Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2021, vol. 26, no. 2, pp. 178–182
https://doi.org/10.2478/acss-2021-0022
https://content.sciendo.com

Evaluation of Fingerprint Selection Algorithms for Two-Stage Plagiarism Detection

Gints Jēkabsons*
Riga Technical University, Riga, Latvia

***Abstract* –Generally, the process of plagiarism detection can be divided into two main stages: source retrieval and text alignment. The paper evaluates and compares effectiveness of five fingerprint selection algorithms used during the source retrieval stage: *Every p-th*, *0 mod p*, *Winnowing*, *Frequency-biased Winnowing* (*FBW*) and *Modified FBW* (*MFBW*). The algorithms are evaluated on a dataset containing plagiarism cases in Bachelor and Master Theses written in English in the field of computer science. The best performance is reached by *0 mod p*, *Winnowing* and *MFBW*. For these algorithms, reduction of fingerprint size from 100 % to about 20 % kept the effectiveness at approximately the same level. Moreover, *MFBW* sends overall fewer document pairs to the text alignment stage, thus also reducing the computational cost of the process. The software developed for this study is freely available at the author's website http://www.cs.rtu.lv/jekabsons/.**

***Keywords* –Document fingerprinting, fingerprint selection, indexing, plagiarism detection, text alignment, text reuse detection.**

I. Introduction

Generally, the process of plagiarism detection can be divided into two main stages: source retrieval and text alignment [1]. At the source retrieval stage, given a query document and a large collection of stored documents, the task is to retrieve all potential sources from which the query document might have "borrowed" the text. At the text alignment stage, the task is to identify contiguous passages of reused text between the query document and the retrieved sources. The reused text might span the whole document, some paragraphs, or just one sentence; the text might be modified by inserting, replacing, removing, or rearranging words or sentences to obfuscate the fact of copying.

One of the established approaches for the source retrieval stage is document fingerprinting [2]–[11]. In this approach, fingerprints are extracted from documents through hashing sequences of consecutive words or characters. Two documents sharing their hashes indicates a possible text reuse.

Storing and handling large sets of fingerprints can often be impractical; thus, various algorithms for fingerprint selection have been proposed. Some subsets of these fingerprint selection algorithms have been empirically evaluated and compared on text reuse detection in news stories [5], [12], books [3], blogs and other websites [4], as well as in Bachelor and Master Theses [11].

In the latest study [11], the algorithm evaluation process involves a source retrieval stage of a retrieval system but does not involve a text alignment stage. To broaden the applicability of results, the current study builds on the research conducted in [11] by extending it to a two-stage retrieval system. The effectiveness of five fingerprint selection algorithms is evaluated and compared.

The rest of the paper is organised as follows: Section II describes the implementation of a two-stage local text reuse detection system. Section III describes experimental setup for comparing the algorithms. Section IV presents the results. Finally, Section V concludes the paper.

II. Implementation of a Two-Stage System

A. Fingerprint Extraction and Selection

To extract a sequence of fingerprints from a document, first, some pre-processing is done. This study follows the best results in [11] and performs the same pre-processing steps in the following order: A text is case-folded and then tokenized by non-alphanumeric characters into a sequence of words. Next, common stop words and any words shorter than three characters are removed from the sequence. Finally, each remaining word is stemmed using the well-known Porter stemmer.

Result of the pre-processing is a clean sequence of normalized words. It is then divided into overlapping n-grams of n consecutive words. Each n-gram is converted into an integer using 32-bit FNV-1a hash function [13].

The total number of n-grams for a document with l words is $m = l - n + 1$. The simplest fingerprinting strategy is "full fingerprinting" where all n-grams (i.e., technically – their computed hashes) become document fingerprints. While this strategy is expected to be the best case for finding as many correct text reuse sources as possible, it also requires the largest amount of storage space. Thus, the strategy is too expensive for use with large collections even if an inverted index is used.

Fingerprint selection algorithms are designed to take only a representative subset of the whole set of n-grams to lower the fingerprint storage requirements while at the same time trying to maintain the quality of results. This is achieved through some ideas for how to consistently select approximately the same fingerprints in source documents and in query documents. The

*Corresponding author's e-mail: gints.jekabsons@rtu.lv

interested reader is referred to summarisation of those ideas in [4], [5], [11].

The present paper focuses on the following five fingerprint selection algorithms: *Every p-th*, *0 mod p* [10], *Winnowing* [8], *Frequency-biased Winnowing* (*FBW*) [3] and *Modified Frequency-biased Winnowing* (*MFBW*) [11].

Each of the algorithms has one parameter controlling the number of selected fingerprints. The parameter can be used to strike a trade-off between quality of results and storage requirements. This also allows comparing the effectiveness of the algorithms at fixed storage size. For *Every p-th* and *0 mod p*, the number of selected fingerprints can be computed as m / p, where p is the parameter. For the three *Winnowing*-based algorithms, the number is approximately $2m / (w + 1)$, where w is the parameter. For instance, to select 5 % of fingerprints, p has to be set to 20, and w has to be set to 39.

B. Fingerprint Indexing and Source Retrieval

A retrieval subsystem of a text reuse detection system can be implemented using an inverted index. In such an index, document fingerprints, i.e., the sequence of hashes, can be indexed in the same way as in classic document indexing where the documents are treated as a sequence of words or word n-grams. Each hash is put into the index, while maintaining a list of documents where it can be found. After the collection of documents are indexed, query document hashes can be searched in the index one-by-one and potential source documents retrieved.

In this study, an inverted index is implemented using the well-known Lucene library [14] version 8.8.0 with the default configuration. For each searched hash, the one first returned document is taken as the potential source and sent to the aligner.

C. Text Alignment

The alignment stage of a text reuse detection system involves detailed comparison of the query document and each retrieved potential source document that was found during the source retrieval stage. The system should identify contiguous passages of reused text between the query document and the source. This can filter out some of the false positives among the retrieved sources if no similar passages are identified and ultimately allows highlighting the matching passages for the user.

For text aligning, the present study uses the winner system of PAN 2014 text alignment competition [1]. The system is developed by Sanchez-Perez et al. [15], [16] and is provided for download by one of the authors in a form of source code [17]. The authors of the aligner later optimized the algorithm parameters in [18]. In the current study, the parameter set denoted as "simpler" is used; other provided parameter sets are too dataset-specific (i.e., have too many assumptions about the data, such as a very specific way in recognising text summary obfuscation, or have lower thresholds for sentence similarity, which is expected to bring more false positives).

In general terms, the text alignment in [15]–[17] is a process comprising three stages: seeding, extension and filtering. During the seeding stage, each sentence of query document is compared with each sentence of source document, and a set of similar sentence pairs, called *seeds*, is retrieved. During the extension stage, a recursive algorithm is used to join the seed sentences into larger passages of text being similar between query and source documents. Finally, the filtering stage removes overlapping passages and passages that are too short.

III. Experimental Setup

A. The Dataset

To evaluate the fingerprint selection algorithms, the dataset created and described in [11] is used. It contains 35 query documents and a collection of 1021 source documents where 348 are used sources (i.e., parts of their text are reused in query documents) and 673 are unused sources. The query documents are Bachelor and Master Theses. The source documents are scientific articles, technical reports, websites, theses, books and presentation slides. All documents are written in English on a variety of computer science topics. On average, each query document contains reused text from about 10 sources. For text reuse detection, there are a total of $35 \times 1021 = 35\,735$ document pairs.

B. Parameters

There are two parameters to be set. The first parameter is the n-gram size n. Smaller n generally increases the number of false matches among documents, while larger n makes fingerprinting more sensitive to changes in text since exact matches of less than n words in a row cannot be detected. The second parameter is the parameter of a fingerprint selection algorithm controlling the number of selected fingerprints (as discussed in Section II A).

C. Evaluation Measures

For evaluation of algorithms and parameter choices, the well-known *precision*, *recall* and *F-score* measures are used:

$$precision = \frac{|true\ positives|}{|true\ positives| + |false\ positives|}, \quad (1)$$

$$recall = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|}, \quad (2)$$

$$F_\beta = \left(1 + \beta^2\right) \frac{precision \cdot recall}{\left(\beta^2 \cdot precision\right) + recall}, \quad (3)$$

where β determines the balance between precision and recall, i.e., it specifies a ratio of how much one is willing to tolerate a decrease in precision to increase in recall.

In [11], the evaluation of fingerprinting selection algorithms was done without the text alignment stage and it was argued that if the alignment stage were added, the aligner would be able to discard many of the false positive documents retrieved. Therefore, β was set to 10 (meaning recall was weighted 10 times higher than precision). In the present study, the algorithms are evaluated in a two-stage system and β can be considerably lowered. However, it can be argued that higher

recall is still slightly more important than higher precision and therefore β is set to 2.

It should also be noted that, while this study involves text alignment, the focus is on source retrieval evaluation and, therefore, the precision, recall, and F_2 are computed at the document level, not at the text passage level, i.e., whether a case is positive or negative depends on whether the document is or is not retrieved and then kept by the aligner (because at least one passage is found to be matching).

In addition to the measures above, it is also important to take into account how many document pairs end up being fed to the aligner. While in practical applications source retrieval usually is computer disk intensive process, text aligning is computer processor intensive and should be reduced as well.

IV. Results

A. F_2 vs. Number of the Selected Fingerprints

Since the goal of fingerprinting selection algorithms is to reduce the number of selected fingerprints while preserving high effectiveness, F_2 is first viewed as a function of the number of fingerprints (which directly corresponds to the index size). The parameter of each algorithm is set accordingly and then n-gram size n is chosen for maximum F_2. Table I shows parameter value and n value for each algorithm, precision, recall and F_2 values obtained before the alignment stage, the number of document pairs fed to the aligner, as well as the final precision, recall and F_2 values (optimized by choosing n). The same final F_2 values are also shown in Fig. 1 (with additional fingerprinting size of one third).

First, looking at results obtained before the alignment stage, they are overall in line with those obtained in [11]. The best performance is achieved by *0 mod p*, *Winnowing*, and *MFBW*. To keep the *F*-score high, reduction of fingerprint size requires smaller n. For *FBW* and *Every p-th* algorithms this aspect is more extreme as those algorithms lose their effectiveness faster than others.

TABLE I
Best Performance of Full Fingerprinting and Selection Algorithms at Four Different Fingerprint Sizes

Algorithm	Parameters		Before alignment stage			# of document pairs to align	After alignment stage		
	p* or *w	***n***	**Prec. (%)**	**Rec. (%)**	**F_2 (%)**		**Prec. (%)**	**Rec. (%)**	**F_2 (%)**
All fingerprints									
Full fingerprinting	N/A	5	28.82	91.93	63.93	1107	61.25	87.90	80.86
Number of fingerprints: ~50 % of full fingerprinting									
Every p-th	$p = 2$	5	46.53	79.25	69.48	591	70.51	75.79	74.67
0 mod p	$p = 2$	5	40.72	88.47	71.66	754	71.78	85.01	81.99
Winnowing	$w = 3$	4	23.87	95.97	59.83	1395	52.66	91.35	79.65
FBW	$w = 3$	4	57.48	85.30	77.77	515	73.51	81.56	79.81
MFBW	$w = 3$	5	43.24	87.61	72.69	703	71.50	83.86	81.06
Number of fingerprints: ~20 % of full fingerprinting									
Every p-th	$p = 5$	3	13.72	86.46	41.96	2187	44.24	81.84	69.95
0 mod p	$p = 5$	4	33.91	89.63	67.46	917	63.46	85.59	80.01
Winnowing	$w = 9$	4	37.50	85.59	68.12	792	65.66	81.56	77.79
FBW	$w = 9$	2	5.97	93.66	23.80	5441	36.61	89.05	69.22
MFBW	$w = 9$	4	44.58	85.30	72.12	664	67.79	81.27	78.16
Number of fingerprints: ~10 % of full fingerprinting									
Every p-th	$p = 10$	2	3.98	88.18	16.85	7694	33.80	83.57	64.56
0 mod p	$p = 10$	3	15.81	90.49	46.53	1986	49.18	86.17	74.90
Winnowing	$w = 19$	3	18.45	89.91	50.67	1691	50.95	85.01	74.99
FBW	$w = 19$	1	3.32	93.37	14.54	9753	38.28	88.47	70.09
MFBW	$w = 19$	3	23.32	87.90	56.57	1308	58.75	84.15	77.45
Number of fingerprints: ~5 % of full fingerprinting									
Every p-th	$p = 20$	2	6.13	65.71	22.32	3720	39.06	62.25	55.64
0 mod p	$p = 20$	3	23.13	80.40	53.78	1206	57.66	78.10	72.93
Winnowing	$w = 39$	3	25.86	75.79	54.68	1017	56.85	72.91	69.01
FBW	$w = 39$	1	5.06	83.57	20.37	5731	42.37	79.25	67.50
MFBW	$w = 39$	3	34.18	73.78	59.90	749	69.60	70.61	70.40

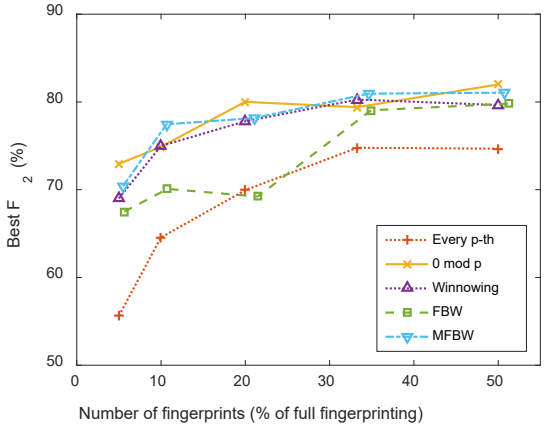

Fig. 1. Best F_2 value vs. the number of selected fingerprints. Full fingerprinting result is not shown – it is located at 100 % fingerprints and F_2 = 80.86 % as given in Table I.

Next, looking at the number of document pairs fed to the aligner it can be seen that overall the number gradually increases before a decrease of fingerprint size reaches about 5 %. This behaviour can be explained by the reduction in n-gram size – the smaller the n-grams used, the more documents will be retrieved and fed to the aligner. However, reaching the 5 % size, the index is missing too much information and fewer documents are retrieved. Here again *0 mod p*, *Winnowing* and *MFBW* are the winners, especially *MFBW* since overall it retrieves fewer documents and still has good recall. This can be explained by the fact that *MFBW* tends to select fingerprints in a document that are less frequent in the collection of documents [11].

Finally, analysing precision, recall and F_2 obtained after the alignment stage, it can be observed how precision is inversely correlated with the number of retrieved documents, while recall is quite stable before the number of fingerprints reaches 5 %. As a result, with the reduction of fingerprint size, F_2 is falling slowly and smoothly, while somewhere between 5 % and 10 % the fall is accelerating.

In addition, it is also important to note a clear increase of precision before alignment and precision after alignment by about 30 percentage points in all cases. This is thanks to the aligner discarding documents without any detected similar passages.

It can be concluded that in the used dataset with the three best algorithms the number of fingerprints (and therefore index size) can be reduced to one fifth without losing much of effectiveness.

B. Precision vs. Recall

Fig. 2 shows precision-recall curves when 50 % and 5 % of fingerprints are selected. *0 mod p*, *Winnowing* and *MFBW* are overall the best ones and have very similar behaviour while the performance of *Every p-th* and *FBW* deteriorates much faster.

It can also be seen that it is not difficult to outperform full fingerprinting in terms of precision, but this is achieved at the expense of recall.

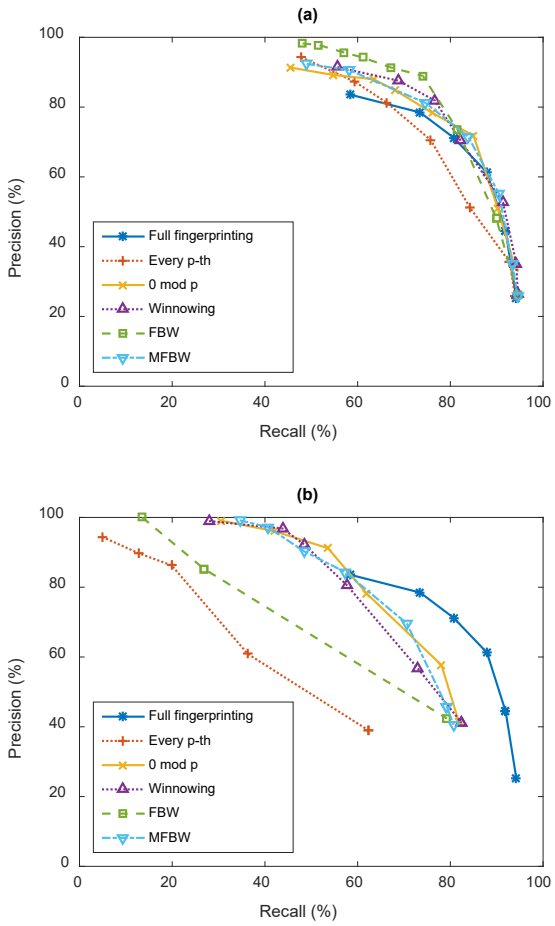

Fig. 2. Precision-recall curves when (a) 50 % and (b) 5 % of fingerprints are selected. Full fingerprinting is shown for reference only since it always selects 100 % of fingerprints.

C. Passage-Level Evaluation

In addition to the dataset described in Section III A (hereafter referred to as D1) which was used in the experiments in Sections IV A and IV B, another dataset (D2) was created for additional experiments with the aligner. D2 was created as a subset of D1. The documents of D2 have precisely marked reused text passages and therefore can be used for passage-level evaluation of text reuse detection. D2 contains 18 query documents and 378 collection documents. On average, each query document contains reused text from about six sources. In total, 378 pairs of documents have to be aligned.

The evaluation is done using precision, recall and F_1 measures computed in accordance to [1] using the same aligner parameters as in previous experiments of this study. The obtained micro average of the measures is 50.36 %, 80.56 %, 61.98 %, respectively, while macro average is 34.10 %, 73.14 %, 46.51 %, respectively. As can be seen, the aligner mostly suffers from low precision, i.e., too many of its marked passages are false positives. Therefore, it is possible that replacing this aligner with another one having much better precision might considerably enhance the results obtained in Sections IV A and IV B.

V. Conclusion

Five different fingerprint selection algorithms have been evaluated and compared. Overall, the best results were obtained by *0 mod p*, *Winnowing* and *MFBW*. While for full fingerprinting F_2 was about 81 %, reduction of fingerprint size to only one fifth kept the effectiveness at approximately the same level. Further reduction smoothly reduced F_2 to about 70 % when one twentieth of fingerprints was used.

It should also be noted that reduction of fingerprint size required to use smaller n-grams, which resulted in more retrieved documents sent to the aligner for detailed analysis. This requires more computational resources as well as brings more false positives. The results show that the aligner has insufficient resistance against false positives. Development of effective and efficient aligners is still a necessary research.

References

[1] M. Potthast, M. Hagen, A. Beyer, M. Busse, M. Tippmann, P. Rosso, and B. Stein, "Overview of the 6th International competition on plagiarism detection," in *CEUR Workshop Proceedings*, vol. 1180, 2014, pp. 845–876.

[2] D. T. Citron and P. Ginsparg, "Patterns of text reuse in a scientific corpus," in *Proceedings of the National Academy of Sciences of the USA, PNAS*, vol. 112, no. 1, pp. 25–30, Jan. 2015. https://doi.org/10.1073/pnas.1415135111

[3] Y. Sun, J. Qin, and W. Wang, "Near duplicate text detection using frequency-biased signatures," in *Web Information Systems Engineering (WISE 2013), Lecture Notes in Computer Science*, vol. 8180. Springer, Berlin, Heidelberg, 2013, pp. 277–291. https://doi.org/10.1007/978-3-642-41230-1_24

[4] O. Abdel-Hamid, B. Behzadi, S. Christoph, and M. Henzinger, "Detecting the origin of text segments efficiently," in *WWW'09: Proceedings of the 18th international conference on World wide web*, ACM, New York, NY, USA, 2009, pp. 61–70. https://doi.org/10.1145/1526709.1526719

[5] J. Seo and W. B. Croft. "Local text reuse detection," in *Proceedings of SIGIR'08*, Singapore,ACM, ACM Press, July 2008, pp. 571–578. https://doi.org/10.1145/1390334.1390432

[6] D. Sorokina, J. Gehrke, S. Warner, and P. Ginsparg, "Plagiarism detection in arXiv," Cornell University, Ithaca, NY, USA, Tech. Rep. TR2006-2046, 2006. https://doi.org/10.1109/ICDM.2006.126

[7] T. C. Hoad and J. Zobel, "Methods for identifying versioned and plagiarized documents," *Journal of the American Society for Information Science and Technology*, vol. 54, no. 3, Jan. 2003, pp. 203–215. https://doi.org/10.1002/asi.10170

[8] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: local algorithms for document fingerprinting," in *Proceedings of SIGMOD'03*, June 2003, pp. 76–85. https://doi.org/10.1145/872757.872770

[9] R. A. Finkel, A. B. Zaslavsky, K. Monostori, and H. W. Schmidt, "Signature extraction for overlap detection in documents," in *Proceedings of the 25th Australasian Computer Science Conference, Conferences in Research and Practice in Information Technology*, vol 4, Melbourne, Australia, 2002, pp. 59–64.

[10] U. Manber, "Finding similar files in a large file system," in *WTEC'94: Proceedings of the USENIX Winter 1994 Technical Conference*, USENIX Association, Berkeley, CA, USA, 1994, pp. 1–10.

[11] G. Jēkabsons. "Evaluation of fingerprint selection algorithms for local text reuse detection," *Applied Computer Systems*, vol. 25, no. 1, 2020, pp. 11–18. https://doi.org/10.2478/acss-2020-0002

[12] A. Mittelbach, L. Lehmann, C. Rensing, and R. Steinmetz, "Automatic detection of local reuse," in *Sustaining TEL: From Innovation to Learning and Practice – Proceedings of the 5th European Conference on Technology Enhanced Learning, EC-TEL 2010*, no. LNCS 6383, Springer Verlag, Sep. 2010, pp. 229–244. https://doi.org/10.1007/978-3-642-16020-2_16

[13] G. Fowler, L. C. Noll, K.-P. Vo, D. Eastlake, and T. Hansen, "The FNV non-cryptographic hash algorithm," Internet Engineering Task Force, Internet-Draft, 2019. [Online]. Available on: https://tools.ietf.org/html/draft-eastlake-fnv-17 [Accessed: Apr. 2, 2021].

[14] The Apache Software Foundation, Lucene, 2021. [Online]. Available: https://lucene.apache.org/ [Accessed: Apr. 9, 2021].

[15] M. A. Sanchez-Perez, A. Gelbukh, and G. Sidorov, "Adaptive algorithm for plagiarism detection: The best-performing approach at PAN 2014 text alignment competition," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction – 6th Int. Conf. CLEF Association, CLEF 2015*, Lecture Notes in Computer Science, J. Mothe*et et al*., Eds. vol. 9283, Springer, Nov. 2015, pp. 402–413. https://doi.org/10.1007/978-3-319-24027-5_42

[16] M. A. Sanchez-Perez, A. Gelbukh, and G. Sidorov. "Dynamically adjustable approach through obfuscation type recognition," in *Working Notes of CLEF 2015 –Conference and Labs of the Evaluation forum*, Toulouse, France, Sep. 2015. CEUR Workshop Proceedings, vol. 1391, 2015, pp. 1–10.

[17] M. A. Sanchez-Perez, A. Gelbukh, and G. Sidorov, "Text alignment system for plagiarism detection, version 2.0," 2015. [Online]. Available: https://www.gelbukh.com/plagiarism-detection/PAN-2015/index.html [Accessed: May 19, 2021]

[18] M. A. Sanchez-Perez, A. Gelbukh, G. Sidorov, and H. Gómez-Adorno, "Plagiarism detection with genetic-based parameter tuning,"*International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 1, Art no. 1860006, 2018, pp. 1–23. https://doi.org/10.1142/S0218001418600066

Gints Jēkabsons received his Doctoral degree from Riga Technical University in 2009. At present, he is an Associate Professor and Researcher at the Department of Software Engineering of the Institute of Applied Computer Systems, Riga Technical University. His current research interests include information retrieval, machine learning and natural language processing.
E-mail: gints.jekabsons@rtu.lv
ORCID iD: https://orcid.org/0000-0002-9575-2488