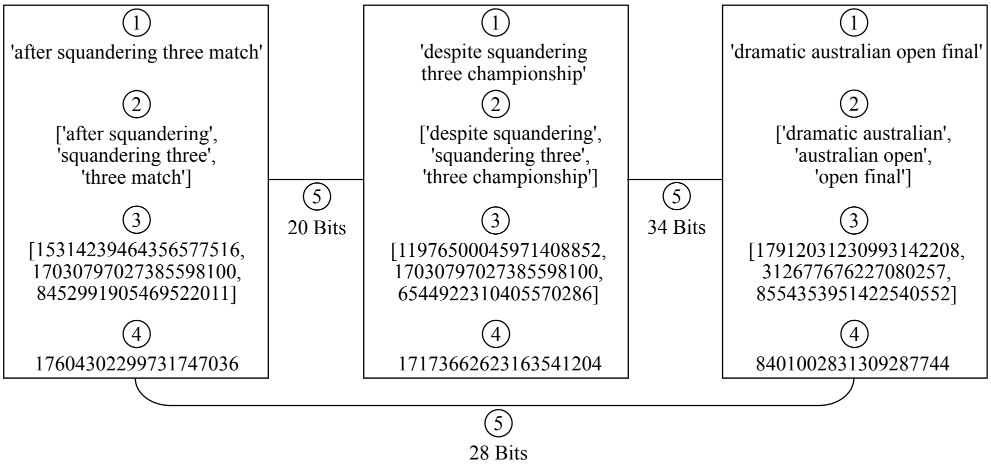


Fig. 2 A text (1) is transformed to word bi-grams (2) and an md5 hash is calculated for each of those (3). A locality sensitive hash of this bi-gram sequence serves as a fingerprint (4). Fingerprints can be compared based on their Hamming distance (5).

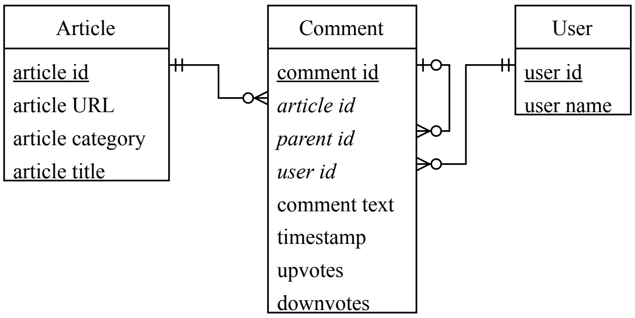


Fig. 3 The unified data model for online news discussions considers comments and their metadata, including references to users and news articles.

a re-scraped dataset might differ from the original one. In this particular scenario of user comments at online news platforms, the concrete reasons are:

1. a comment has been added;
2. a comment has been deleted;
3. the full article has been deleted or moved;
4. the way to access comments has been changed for all articles.

We assume that the discussion sections of most articles remain unchanged within one year. One reason for this assumption is the relatively short attention span in online news. An article is rarely commented on a few days after its publication. Research on comment volume prediction found that 90 percent of an article's comments are posted within two to three days, which supports our assumption [1]. A second reason is that news platforms typically close an article's comment section after some time. No more comments can be added and thus, discussion moderators can focus on a smaller set of articles.

Except for The Guardian, news platforms in our study do not provide identifiers for comments. Only news articles and their full discussions are identified by URLs. Therefore, we calculate one fingerprint per full discussion. More specifically, a first step calculates one fingerprint per comment based on shingles of length eight and the simhash function [4]. The comparison of fingerprints uses the Hamming distance. Thus, if a comment is slightly changed, its fingerprint remains similar. A second step calculates a fingerprint for the full sequence of an article's comments' fingerprints. Again the fingerprint is based on shingles of length eight and the simhash function. As a result that can be published online, we store an article URL and a fingerprint per discussion.

After one year, we use the URLs to re-scrape the comments. In the following, we first compare the actual records of the two datasets and then their fingerprints. Two comments are assumed identical if their timestamps and texts are exact matches. The question is: How many comments of the original dataset have been re-scraped successfully and did not change within one year?

Table 2 lists the relative number of re-scraped comments and articles per news platform. About 90 percent of the original number of comments and articles have been retrieved. For The Guardian, 61,469,631 out of 61,491,776 comments are retrieved (more than 99.9 percent). In contrast, Fox News switched its third-party commenting system from Disqus to Spot.IM within the considered period, which renders earlier comments inaccessible. Thus, this platform is excluded from the rest of our study.

For the Independent, the original dataset contains about 5.6 million comments. When we re-scrape the same article URLs, 1.6 million comments are missing.

Table 2 Fraction of unchanged comments and articles

News Platform	Re-scraped Comments	Re-scraped Articles
Daily Mail	0.89	0.94
The Guardian	1.00	0.93
Fox News	-	-
The Independent	0.73	0.83
Russia Today	0.88	0.99

For the large majority (1.4 million) of the missing comments, the article URL does not correspond to an article anymore. The remaining 0.2 million missing comments have been deleted from the platform since the first crawling. The median percentage of unchanged comments is 0.97, while the mean percentage of unchanged comments is 0.76. That the median is much higher than the mean is because there is only a small number of articles with a large number of missing comments.

However, a similar number of retrieved comments does not necessarily mean that their content did not change. For example, on the English-language platform Russia Today, moderated comments are replaced with the text “DELETED”. While 99 percent of the articles are re-scraped successfully, only 88 percent of the comments are retrieved. The 12 percent missing comments are distributed across 55 percent of the article discussions. Thus, only 45 percent of all discussions remain unchanged. We come to a similar conclusion, if we compare 32-bit fingerprints of the discussions based on shingles of length four. They suggest that a relatively large set of article discussions changed (34 percent). We assume that this underestimation is due to rather small shingles and a too small number of bits per fingerprint. A study on the influence of shingle length and fingerprint length on the estimation quality remains future work.

Moderated comments on The Guardian’s platform are typically replaced with the text “Deleted by Moderator.”. Replacements like this make up about five percent of the comments in our dataset. Thus, the scraping can recreate at most 95 percent of the original data. This is the reason why only *partial* data repeatability can be achieved.

The second experiment is a simulation. We artificially alter the initially crawled dataset stepwise by deleting comments and adding others. Each step randomly selects an article and replaces a random comment of this article with a random comment from a different article. Roulette wheel selection favors articles with more comments. We assume that longer discussions are more likely to change, not only because they contain more comments but also because they involve more users. Each simulation step updates also the discussion’s fingerprint and the distance to the fingerprints of the original dataset. Figure 4 visualizes the linear relationship of comments changed and fingerprint bits changed. For example, if less than five fingerprint bits change, we assume that less than 20 percent of the comments have changed.

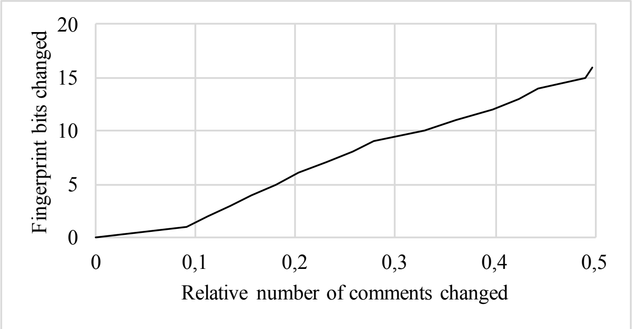


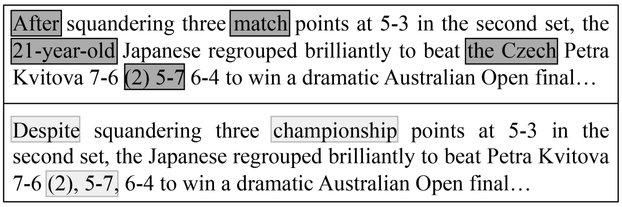
Fig. 4 We simulate deletion and addition of comments in an article’s discussion section and measure how the difference of fingerprints increases. Due to a linear correlation the relative number of comment changes can be estimated based on the number of fingerprint bits changed.

4.3 News Articles

We study a second use case besides user comments, which are news articles. We consider experiments on a dataset of news articles (partially) repeatable if the articles’ texts change only slightly or not at all. Table 2 shows that the large majority of articles can be re-scraped from the same URL even after one year. Typical changes of an article are not complete deletions but text corrections and event updates, which happen within a short time after publication. To analyze changes of article text within a shorter time period, we crawled news articles published on a particular day and re-scrape the same articles two days later. Out of 147 articles from The Guardian, the texts of 132 articles (90 percent) remained unchanged within this time period. Based on fingerprint comparison, we correctly identify all of these articles, but also mis-classify two additional articles as unchanged. However, only minor stylistic changes are made, which do not significantly alter its meaning. Figure 5 exemplifies how an article text changed over time.

The second experiment simulates changes of an article text iteratively. We hypothesize that the fingerprint distance of the original text and the altered text is a

[10] http://gu.com/sport/2019/jan/26/naomi-osaka-wins-australian-open-final-petra-kvitova-grand-slams-tennis



After squandering three match points at 5-3 in the second set, the 21-year-old Japanese regrouped brilliantly to beat the Czech Petra Kvitova 7-6 (2) 5-7 6-4 to win a dramatic Australian Open final…

Despite squandering three championship points at 5-3 in the second set, the Japanese regrouped brilliantly to beat Petra Kvitova 7-6 (2), 5-7, 6-4 to win a dramatic Australian Open final…

Fig. 5 An excerpt of an article on sports scraped right after publication (top) and a few hours later (bottom) exemplifies that only minor stylistic, changes are made.[10]

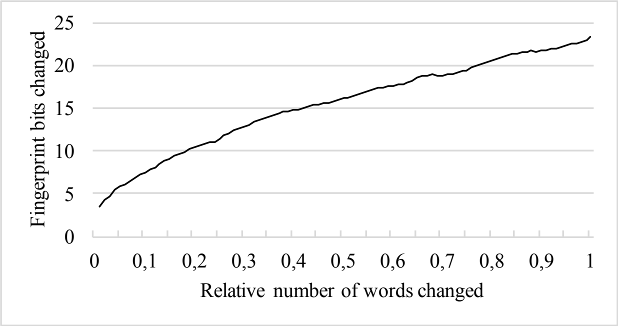


Fig. 6 We simulate deletion and addition of words in an article's text and measure how the difference of fingerprints increases. Due to a linear correlation the relative number of changed words can be estimated based on the number of fingerprint bits changed.

good estimation of how similar the texts are. We further hypothesize that measured distance can be used to estimate the number of changed words. We test these hypotheses with this experiment. To alter an article's text in a way that keeps a realistic language use and sentence structure, we blend it with the text from another article. With increasing probability, we replace a word of article A with the word of article B at the same position. For example, the first word of A is replaced with the first word of B with a low probability. Each iteration increases this probability until, at the end of the process, all words are replaced with a probability of 1.

Figure 6 visualizes the relation between the relative number of changed words and the number of fingerprint bits changed. The plot corresponds to the average of ten randomly selected pairs of articles with ten random simulations each. The linear relationship allows to estimate the number of changed words based on fingerprint distance. For example, if the fingerprints of two scraped versions of an article differ in five bits, we assume that about five percent of the words have changed. The experiment hence confirms that fingerprint distance is a good estimation for how similar two article texts are.

5 Discussion

Our process enables scientists to re-use datasets even if distribution is restricted. However, repeatability by definition relies on the exact same conditions for re-running an experiment. Our experiments indicate that most web comments and web articles remain unchanged but not all of them. In accordance with the definition of repeatability, the identification of unchanged subsets is strictly speaking not enough. However, *partial repeatability* can be ensured by our process. At the borderline of repeatability and reproducibility, our process reproduces the data as good as possible so it tries to reproduce the conditions with regards to the dataset as precise as possible. Thus, we aim for repeatability but if the exact conditions cannot be reproduced, we make it as similar as possible. Assuming that the reproduced data subset is a representative sample of the full set, re-running an experiment only on the former is sufficient and justifiable. A similar assumption underlies training, validation, and test data splits in machine learning in general. In the context of dynamic web data, the subset is presumably not a perfectly random sample of the full set. However, there is no reason to assume a systematic bias either.

A limitation of our study is that we looked at the data aspect of repeatability and neglected software, algorithms, theorems, and proofs. All aspects need to be taken into account to make a complete experiment repeatable. In addition to publishing scrapers and fingerprints, a thorough description of how the data was selected is required. This information is needed to rule out any sampling bias and to understand whether the data is useful for other experiments. A list of URLs to scrape defines the data selection within our proposed process. The question is how is this list compiled? Is there some potential bias in this compilation? In our example with online news comments, we crawled all available news articles and their comments from selected platforms at a fixed point in time. While the point in time was chosen arbitrarily, the selection of platforms might introduce some bias, which we are, however, unaware of.

The impact on web content providers must not be neglected. On the one hand, scraping the web supplies researchers with precious datasets. On the other hand, it increases load for platform providers. The need of users and platform providers for an option to delete data and the need of researchers to repeat an experiment on the same data are a trade-off. For example, hate speech detection focuses on comments that are typically deleted by discussion moderators. Once deleted, there is no way to re-scrape them. Our approach supports this option for platform providers and therefore experiments on deleted (hate speech) comments cannot be repeated.

Sometimes datasets are manually labeled after scraping them. These labels need to be distributed together with the scraping component. To match these labels to the re-scraped data records, the mapping of labels to comments needs to be documented. This documentation could consist of pairs of identifiers and labels. In the example use case, comments might not have unique identifiers but only the full news article. In this case a fingerprint of a comment itself might be used as an identifier, although there is no guarantee that there is a unique match.

6 Conclusions and Future Work

We studied the data aspect of repeatability in the field of web science. Repeatability is especially difficult to establish in this field because of the dynamics of the web data it deals with. To comply with legal and ethical restrictions on distributing such datasets, we presented a process that consists of scraping and fingerprinting. Researchers do not need to share the actual dataset but only the implementation of a process to gather the data and calculate its fingerprints. When the dataset is rescraped to repeat an experiment, fingerprints measure integrity and identify unchanged subsets of the data. As a consequence, experiments can be re-run on the exactly or almost same data without researchers publishing the dataset itself. The option to edit or delete records remains with the content provider while researchers can check to what extent changes occurred. Thereby not only copyrights of the provider but also privacy concerns of users are taken into account.

We showed the practical feasibility of the proposed process with implementations for two different kinds of data: user comments and news articles. To this end, we collected a dataset of 250 million user comments from five different online news platforms and checked its similarity to a re-scraped dataset after one year. The results show that only a small subset of comments is changed or deleted within one year and that news articles are also only slightly changed once published. A promising direction for future research is to constantly monitor data repeatability for a particular experiment over time. Thereby, authors of a paper could be notified, when an API changes and as a consequence the provided scraper needs to be adapted. Last but not least, we did not show that the proposed process is generally applicable to different kinds of web data. Thus, an open task for future work is to test our process in the context of various web science scenarios.

References

1. Ambroselli, C., Risch, J., Krestel, R., Loos, A.: Prediction for the newsroom: Which articles will get the most comments? In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 193–199. ACL (2018)
2. Blanco, R., Halpin, H., Herzig, D.M., Mika, P., Pound, J., Thompson, H.S., Tran, T.: Repeatable and reliable semantic search evaluation. Web Semantics: Science, Services and Agents on the World Wide Web **21**, 14–29 (2013)
3. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: European Conference on Principles of Data Mining and Knowledge Discovery (ECML PKDD), pp. 6–17. Springer (2007)
4. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the ACM Symposium on Theory of Computing, pp. 380–388. ACM (2002)
5. Chirigati, F., Rampin, R., Shasha, D., Freire, J.: Reprozip: Computational reproducibility with ease. In: Proceedings of the International Conference on Management of Data (SIGMOD), pp. 2085–2088. ACM (2016)
6. Cohen, K.B., Xia, J., Zweigenbaum, P., Callahan, T.J., Hargraves, O., Goss, F., Ide, N., Névéol, A., Grouin, C., Hunter, L.E.: Three dimensions of reproducibility in natural language processing. In: International Conference on Language Resources and Evaluation (LREC), vol. 2018, p. 156. NIH Public Access (2018)
7. Collberg, C., Proebsting, T.A.: Repeatability in computer systems research. Communications of the ACM **59**(3), 62–69 (2016)
8. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of the International Conference on Web and Social Media (ICWSM), pp. 512–515 (2017)
9. Drummond, C.: Finding a balance between anarchy and orthodoxy. In: Proceedings of the International Conference on Machine Learning: Workshop on Evaluation Methods for Machine Learning (ICML) (2008)
10. Godbole, S., Bhattacharya, I., Gupta, A., Verma, A.: Building re-usable dictionary repositories for real-world text mining. In: Proceedings of the International Conference on Information and Knowledge Management (CIKM), pp. 1189–1198. ACM (2010)
11. Halder, R., Pal, S., Cortesi, A.: Watermarking techniques for relational databases: Survey, classification and comparison. Journal of Universal Computer Science **16**(21), 3164–3190 (2010)
12. Howe, B.: Cde: A tool for creating portable experimental software packages. Computing in Science & Engineering **14**(4), 32–35 (2012)
13. Janin, Y., Vincent, C., Duraffort, R.: Care, the comprehensive archiver for reproducible execution. In: Proceedings of the SIGPLAN Workshop on Reproducible Research Methodologies and New Publication Models in Computer Engineering, pp. 1:1–1:7. ACM (2014)
14. Kovačević, J.: How to encourage and publish reproducible research. In: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 4, pp. IV–1273. IEEE (2007)
15. Manku, G.S., Jain, A., Das Sarma, A.: Detecting near-duplicates for web crawling. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 141–150. ACM (2007)

16. Napoles, C., Tetreault, J., Pappu, A., Rosato, E., Provenzale, B.: Finding good conversations online: The yahoo news annotated comments corpus. In: Proceedings of the Linguistic Annotation Workshop, pp. 13–23 (2017)
17. Pandit, H., Hamed, R.G., Lawless, S., Lewis, D.: The use of open data to improve the repeatability of adaptivity and personalisation experiment. In: Proceedings of the Conference on User Modelling, Adaptation and Personalization (UMAP Extended Proceedings) (2016)
18. Pedersen, T.: Empiricism is not a matter of faith. Computational Linguistics **34**(3), 465–470 (2008)
19. Pham, Q., Malik, T., Foster, I.T.: Using provenance for repeatability. In: Proceedings of the Workshop on the USENIX Theory and Practice of Provenance, pp. 2:1–2:4 (2013)
20. Rozier, K.Y., Rozier, E.W.D.: Reproducibility, correctness, and buildability: The three principles for ethical public dissemination of computer science and engineering research. In: Proceedings of the International Symposium on Ethics in Engineering, Science, and Technology (ETHICS), pp. 1–13. IEEE (2014)
21. Schabus, D., Skowron, M., Trapp, M.: One million posts: A data set of german online discussions. In: Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR), pp. 1241–1244 (2017)
22. Sonnenburg, S., Braun, M.L., Ong, C.S., Bengio, S., Bottou, L., Holmes, G., LeCun, Y., Müller, K.R., Pereira, F., Rasmussen, C.E., et al.: The need for open source software in machine learning. Journal of Machine Learning Research **8**(Oct), 2443–2466 (2007)
23. Vandewalle, P., Kovacevic, J., Vetterli, M.: Reproducible research in signal processing. Signal Processing Magazine **26**(3), 37–47 (2009)
24. Vitek, J., Kalibera, T.: Repeatability, reproducibility, and rigor in systems research. In: Proceedings of the International Conference on Embedded Software (EMSOFT), pp. 33–38. ACM (2011)
25. Wulczyn, E., Thain, N., Dixon, L.: Ex machina: Personal attacks seen at scale. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 1391–1399. International World Wide Web Conferences Steering Committee (2017)