The Search of New Issues in the Detection of Near-duplicated Documents

Hasan Naderi (PHD) [1], Narges salehpour[2], Mohammad Nazari farokhi[3], Behzad Hosseini chegeni[4]

[1]Department of Computer Engineering(IUST)

[2]Department of Computer Engineering(IAU)

[3]Department of Computer Engineering(IAU)

[4]Department of Computer Engineering(IAU)

ABSTRACT:

Identifying the same document is the task of near-duplicate detection. Among the near-duplicate detection algorithms, the fingerprinting algorithm is taken into consideration using in analysis, plagiarism, repair and maintenance of social softwares. The idea of using fingerprints is in order to identifying duplicated material like cryptographic hash functions which are secure against destructive attacks. These functions serve as high-quality fingerprinting functions. Cryptographic hash algorithms are including MD5 and SHA1 that have been widely applied in the file system. In this paper, using available heuristic algorithms in near-duplicate detection, a set of similar pair document are placed in a certain threshold, an each set is indentified according to being near- duplicate. Furthermore, comparing document is performed by fingerprinting algorithm, and finally, the total value is calculated using the standard method.

Keywords: Near-Duplicate Detection, Fingerprinting, Similarity, Heuristics, Shingle, Hash, Random

I. INTRODUCTION

Fingerprinting and watermarking in relational database, is used in order to protect copyright, tamper detection, traitor tracing and keep data integrity. Since the web is full of copied content so, detection duplicate documents is difficult. One of the methods that is used for detecting repeated, is the last editing which is done on the document and, the exact solution can be found by fingerprinting method, and its duplication is detectable. Near-duplicate is also used in many documents of the World Wide Web. Among negative effects on Web search engines can name storage space, waste indicators, and results of the same writings cluttered and plagiarism which are the results of repetition. Aside from the deliberate repetition of content, copying is done randomly in companies, universities or different departments that store documents. A Possible solution for solving the problems mentioned is that reliable knowledge to near-duplicate must be existed.

$$\varphi(D, dq) \geq 1-\epsilon \text{ with } 0<\epsilon<1$$

So that φ in the interval [0, 1] is considered as a similarity function. However, to achieve comprehensiveness each pair of documents must be analyzed. According to that is Dq ⊂ d, the comparison of o (| d |) is needed [1]. The main idea of the similarity between two document dq, d is using of fingerprinting that fingerprint Fd is a set of numbers k from d, on the other words, if we have two document fingerprinting by Fd and Fdq , and dq, d have near-duplication ,the similarity of two documents is estimated as follows via the Jaccard coefficient. (The numbers k that has subscription can be k ≤k):

$$\varphi(D, dq) \geq 1-\epsilon \text{ with } 0<\epsilon<1 \quad \textit{Is close to} \rightarrow \frac{|Fd \cap Fdq|}{|Fd \cup Fdq|}$$

The meaning of fingerprints community is that all documents are in D, in fact X → μ (X), Fd → D. The inverted file index must be X ∈ Fd.In other words X, is included all documents, and also μ (X) is the Postlist of X. To document dq with fingerprinting Fdq, now the collection $Dq \in D$ includes documents that contain at least K, Postlist but provided that X ∈ Fdq and μ (X) is existed. If (Dq) includes all the documents, Fdq shares fingerprints for the minimum number K, according to the (Dq) is as a heuristic approximation of the function Dq recovery depends on the construction of fingerprinting, it is calculated as follows:

$$\text{req}= \frac{\widehat{D}\bar{q} \cap Dq}{Dq}, \text{ Pre}= \frac{\widehat{D}\bar{q} \cap Dq}{Dq}$$

II. MATERIALS AND METHODS

2.1. Construction of fingerprinting

A piece of document into a sequence of n words can be cascaded as Cd. Cd is a collection of all different patches of documents which its size measures form | d |- N and in time o (| d |) is measurable Provided that c ∈ c d and c are one dimension qualifier with a non-zero weight. Three steps in the construction of fingerprinting must be understood [1]:

2.1.1. Dimensions reducing with mapping are realized

This algorithm selects the dimensions of the previous dimensions so that, d is unchangeable and $\dot{d}$ is for reducing the vector. The algorithm is modified in d mode and additional information may be deleted until it would be possible.

2.1.2. Counting vector (quantization) of d elements

It contains a finite number of integers d.

2.1.3. Calculation coding of one or more d '' code which eventually led to the d be fingerprinted

Fingerprinting algorithms primarily are different, and are used in the reduced dimensions method [2]. Figure 1 shows the organization of fingerprinting algorithms construction method.

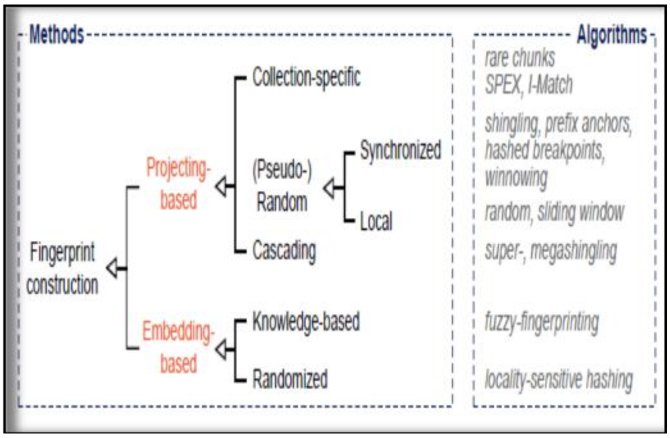


Figure 1: Structure of fingerprints [1].

2.2. Mapping to decrease the dimensions

Fingerprinting for Fd as follows:

$$F\,d = \{h\,(c) \mid c \in cd \text{ and } \sigma\,(c) = \text{true}\}$$

σ denotes the heuristics selection to reduce the dimensions that are correct and the condition is satisfied in it, and it would be realized as a piece of document with special characteristics . In this algorithm, h represents a hash function like Rabins and MD5 algorithm that hash function acts as quantization. Purpose of σ is selecting pieces of document for fingerprints to identify near duplicated documents that are reliable and suitable [1].

2.3. Embedded dimension reduction

The fingerprinting is based on embedding Fd which normally made to a document called similarity hashing that unlike function hash is standard. Its goal is to minimize the number of hash collisions.

The hash function is hφ: D → U, U ⊂ N and also φ (d, dq) ≥ 1-ε if (d, dq) ∈ D. In order to make fingerprinting Fd in d document, a limited number of k models that have used the function h , must form: Fd = {hφ ^ ((i)) (d) | i ∈ {1, ..., k}}. In Figure 1 you can see two types of similarity hash functions that based on the random techniques calculations are done. Similarity hash functions are computed by using hash code. Construction of fingerprinting depends on the length of the document d that must be analyzed at least once, which have the same complexity in all methods, but with each number of fingerprinting, a query FD document is achieved for document. Therefore, the execution time of retrieving fingerprints is depends on K and its construction method has two steps: 1. the method that fingerprinting according to the size of the document raises the length of document. 2. The method that k is independent of | d | [1].

III. HEURISTIC ALGORITHMS FOR DETECTING NEAR- DUPLICATE

Today, about 30% of the documents in the web are repeated. According to the same document are more similarities to each other but just in content is the same content. As is clear, duplication and near- duplication of web pages causes problems for search engines, which makes the Users unsatisfied. This requires the creation of efficient algorithms for computing repeated clusters which the goal is to find duplicate clusters. For this purpose, two syntactic and lexical approaches are applied. In syntactic approach only the syntactic structure of documents are examined. This means that the meaning of words contained in the documents is not reviewed, and just the existing of the same words without attention to the meaning.
of them is sufficient for announcing duplication in documents. One of the reasons for increasing documents and similar availability in the Web is easily access to data in web and the lack of semantic method in detection the same availabilities. Also, the decision about the reliability of these documents, when a different version of it with the same content is available, it will be difficult. Every key word in a document and a query user may have different meanings. Therefore, only apparent similarity measurement of documents cannot give the best results to the user query. Most of the current approaches are based on the words semantic properties, such as the relationship between them. Therefore, needed to use semantic in purpose of meaningful comparison in identifying similar documents is feeling [2,13]. Algorithms that are adaptive with intelligent technology, and use heuristic approaches (heuristics) are shown in Figure 2. Among these algorithms:

3.1. SPEX

The basic idea of SPEX operations is that we can show uniquely each sub-segment from a certain piece, and then the whole piece is unique [3].

3.2. I-Match

This algorithm calculates inverse document frequency weighting in order to extract words. Algorithm I-Match, proposes an algorithm based on (multiple random lexicon) that even to improve the reminder a single sign may also be used [3].

3.3. How to Shingle

Shingle way out is one of the old syntactic approaches, that in order to compute the similarity between two documents was proposed by “Broder“. He pointed to this fact none of the criteria for measuring the distance between the strings measurements are discussed, for example, Hamming distance and edit distance cannot consider duplication so well. In Shingle method each document is broken to overlap pieces that are called «Shingle». In this method Shingles don not rely on any linguistic knowledge except converting document into a list of words. Every text in this way is considered as a sequence of symbols. Every symbol can be a character, word, sentence or paragraph of text. There are no restrictions on the choice of symbol or text units, except that the signs or symptoms of text must be countable. Before any operation, it is assumed that each document using a parser program is converted to a canonical form.

The canonical form is the form in which additional information that may not be of interest, such as punctuation, mark and tags of HTML are removed and then the text will consist of a sequence of symbols.
After preprocessing the document, a list of all Shingles and documents that have appeared in it, is prepared. At this stage, documents are converted into binary that the first value is Shingle itself and the second value is document identifier in which it appears. Shingle size selection usually depends on the desired language and is proportionate to the average length of words in that language. The next stage generates a list of all documents that have common Shingle; the number of shared Shingles is also recorded in this stage. In fact the input to this phase is an ordered list < amount of Shingle and document identifier > and the output is a ternary form < the number of common Shingle and the first document identifier and the second document identifier >.
Then the ternary based on the first document identifier are merged and sorted, at last the total number of common Shingles of two documents regarding the measure similarity of the proposed relationships are reviewed. If the similarity is greater than the threshold, two documents are considered as the equivalent and finds it almost duplicated.
Relationships can help to reduce the allocated space has been sampled from a set of Shingle. Also, we can attribute a length1 specific identifier to every Shingle. This mapping is usually done by Rabin fingerprints .The choice of the length 1 is very important because considering the length less than 1 increases the risk of collision and greater than 1 increases the storage space [4, 5].

3.4. Shingle Method with large size

Shingle method with large size, is re-Shingling of Shingles. In fact, in this method at first, the Shingle method outlined in the previous section on applies to any document. Then abstract obtained on a Shingle is sorting and re- arranged method applies on sorting Shingles. In other words, in this way every text is displayed by Shingles with large sizes. In this way, having just one Shingle with large size common is almost sufficient for announcing two documents duplicated. Have a Shingle with equal large size in the two documents is defined as having a common Shingle sequence [6].

In this method in contrast to conventional Shingle methods, there is no need to be collected and counted common Shingles. So, this method for comparing similarities to normal Shingle is simpler and more efficient. But the problem is that this method does not work well for small documents, because it is difficult to prepare Shingle large size of a small document and doesn't have the usual accuracy Shingle. In addition, this method is not able to identify the scope [4]. However, this method before re- Shingle, arranges existing Shingle. Indeed, this is a sampling of available Shingle. It should be noted that both methods have high positive error [6].

3.5. Hashed breakpoints

According to that hashed value is significant breakpoints for searching; large collections of documents can be used. Each word in its hash value (for example, the total number of ASCII in words) is divided into n parameters [7].

3.6. Winnowing

This algorithm is for selecting strings hash fingerprint for k gramme. Winnowing selects a document that has little similarity with other documents. If the sub-strings have similarity, a threshold guarantee for similar items is considered. Winnowing procedure at each step selects the minimum hash value [8].

3.7. Random (random design)

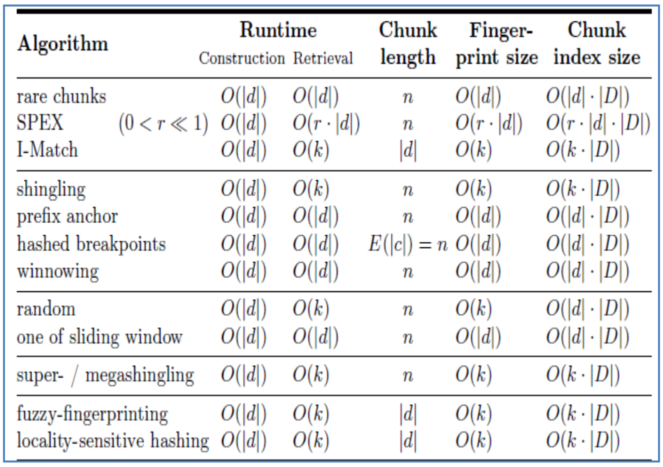
This algorithm is used the cosine similarity relationship between the array of words. This algorithm produces an array of binary with m bits to represent documents. The way it works is that each unique word in the target document is written to a random m -dimensional array, where each element randomly is chosen from [-1.1]. Then all of the generated random arrays of words in the document of the previous stage are added together. The m -dimensional array is produced from results that are accumulated before. Now, with each array element if its value is positive, one element, and if the value is negative, zero element is located instead. Random sampling needs Ω (n) to provide exact estimates. When the size of the data set increases, the accuracy of the random projection method algorithm greatly reduced. This is because, random sampling of pairs showing almost a zero proliferative. It is expected because when the size of the data set will increase, the size of the random sample does not alter and the probability that pairs of proliferative appear in random samples is reduced. While dataset is large and sample size is small, it is very likely that almost any pair of samples is not protected. These conditions, even when the data set contains less pairs that are almost duplicated, will be worsening [9]. This is the point where this method in comparison with the Shingle often has less positive error rate. Ignoring other words, the number of occurrences of words and their weights is the problem of this approach [10].

3.8. The independent permutation procedure (min-wise)

As noted above, calculating the similarity between two documents which the size of Shingle set is large is difficult. Min-wise independent permutation algorithm to solve this problem was proposed by Broder. In this algorithm, the degree of similarity between the hashed values is calculated with Jaccard relationship. The procedure of this algorithm is that any set of Shingle is mapped into m -dimensional array that m is much smaller than the actual number of symbols in a document. In this process, m different hashed function with the names of h1, hm, are produced and will be applied to all Shingle. If the final document displays with S (n) = {s_1, s_2, s_n}, J_{th} element of this set can show the lowest level hash of pervious stage [11].

3.9. Locality Sensitive Hashing Algorithm (LSH)

In method LSH some hash function is used to determine similar documents. The procedure is that first hash functions are classified to k Triad bands in other words, each band consists of k hash functions. All hash functions applied to the input document and the result is stored in each respective band. The hash function is specified for each band and for each pair of documents which are identical to each other. In LSH hash functions in order to combine the results of conditions two approaches are introduced which are known as AND-OR and OR-AND. The method of hash functions into an AND-OR bond with each other AND the results are the output of the AND for each band are the documentation of those couples that like all the bands are known as hash functions . Then the results of different bands are OR together. As a result, the output is each pair that at least by a band of almost duplicated is known [12]. Since the similarity search is an important research issue which is different in application programs for example, media companies such as broadcasters and newspapers are constantly your pictures and videos uploaded to a repository of multimedia the issue of copyright is one of its main concerns. If the near-duplicated versions are retrieved and reported to the users, the pirated versions are diagnosed quickly. So if new documents are illegal the user must terminate the process of uploaded. Although much research has been done in the context of similarity search, it is still a challenge and to accelerate the search process, in view of N-dimensional similarity space is needed. However, plenty of LSH from space to achieve fast query response is needed. Sim pair LSH is a new approach to speed up the basic LSH method that uses the same parts. Sim pair LSH is better way than LSH because it requires less memory cost [14].



Algorithm		Runtime		Chunk	Finger-	Chunk
		Construction	Retrieval	length	print size	index size
rare chunks		$O(\|d\|)$	$O(\|d\|)$	n	$O(\|d\|)$	$O(\|d\| \cdot \|D\|)$
SPEX	$(0 < r \ll 1)$	$O(\|d\|)$	$O(r \cdot \|d\|)$	n	$O(r \cdot \|d\|)$	$O(r \cdot \|d\| \cdot \|D\|)$
I-Match		$O(\|d\|)$	$O(k)$	$\|d\|$	$O(k)$	$O(k \cdot \|D\|)$
shingling		$O(\|d\|)$	$O(k)$	n	$O(k)$	$O(k \cdot \|D\|)$
prefix anchor		$O(\|d\|)$	$O(\|d\|)$	n	$O(\|d\|)$	$O(\|d\| \cdot \|D\|)$
hashed breakpoints		$O(\|d\|)$	$O(\|d\|)$	$E(\|c\|) = n$	$O(\|d\|)$	$O(\|d\| \cdot \|D\|)$
winnowing		$O(\|d\|)$	$O(\|d\|)$	n	$O(\|d\|)$	$O(\|d\| \cdot \|D\|)$
random		$O(\|d\|)$	$O(k)$	n	$O(k)$	$O(\|d\| \cdot \|D\|)$
one of sliding window		$O(\|d\|)$	$O(\|d\|)$	n	$O(\|d\|)$	$O(\|d\| \cdot \|D\|)$
super- / megashingling		$O(\|d\|)$	$O(k)$	n	$O(k)$	$O(k \cdot \|D\|)$
fuzzy-fingerprinting		$O(\|d\|)$	$O(k)$	$\|d\|$	$O(k)$	$O(k \cdot \|D\|)$
locality-sensitive hashing		$O(\|d\|)$	$O(k)$	$\|d\|$	$O(k)$	$O(k \cdot \|D\|)$

Figure 2: (complexity of NDD algorithms) [1].

IV. ASSESSMENT

Wikipedia by evaluating sets concludes when evaluating of near-duplicate detection methods face the problem for choosing, in standard companies such as TREC or Reuters resemblance is reduced exponentially. It means, documents from the high percentage of very low at similar low percentage of intervals with high similarity are changed. As Figure 3 that shows this feature is shown in Reuters. Figure 3 shows the great size of Wikipedia. The similarity distribution of Reuters and Wikipedia are in conflict with each other.

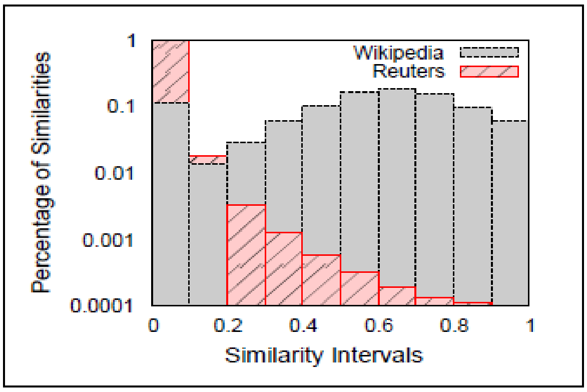


Figure 3: Diagram of the evaluation Wikipedia and Reuters [1].

Figure 4 shows too much similarity universality for fuzzy fingerprint this technique can be used to authenticate and identify that is sensitive to the location of hash and has a significant amount of breakpoint [1].

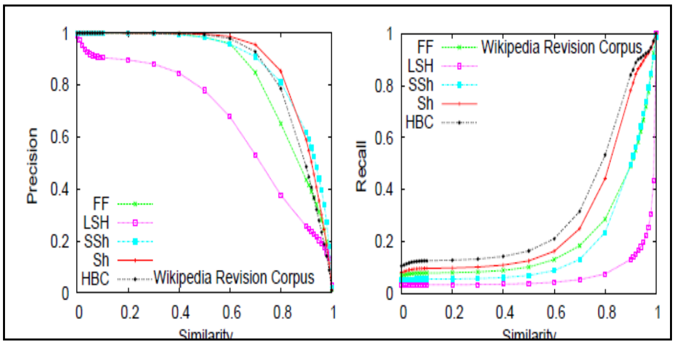


Figure 4: Similarity between Documents [1].

Fingerprinting algorithms are algorithms that are used to detect near-duplication. Wikipedia articles that are regularly listed at the beginning of the address, analyze fingerprinting algorithms of 7 million pairs of documents. And strategies like the first version of each article as a document puts dq search, and compared with other versions it is the first paper that to be replaced. Also near-duplication improves the reliability and accuracy of the assessment (Fig. 4). The results show that pairs of documents that are similar, put into a certain threshold and each set is identified according to the near - duplication. And finally comparison is done by the fingerprint algorithm then the value of integrity using standard methods is computed. The aim is to reduce the ambiguity of user interface. Of course by near-duplication grouping and clustering can hide contents into the cluster. Near-duplication detection is usually done by the search engines on the Web to confirm the approximate fingerprinting, a set of fingerprints that is generated for each document and is used to detect similar documents [1].

The idea of using fingerprints is for identifying duplicates like cryptographic hash functions that are secure against harmful attacks .These functions serve as a high-quality fingerprinting functions. Cryptographic hash algorithms, including MD5 and SHA1 are widely used in the file system. These algorithms are used for data accuracy, and any changes are identified, also approximate fingerprinting detects similar files in a large file system. Fingerprinting normally is used to avoid comparison and transfer great data. In order to realize that remote file browser or proxy server has been modified or not, by fingerprinting and comparison with the previous version, this goal is reached. In other words, virtual fingerprinting is unable to identify a file.

V. FUTURE WORK

Hash method is a new design for almost the same documents. According to data mining techniques that today are used, times of query is significantly improved, using a combination of heuristic methods and techniques including fingerprinting data encryption technique mining the number of comparisons can be minimized.

VI. RESULTS

Fingerprinting and Secrecy algorithms in the relational data base are used in order to protect documents against copyright, and because the reliable identify to near-duplicate documents must be existed so, run time is high. Also near-duplicated of the web pages on the search engines have a combination of problems including user dissatisfaction. For this reason, heuristic algorithms such as: SPEX, Shingle off, Shingle large size, hashed breakpoints, Winnowing, Random, min-wise are used. By comparison that is done between Wikipedia and Reuters; the similarity between documents exponentially is decreased. It was shown that the distribution of similarities Reuters and Wikipedia are in conflict with each other. Cryptographic hash algorithms for data verification are used to identify any changes. Also approximate fingerprinting detects similar files in a large file system. Thus, the results show that a set of documents that are similar pairs are placed at a certain threshold, and each set according to near-duplication is identified. And finally comparison is done by the fingerprint algorithm then the value is calculated using the standard method.

REFERENCES

[1] Potthast, M., and Stein, B.: ‘New Issues in Near-duplicate Detection’, 2008, pp. 601-609

[2] Ignatov, D., J´anosi-Rancz, K., and Kuznetsov, S.: ‘Towards a framework for near-duplicate detection in document collections’, 2009, pp. 215–233

[3] Pamulaparty, L., Rao, D.M.S., and Rao, D.C.V.G.: ‘A Survey on Near Duplicate Web Pages for Web Crawling’, 2013, 2 (9), pp. 1-6

[4] Broder, A.Z.: ‘On the resemblance and containment of documents’, 1997, pp. 1-9

[5] Giancarlo, R., and Sanko, D.: ‘Identifying and Filtering Near-Duplicate Documents’, 2000, pp. 1-10

[6] Chowdhury, A.K.A.: ‘Lexicon randomization for near-duplicate detection with I-Match’, 2008, pp. 255–276

[7] Finkel, R.A., Zaslavsky, A., Monostori, K., and Schmidt, H.: ‘Signature extraction for overlap detection in documents’, 2001, 4, pp. 1-6

[8] Schleimer, S., Wilkerson, D.S., and Aiken, A.: ‘Winnowing: Local Algorithms for Document Fingerprinting’, 2003, pp. 76-85

[9] Deng, F., and Rafiei, D.: ‘Estimating the Number of NearDuplicate Document Pairs for Massive Data Sets using Small Space’, 2007, pp. 1-10

[10] Fan, J., and Huang, T.: ‘A fusion of algorithms in near duplicate document detection’, 2012, pp. 234-242

[11] Broder, A.Z., Charikar, M., Frieze, A.M., and Mitzenmacher, M.: ‘Min-Wise Independent Permutations’, 1998, pp. 1-36

[12] Gionis, A., Indyk, P., and Motwani, R.: ‘Similarity Search in High Dimensions via Hashing’, 1999, pp. 1-12

[13] Alsulami, B.S., Abulkhair, M.F., and Eassa, F.E.: ‘Near Duplicate Document Detection Survey’, Computer Science & Communication Networks, 2012, 2(2), pp. 147-151

[14] Fisichella, M., Deng, F., and Nejdl, W.: ‘Efficient Incremental Near Duplicate Detection Based on Locality Sensitive Hashing’, 2010, pp. 152-166