# A preliminary study on similarity-preserving digital book identifiers

**Klemo Vladimir**[1], **Marin Silic**[1], **Nenad Romic**[2], **Goran Delac**[1], **and Sinisa Srbljic**[1]

[1]University of Zagreb, Faculty of Electrical Engineering and Computing
Consumer Computing Lab, Unska 3, 10000 Zagreb, Croatia
{klemo.vladimir, marin.silic, goran.delac, sinisa.srbljic}@fer.hr

[2]Leuphana Universität Lüneburg, DCRL Digital Cultures Research Lab
Am Sande 5, 21335 Lüneburg, Germany
ki.ber@kom.uni.st

## Abstract

Due to proliferation of digital publishing, e-book catalogs are abundant but noisy and unstructured. Tools for the digital librarian rely on ISBN, metadata embedded into digital files (without accepted standard) and cryptographic hash functions for the identification of coderivative or near-duplicate content. However, unreliability of metadata and sensitivity of hashing to even smallest changes prevents efficient detection of coderivative or similar digital books. Focus of the study are books with many versions that differ in certain amount of OCR errors and have a number of sentence-length variations. Identification of similar books is performed using small-sized fingerprints that can be easily shared and compared. We created synthetic datasets to evaluate fingerprinting accuracy while providing standard precision and recall measurements.

## 1 Introduction

The need and then creation of a system to identify every particular book in an archive or repository has a long history. An invention and iterative development of a card catalog, as we know it today, a universal discrete machine which stores, processes and transfers data took several centuries (Krajewski, 2011). However, only in late 1960s, when computer technology began to become an important part of trade, publishers came up with a standardized numeric identifier describing (only) a geographical or language area, publisher and a specific edition and title of the book.

It's hard to imagine a book today which is not prepared and processed as a digital file before it gets published. Still, the unique book identifier in use is created by (and for) bureaucracy and as a consequence it only reflects book's context related to commerce - nothing else (ISBN Information, 2015).

Today's available digital books are coming from many different sources: comprehensive scanning projects like the Internet archive or the National Library of Norway, community driven repositories like Library Genesis, Aaaaarg.org, Monoskop.org, Ubu.com or commercial providers like Amazon, Google or Apple.

There is contextual information already embedded in the content of every digital book which could improve and optimize file storage (detection of duplicates), network transfer (detection of network peers), classification, topic clustering, language analysis and more. We envision a different kind of digital book identifier which will embed and carry much more of its relevant context than what is the case with existing ones.

In this paper we present a feasibility study of using locality sensitive hashing for construction of similarity-preserving digital book identifiers. In order to evaluate the suggested approach, we have constructed a comprehensive dataset that contains highly similar book entries[1]. Proposed identifiers can be used in practice for scalable comparison of books, retrieval of near-duplicate books or as an index for metadata provisioning services that tolerate different e-book formats, imperfect metadata or minor changes on text itself.

The rest of the paper is organized as follows. Next section gives an overview of related work. Section 3 describes construction and structure of the dataset used in the experiments. Implementation and characteristics of similarity preserving fingerprinting are presented in Section 4. Section 5 presents and discusses experimental results. Fi-

[1]Dataset and code available at http://ccl.fer.hr/ds/2015/readme.html



*Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 78–83,
Beijing, China, July 30, 2015. 

nally, Section 6 concludes the paper and proposes future research directions.

## 2 Related work

A general overview of two dominant approaches for the identification of near-duplicate documents, ranking and fingerprinting, is presented in (Hoad and Zobel, 2003). The ranking relies on vector space models where documents are represented using high-dimensional vectors. Document fingerprinting is used to create compact representation of document vectors using hashing functions. A number of methods were proposed and evaluated for features of various resolutions, such as characters, words or sentences (Manber and others, 1994; Shivakumar and Garcia-Molina, 1995; Brin et al., 1995).

The dimensionality of document vectors can be reduced using locality sensitive hashing, mostly using simhash or min-hash algorithms. Min-hash (Broder, 1997) was used for large-scale detection of similar books at the page level (Spasojevic and Poncin, 2011). While min-hash uses many hash values to represent a document, having each value computed with a different hash function, simhash gives a more compact output by reducing document vectors to a small sized real-valued fingerprints (Charikar, 2002). Simhash was successfully evaluated for duplicate detection of web pages (Manku et al., 2007; Henzinger, 2006), code segments (Uddin et al., 2011), short messages (Pi et al., 2009), spam (Ho et al., 2014) and academic papers (Williams and Giles, 2013). Our contribution to the literature is in the use of simhash fingerprinting for larger texts in form of digital books.

Partial duplicates detection in large collections of scanned books was proposed in (Yalniz et al., 2011). Here, books were represented by a sequence of unique words and duplicates were identified by the longest common sub-sequence alignment. However, book representation using unique words is still to large to be useful as an identifier, e.g. for a 100k words book there are $2-3$k unique words.

Other approaches rely on hashing metadata contents only (Padmasree et al., 2006; Voß et al., 2009). Near-duplicate detection based on metadata is also well researched in the field of record linkage where matching of records that relate to the same entities from several databases is studied (Christen, 2012). However, primary motivation for this preliminary study is to derive similarity book identifier based on content, not the metadata.

## 3 Dataset construction

Synthetic book collections were generated to evaluate book fingerprints constructed using locality sensitive hashing. Datasets were generated by "mutating" referent, or "seed", books taken from the *Project Gutenberg* website (Project Gutenberg, 2015). We randomly sampled books from the larger collection of available books and preprocessed so that only raw text files without any additional data remain.

### 3.1 Synthesis methodology

For each canonical book a random number of mutations were performed. There are two main types of mutations: (1) OCR errors and (2) random text mutations.

(1) Introduction of OCR (Optical Character Recognition) errors simulates existence of multiple versions of the single book scanned and post-processed by different equipment, different software stack or different librarians. Following the previous work (Reynaert, 2011; Feng and Manmatha, 2006; Esakov et al., 1994), mutations were created by building a custom discrete distribution of basic "errors" derived from common OCR character confusions. For example, the most common character-level mistake is the insertion of space. Other included mistakes were random character insertion, replacement of a single character with another character or a number of characters, and merging of two characters into single character (e.g. $\texttt{rn} \rightarrow \texttt{m}$). As an illustrative example, a random text *"the immortality of the soul"* with $3\%$ character-level corruption rate (at book level) becomes *"1he immttality of th e soul"*. Character-error rates reported in the literature range from $0.5-10\%$ (Esakov et al., 1994; Yalniz et al., 2011; Abdulkader and Casey, 2009).

(2) In addition to character-level mutations, a certain amount of sentence-level mutations were introduced. Addition or removal of random sentences can simulate book annotations, bookmarks or metadata insertions performed by editors, readers or e-book authoring or reading software. Addition of new material was sourced from a random book in the collection while keeping the length of the corrupted text similar to the canonical text.

### 3.2 Dataset structure

We compiled two datasets by corrupting texts with different corruption rates. Smaller, *1k* dataset was used for the exploration of fingerprinting parameters. The 1k dataset was generated from the seed of 100 distinct books where another 9 versions were derived from each canonical book using mutations described in the previous section. Additionally, 1k dataset was replicated for various corruption rates.

Larger, *75k* dataset was used to evaluate quality of the generated fingerprints, as well as performance in terms on execution time. The 75k dataset (28GB uncompressed) was generated from the seed of 9k distinct books where a random number of derivatives were created in range $1-15$ with random corruption rates in range $0-5\%$.

## 4 Similarity preserving fingerprints

We chose to work with simhash fingerprints because of it's compactness and simplicity. In order to create a simhash fingerprint for the given book, a clear text must be extracted from the book file and converted to a set of features. Since our setting does not resemble traditional monolithic database, but rather a set of distributed libraries, our approach is not able to use *inverse document frequency* (IDF) analysis[2]. Thus, feature vector extraction is minimal and consists of identifying lower-cased terms as character n-grams and counting term occurrence.

### 4.1 Fingerprint computation

An arbitrary hash length $n$[3] is selected and $n$-dimensional fingerprint vector *sh* is initialized to all zeroes. In order to calculate a fingerprint, every feature in the feature vector $f$ is hashed to a $n$-bit digest $h(f_i)$ using an arbitrary (cryptographic) hash function. Bit $b$ at the position $j$ in the computed digest $h(f_i)$ impacts the value at the same position in $sh$ vector as follows: if $b$ is 1, then $sh[j]$ is incremented by the weight of feature $f_i$; if not, then $sh[j]$ is decremented by the same weight. Weight of feature is equal to term occurrence calculated for a given feature in the feature extraction phase. The final fingerprint is calculated by reducing vector $sh$ to a $n$-bit number where bit at position $i$ is determined by the sign of the $i$-th element in the $sh$.

### 4.2 Fingerprint similarity

Books that differ in small number of characters or words will have fingerprints that differ in a small number of bits. In order to illustrate this property of the simhash fingerprint, we corrupted three books of different sizes at a corruption rate in range $1-10\%$. Difference for 128-bit fingerprints (in Hamming distance) between canonical book and each corrupted version is presented in Fig. 1.

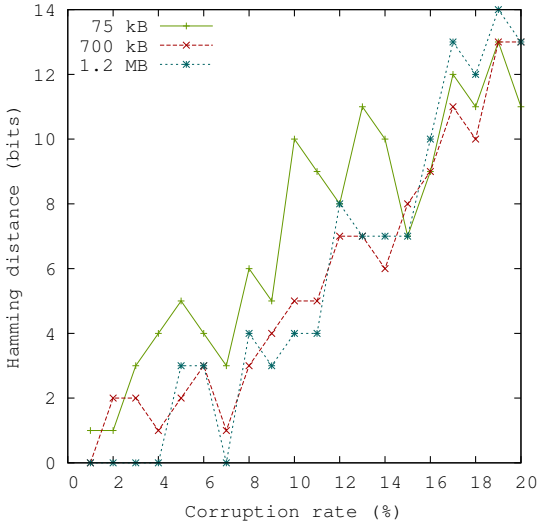


Figure 1: Impact of corruption rate on fingerprint similarity for books with different text lengths

Results show that distance between canonical and derivative texts correlates with corruption rate. Moreover, figure indicates that fingerprints of smaller books are more sensitive to text variations[4].

## 5 Experimental results

In this section we present and discuss evaluation methodology and results. First, we describe preliminary evaluation of the proposed method using k-means clustering on the smaller dataset. Clustering is used to confirm, in a very simple and intuitive way, that coderivative book fingerprints group (or "gravitate") well around known canonical books. In a realistic setting a number of canonical books and their distribution is unknown. Thus, we use efficient bucket-based similarity queries to

[2] IDF requires access to global collection

[3] We used hash lengths between 64 and 256 bits with the step of 32

[4] We plan to address this issue in future work with the encoding of size information in the fingerprint itself

identify coderivative books on the larger dataset with variable number of coderivative books.

### 5.1 Fingerprint clustering

Initial feasibility test and exploration of design parameters for the proposed similarity preserving fingerprints is performed using k-means clustering on the smaller *1k* dataset. The main idea is not to identify coderivative books using clustering, but to test how well the proposed method groups books based only on distance between corresponding fingerprints. Generated book fingerprints are converted from an integer representation to a feature vector of zeros and ones. Finally, book fingerprints are clustered using k-means algorithm where the number of clusters equals the number of canonical books. Standard accuracy measures are calculated based on the difference between the obtained clusters and the gold truth cluster information.

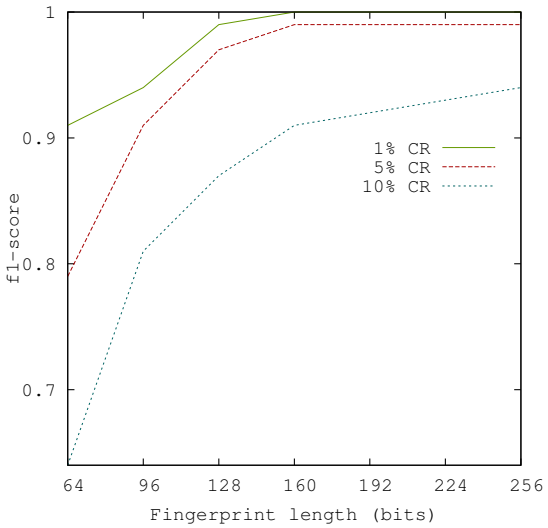


Figure 2: Clustering accuracy for different fingerprint lengths on the 1k dataset

Results for different fingerprint lengths and corruption rates are presented using $F_1$ score on Fig. 2. Please note that we have evaluated different n-gram lengths for feature extraction on 1k dataset, of both characters and words, and character-level approach outperformed word-level approach (also observed in (Spasojevic and Poncin, 2011)). Best results were obtained with character n-grams of size 4 (we did not include these results due to limited space). It is clear that accuracy grows as fingerprint length increases and corruption rate decreases. Results suggest that 128-bit fingerprints achieve satisfactory accuracy ($F_1 = 0.97$) for the average corruption rate of $5\%$.

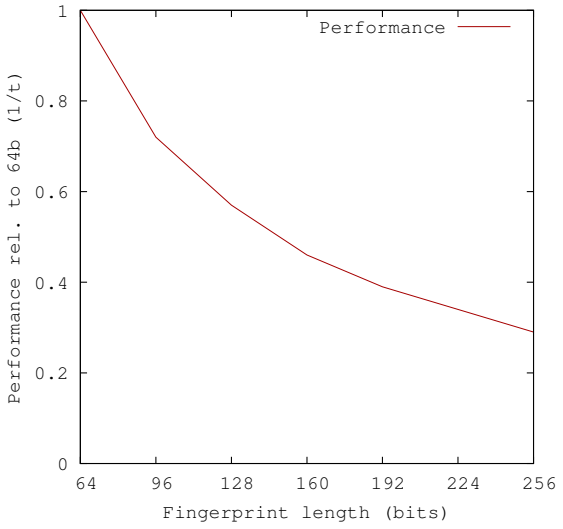


Figure 3: Performance drop for the increasing fingerprint lengths on the 1k dataset

However, note that there is a trade-off between performance and quality of results. Performance, defined as an inverse of the fingerprint generation execution time in minutes related to the 64-bit fingerprint, expectedly drops with the increase in fingerprint length (Fig. 3).

### 5.2 Similarity queries

In the real-world setting a number of clusters and distribution of books per cluster are unknown. Thus, evaluation of the proposed algorithm with 128-bit fingerprints is evaluated on the larger *75k* dataset that is generated with the intention to resemble real-world digital library, i.e. number of coderivative books is not fixed. In order to analyze Hamming distance thresholds for coderivative books, instead of clustering a brute-force similarity queries are run over whole dataset. That is, for every book a set of other books from the dataset is identified whose fingerprints have maximum distance of $d$ bits. Since brute-force querying over the whole dataset has $O(n^2)$ time complexity, we have implemented a bucketing algorithm that significantly reduces execution time with minimal accuracy penalty. Fingerprints are divided into an arbitrary number of bands, and a pair of fingerprints are considered candidates for similarity only if they are identical in at least one band (Rajaraman and Ullman, 2011). Precision and recall are calculated for every query as a num-

ber of coderivative books divided by the number of returned results or number of expected results, respectively. Figure 4 presents precision and recall graphs for various $d$ for both brute-force and bucketing versions.

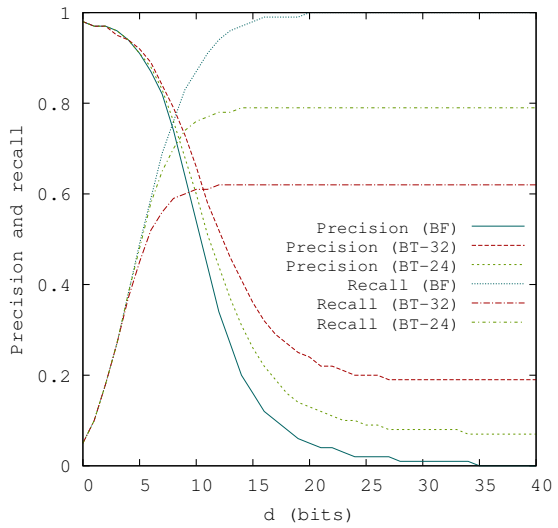


Figure 4: Precision and recall for various $d$ on the 75k dataset for brute-force queries (BF) and bucketing (BT) algorithms

Brute-force queries over the whole dataset achieve the best $F_1$ score of $0.75$ at $d = 7$ bits but also have worst average execution time of $29.73$ minutes[5]. The bucketing version was tested with different band lengths of 24 and 32 bits, respectively. Best $F_1$ score for the 24-bit band was $0.73$ at $d = 7$, which is tolerable ($2.6\%$ lower precision compared to brute-force) since average execution time is reduced to only $1.22$ minutes. With the increased band lengths accuracy decreases but execution time significantly drops, e.g. 32-bit band version achieves $F_1$ of $0.67$ with the execution time of only $8.55$ seconds. However, note that bucketing algorithms can not achieve high recall since some candidates which are not identical in any band never get a chance to be compared. Such performance suggests that bucketing algorithms, with some implementation improvements, could be used for real-time detection of the top-k near duplicates.

[5]All the experiments were conducted on an Intel Core 2 Quad 2.66GHz CPU with 8GB of memory, running Ubuntu 14.04 LTS

## 6 Conclusions and future work

In conclusion, we described an application of the simhash algorithm for generation of similarity-preserving digital book fingerprints derived from the content of the book. We further evaluated our proposed method on the synthetic dataset which was generated by randomly mutating canonical books. Books were mutated at different rates with various mutations that simulate real-world noisy libraries. Preliminary results suggest that proposed techniques could be useful for the identification of coderivative books.

Traditional book identifiers, in form of ISBN numbers, embed metadata (geographical area, publisher, title etc.) and, being only 13 digits long, enable efficient transfer and computer processing. In addition to these benefits, proposed similarity-preserving fingerprints enable quick calculation of the semantic distance between any two books in the universe of all digital books. A combination of these is the apparatus for approaching chaotic world of digital file repositories in the age of the Internet. Resulting composite book identifier, comprised of both metadata (ISBN) and identifiers derived from content, could be part of future infrastructure based on peer-to-peer distributed heterogeneous network or a centralized service provided by the institutions.

In addition to composite book identifier, future explorations will include detection of different editions or translations of a single book and application of similar methods for books comprised of mostly images. Moreover, we are working on crawlers for amateur libraries and public archives with the goal of collecting a larger real-world dataset.

## Acknowledgments

This work was supported in part by the Croatian science foundation through the Recommender System for Service-oriented Architecture research project and in part by Leuphana Universität Lüneburg, DCRL Digital Cultures Research Lab. The authors would like to thank Robert M. Ochshorn and Goran Glavaš for their invaluable comments and suggestions and Project Gutenberg for their book collection.

## References


Ahmad Abdulkader and Mathew R Casey. 2009. Low cost correction of ocr errors using learning in a multi-engine environment. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*, pages 576–580. IEEE.

Sergey Brin, James Davis, and Hector Garcia-Molina. 1995. Copy detection mechanisms for digital documents. In *ACM SIGMOD Record*, volume 24, pages 398–409. ACM.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE.

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

Peter Christen. 2012. A survey of indexing techniques for scalable record linkage and deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1537–1555.

Jeffrey Esakov, Daniel P Lopresti, and Jonathan S Sandberg. 1994. Classification and distribution of optical character recognition errors. In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pages 204–216. International Society for Optics and Photonics.

Shaolei Feng and R Manmatha. 2006. A hierarchical, hmm-based automatic evaluation of ocr accuracy for a digital library of books. In *Digital Libraries, 2006. JCDL'06. Proceedings of the 6th ACM/IEEE-CS Joint Conference on*, pages 109–118. IEEE.

Monika Henzinger. 2006. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284–291. ACM.

Phuc-Tran Ho, Hee-Sun Kim, and Sung-Ryul Kim. 2014. Application of sim-hash algorithm and big data analysis in spam email detection system. In *Proceedings of the 2014 Conference on Research in Adaptive and Convergent Systems*, pages 242–246. ACM.

Timothy C Hoad and Justin Zobel. 2003. Methods for identifying versioned and plagiarized documents. *Journal of the American society for information science and technology*, 54(3):203–215.

ISBN Information. 2015. `http://isbn-information.com/history-of-the-isbn-system.html`. Accessed: 2015-05-05.

Markus Krajewski. 2011. *Paper machines: about cards & catalogs, 1548-1929*. MIT Press.

Udi Manber et al. 1994. Finding similar files in a large file system. In *Usenix Winter*, volume 94, pages 1–10.

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*, pages 141–150. ACM.

L Padmasree, Vamshi Ambati, J Chandulal, M Rao, and Regional Mega Scanning Center. 2006. Signature based duplication detection in digital libraries. *Signature*, 10001011:00001100.

Bingfeng Pi, Shunkai Fu, Weilei Wang, and Song Han. 2009. Simhash-based effective and efficient detecting of near-duplicate short messages. In *Proceedings of the 2nd Symposium International Computer Science and Computational Technology*. Citeseer.

Project Gutenberg. 2015. `http://www.gutenberg.org`. Accessed: 2015-05-01.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Martin WC Reynaert. 2011. Character confusion versus focus word-based correction of spelling and ocr variants in corpora. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(2):173–187.

Narayanan Shivakumar and Hector Garcia-Molina. 1995. Scam: A copy detection mechanism for digital documents.

Nemanja Spasojevic and Guillaume Poncin. 2011. Large scale page-based book similarity clustering. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 119–125. IEEE.

Md Sharif Uddin, Chanchal K Roy, Kevin A Schneider, and Abram Hindle. 2011. On the effectiveness of simhash for detecting near-miss clones in large scale software systems. In *Reverse Engineering (WCRE), 2011 18th Working Conference on*, pages 13–22. IEEE.

Jakob Voß, Hotho Andreas, and Jäschke Robert. 2009. Mapping bibliographic records with bibliographic hash keys.

Kyle Williams and C Lee Giles. 2013. Near duplicate detection in an academic digital library. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 91–94. ACM.

Ismet Zeki Yalniz, Ethem F Can, and R Manmatha. 2011. Partial duplicate detection for large book collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 469–474. ACM.