A fusion of algorithms in near duplicate document detection[1]

Jun Fan[1, 2], Tiejun Huang[1, 2]

[1] National Engineering Laboratory for Video Technology, School of EE & CS, Peking University, Beijing 100871, China

[2] Peking University Shenzhen Graduate School, Shenzhen 518055, China

jfan@jdl.ac.cn， tjhuang@jdl.ac.cn

Abstract. With the rapid development of the World Wide Web, there are a huge number of fully or fragmentally duplicated pages in the Internet. Return of these near duplicated results to the users greatly affects user experiences. In the process of deploying digital libraries, the protection of intellectual property and removal of duplicate contents needs to be considered. This paper fuses some "state of the art" algorithms to reach a better performance. We first introduce the three major algorithms (shingling, I-match, simhash) in duplicate document detection and their developments in the following days. We take sequences of words (shingles) as the feature of simhash algorithm. We then import the random lexicons based multi fingerprints generation method into shingling base simhash algorithm and named it shingling based multi fingerprints simhash algorithm. We did some preliminary experiments on the synthetic dataset based on the "China-US Million Book Digital Library Project"[2]. The experiment result proves the efficiency of these algorithms.

Keywords: duplicate document detection, digital library, web pages, near duplicate document

1 Introduction

Duplicate and near duplicate documents detection plays an important role in both intellectual property protection and information retrieval. The definition of duplicate

[1] The work is partially supported by National Natural Science Foundation of China under grant No. 90820003, the Important Scientific and Technological Engineering Projects of GAPP of China under grant No. GAPP-ZDKJ-BQ/15-6, and CADAL project.

[2] See http://www.ulib.org for more details.

is unclear. The general notion is that files with minor edits of each other are also considered as duplicates.

The digital libraries provide users with on-line access to digitized news articles, book, and other information. This environment greatly simplifies the task of illegally retransmit or plagiarize the works of others which violates their copyrights.

In recent times, the dramatic development of the World Wide Web has led a proliferation of documents that are identical or almost identical. These copies of documents are same or only differ from each other in a very small portion. The appearances of duplicate and near duplicate documents in the search results annoy the users.

Brin et al. developed COPS [1] in the course of deploying a digital library system. COPS is a prototype of a document copy detection mechanism and dependents on sentence overlap. The registration based architecture of this prototype is widely used from then on. Shivakumar et al. [2] [3] proposed SCAM, which is based on comparing the word frequency occurrences of documents.

Andrei Broder et al.'s [7] [8] shingling algorithm and Charikar's [8] random projection based approach are considered "state of the art" algorithms for detecting near duplicate web documents. Henzinger [12] compared these two algorithms in a set of 1.6B distinct web pages and proposed a combined algorithm. The combined algorithm got a better precision compared to using the constituent algorithms individually. Another well known duplicate document detection algorithm called I-Match was proposed by Chowdhury et al. [4] and it was evaluated on multiple data collections. Kolcz et al. [5] studied the problem of enhancing the stability of I-Match algorithm with respect to small modifications on document content. They presented a general technique which makes use of multiple lexicons randomization to improve robustness.

This paper reports two attempts to improve the performance of simhash algorithm. In section 2, we described the three major algorithms (shingling [7], I-match [4], simhash [13]) in duplicate document detection and their developments in the following days. In section 3, we introduced our improvement attempts: take

sequences of words (shingles) as the feature and fuse the random lexicons based multi fingerprints generation method with simhash. In section 4, we presented the experiments results in the "China-US Million Book Digital Library Project" dataset. Finally, Section 5 brings this paper to a conclusion.

2 Major Algorithms in Duplicate Document Detection

2.1 Shingling, Super shingling, Mini-wise Independent Permutation Algorithms

Broder et al. [7] defined two concepts: resemblance and containment, to measure the similarity degree of two documents. Documents are represented by a set of shingles (or k-grams). The overlaps of shingle sets were calculated.

As there are too many shingles in a document, Broder et al. [7] [8] developed some sampling methods. Super shingling [7] and mini-wise independent permutation [8] are two kinds of the sampling methods. Super shingling method is shingling the shingles. The document is then represented by its super shingles. Mini-wise independent permutation algorithm provide an elegant construction of a locality sensitive hashing schema for a collection of subsets with the set similarity measure of Jaccard Coefficient. [8]

2.2 I-Match, Multiple Random Lexicons based I-Match Algorithms

Chowdhury et al. [4] extract a subset of terms from a document according to their NIDF (the normalized inverse document frequency) [4] values. They hashed the terms orderly and claimed that, if the terms are carefully chosen, near-duplicate documents are likely to have the same hash, whereas it is extremely unlikely that two dissimilar documents will hash to the same value. But the gathering of global collection statistic (NIDF) presents a significant challenge.

As the I-Match algorithm is based on the precondition of filtering out all the different words in near duplicate documents, its recall is relatively low. Kolcz et al. [5] proposed the multiple random lexicons based I-Match algorithm, which utilizes additional randomly created lexicons to generate multiple fingerprints. They claimed that, this method is also applicable to other single-signature schemes to improve recall. In our experiments about this algorithm, we set the parameters the same with theirs. This was also discussed in [5] [6].

2.3 Random Projection, Simhash Algorithms

Charikar [8] developed a locality sensitive hashing schema for a collection of vectors with the cosine similarity measure between two vectors, which is based on random projection of words. Henzinger [12] implemented this schema into the application of duplicate web page detection, and called it random projection.

Manku et al. [13] added the concept of feature weight to random projection, and named it simhash algorithm. Given feature vectors and corresponding feature weight, it generates a simhash fingerprint. The hamming distance of two vectors' simhash fingerprints is proportional to the cosine similarity of the two vectors. When the hamming distance of two simhash fingerprints is smaller than a threshold, the two documents of the two fingerprints are considered as duplicate. In our experiments, we set the size of the simhash fingerprint to 32 bits. And we examined the performance of each simhash based algorithm with a broad range of threshold from 0 to 31 bits.

3 Model Enhancements

In this paper, we did certain degree of fusion based on the characters of each class of algorithm mentioned above.

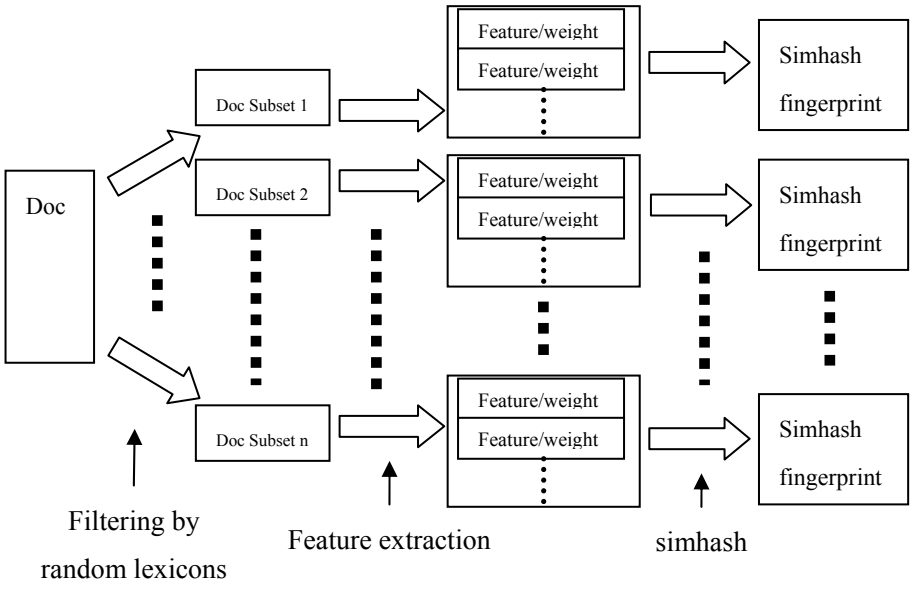

Fig. 1. Framework of the fused algorithm

3.1 Shingling based Simhash Algorithm

Henzinger [12] did a comparison between random projection and the shingling algorithm. At the end of that paper, the author proposed a combined algorithm which is in fact sequentially running random projection algorithm after the shingling algorithm to get a better performance. The author also proposed to study the performance of implement random projection algorithm on sequence of tokens, i.e., shingles, instead of individual tokens. Manku et al. [13] also mentioned to study how sensitive the simhash is to changes in features and weights of features as the future work.

Considering the simhash algorithm is independent of feature selection and assignment of weights to features. Intuitively, sequences of words (shingles) are more representative than individual words for a document. We use the k-shingles (word sequences of length k) as the features of the simhash algorithm, and the sum of IDF value of words in a k-shingle as the weight of the corresponding feature. This is

shown in feature extraction part in Fig. 1. We named this as shingling based simhash algorithm, and regarded it as a fusion of shingling and simhash algorithms. It is different from simply running the two algorithms sequentially in [12].

3.2 Multiple Random Lexicons based Simhash Algorithm

As mentioned by Kolcz et al. [5], randomly creating extra lexicons to generate additional fingerprints is applicable to other single-signature algorithm. We introduce this method into simhash algorithm. We filter documents by randomly created lexicons and generate multi simhash fingerprints as shown in Fig. 1. If the hamming distance between two fingerprints of two documents generated by the same extra random lexicon is smaller than the threshold, the two documents are reckoned as duplicate. We named this as multiple random lexicons based simhash algorithm.

We then fusion the two improvements into an integrated algorithm which is named shingling based multi fingerprints simhash algorithm.

4 Experiments

Although duplicate document detection has been studied for a long time and in a broad area, there isn't any widely accepted experiment dataset. Studies on duplicate document detection use their own datasets [4] [5] [6] [7] [8] [12] [13].

In our experiments, we randomly selected 1403 books (all in English) from the "China-US Million Book Digital Library Project". We then divided these books into 143,798 rough 4KB size texts. We propose these texts are unduplicated. We selected 5 texts randomly and modified (insert, delete, replace word) these 5 texts at random locations. We constructed 600 texts (120 texts for each source text) in this way and considered they are near duplicate documents. We calculated the fingerprints of these 144,403 texts, and used the 5 source texts as the queries. We calculated the precisions, recalls of these 5 queries and counted the macro-averages. P_i, R_i are the precision and recall value corresponding to each query. MacroP, MacroR are the macro-averages of precisions and recalls.

$$\text{MacroP} = \frac{1}{n}\sum_{i=1}^{m} P_i \ . \quad (1)$$

$$\text{MacroR} = \frac{1}{n}\sum_{i=1}^{m} R_i \ . \quad (2)$$

The F-measure is calculated as:

$$\text{F-measure} = 2 * \text{MacroP} * \text{MacroR} / (\text{MacroP} + \text{MacroR}) \quad (3)$$

The experiments results we listed blew are all the macro-averages of the 5 queries. In order to clearly distinguish different curves, results of some parameter values aren't listed in the following figures.

Before the implementation of various copy detection algorithms, each document is first passed through a stopword-removal and stemming process, which removes all the stopwords and reduces every word to its stem.

4.1 Shingling based Simhash Algorithm

As shown in Fig. 2, and the algorithm gets the best performance when shingle size equals 2 (words sequence of size 2). Shingle size of 1 is in fact the original simhash algorithm. The best F-measure value was improved from 0.6117 to 0.7469 as the shingle size grows from 1 to 2. In shingling based simhash algorithm with shingle size k, if we modified n words in random locations, the affected features range from n to n*k. With the increase of shingle size k, the affected features increase multiplied and the performance decreases. In the other side, if we only select single words as features, there maybe two document with roughly the same words, in different sequences and with different meanings are considered as duplicate. With larger shingle size k, we can reduce this kind of false positive obviously. By taking the k-shingles as the features, the effect of word order was considered. Therefore, there is a tradeoff of shingle size k to keep the balance.

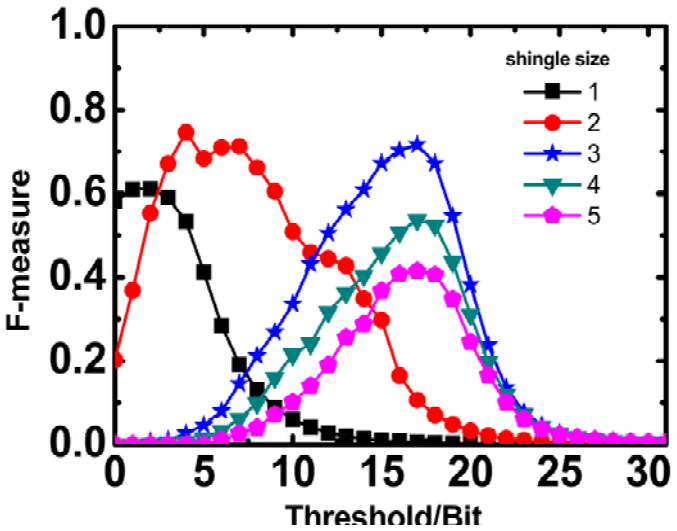

Fig. 2. The F-measure with shingle size range from 1 to 5, and the threshold of hamming distance range from 0 to 31.

4.2 Multiple Random Lexicons based Simhash Algorithm

In this experiment, we set the features of simhash to be shingles with size 2. It is in fact the shingling based simhash algorithm with shingle size 2, when the random lexicon size is set to 1. With the increase of the random lexicon size and the threshold, the recall increases, this was showed in Fig. 3. Chowdhury et al. [4] shown the significant increase in recall of multiple random lexicons method, but didn't illustrate the precision in their paper. In Fig. 4, we can see the precision decreases slightly accordingly. The F-measure was shown in Fig. 5. From the F-measure's view, it doesn't mean that the larger the random lexicon size is, the better the performance will be. There exists a balance between precision and recall on the selection of random lexicon size.

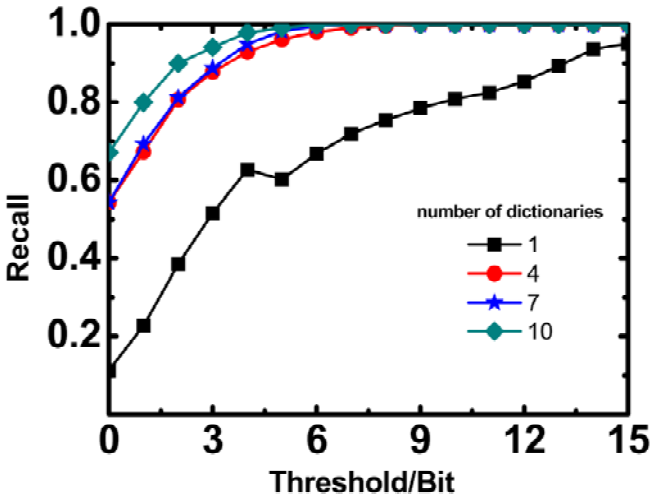

Fig. 3. The recall of the shingling based multi fingerprints simhash algorithm with random lexicon size 1, 4, 7, 10, and the threshold of hamming distance range from 0 to 15.

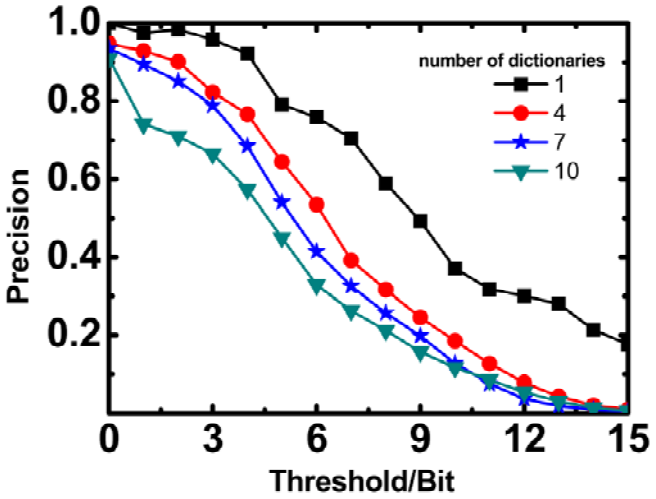

Fig. 4. The precision of the shingling based multi fingerprints simhash algorithm with random lexicon size 1, 4, 7, 10, and the threshold of hamming distance range from 0 to 15.

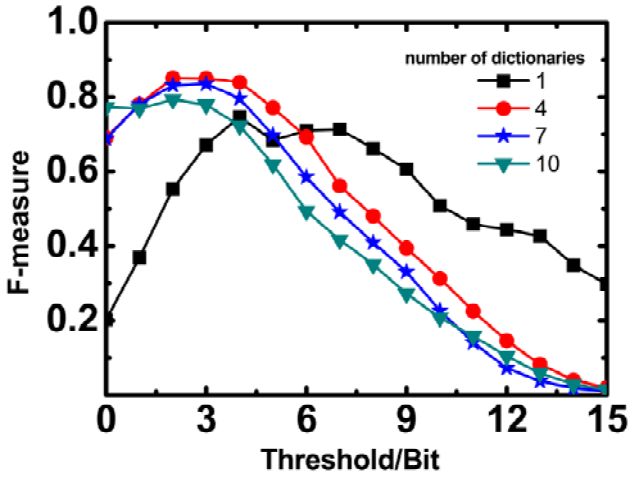

Fig. 5. The F-measure of the shingling based multi fingerprints simhash algorithm with random lexicon size 1, 4, 7, 10, and the threshold of hamming distance range from 0 to 15.

We listed the best F-measure of each random lexicon size with their corresponding thresholds in table 1. We can see that, when random lexicon size equals 5 and threshold is 3, we get the best F-measure value 0.8805 in this experiment environment. There is about an 18% percent improvement compared with shingle based simhash algorithm with shingle size 2, and a 44% improvement compared with the original simhash algorithm.

Table 1. The best F-measure of each random lexicon size with their corresponding thresholds

random lexicon size	Best F-measure score	Threshold
1	0.7469	4
2	0.8071	6
3	0.8213	3
4	0.8515	2
5	0.8805	3
6	0.7855	3
7	0.8356	3
8	0.8077	1
9	0.8139	2
10	0.7942	2

We also tested shingling (we set the parameters the same with D. Fetterly et al. [7]), I-Match and multiple random lexicons based I-Match algorithms, the performances

are roughly the same with most published papers. Especially, the performances of I-Match and multiple random lexicons based I-Match algorithms are similar with Theobald et al. [14] in our experiment environment. In our experiments, the two algorithms got even lower recalls, also with high precisions. Besides the character of the two algorithms themselves, the small size of the experiment dataset that yields a poor collection statistic may be another reason.

5 Conclusions

We described the three major algorithms (shingling, I-match, simhash) in duplicate document detection and their development in the following days. We introduced our idea of fusing these algorithms and presented the experiment results in the "China-US Million Book Digital Library Project" dataset. The performance of shingling based simhash algorithm was affected by test dataset in two sides. There exists a balance in selection of shingle size. Multiple random lexicons based simhash algorithm can improve recall but impair precision slightly. We should seek a balance when choose random lexicon size. As there is no conflict between feature selection and multi fingerprints generation, we implemented the combination which performances much better than the original simhash algorithm in our synthetic dataset. We are now constructing larger test dataset to validate our algorithm, and trying to implement our algorithm on other datasets.

References

1. Brin, S., Davis, J., Garcia-Molina, H. Copy Detection Mechanisms for Digital Documents. In: Proceedings of the ACM SIGMOD Annual Conference. (1995)
2. Shivakumar, N., Garcia-Molina, H. SCAM: A copy detection mechanism for digital documents. In: Proceedings of the 2nd International Conference in Theory and Practice of Digital Libraries (DL'95). (1995)

3. Shivakumar, N., Garcia-Molina, H. Building a scalable and accurate copy detection mechanism. In: Proceedings of the 1st ACM Conference on Digital Libraries (DL'96). (1996)
4. Chowdhury, A., Frieder, o., Grossman, D., Mccabe, M C. Collection statistics for fast duplicate document detection. ACM Transactions on Information Systems, Vol. 20, No. 2. (2002)
5. Kołcz, A., Chowdhury, A., Alspector, J. Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization. In: Proceedings of the tenth ACM SIGKDD, Seattle, WA, USA. (2004)
6. Jack, G., Conrad, Xi S., Guo, Cindy P. Schriber. Online Duplicate Document Detection: Signature Reliability in a Dynamic Retrieval Environment. In: Proceedings of the twelfth international conference on Information and knowledge management. (2003)
7. A.Z. Broder, Glassman SC, Manasse MS. Syntactic clustering of the Web. In: Proceedings of the 6th International Web Conference. (1997)
8. A.Z. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-Wise Independent Permutations. Journal of Computer and System Sciences, pp. 630-659. (2000)
9. Fetterly, D., Manasse, M., Najork, M. On the evolution of clusters of near-duplicate web pages. In: Proceedings of first Latin American Web Congress, pp. 37-45. (2003)
10. Fetterly, D., Manasse, M., Najork, M. Detecting Phrase-level Duplication on the World Wide Web. The 28th ACM SIGIR, pp. 170-177. (2005)
11. Charikar. Similarity estimation techniques from rounding algorithms. In Proceedings of 34th Annual Symposium on Theory of Computing. (2002)
12. Henzinger, M. Finding near-duplicate web pages: a large-scale evaluation of algorithms. in: Proceedings of the 29th ACM SIGIR, pp. 284-291. (2006)
13. Manku, G. S., Jain, A., Sarma, A. D. Detecting near-duplicates for web crawling. In: Proceedings of the 16th international conference on World Wide Web, pp. 141-150. (2007)
14. Martin Theobald, Jonathan Siddharth, Andreas Paepcke. SpotSigs: robust and efficient near duplicate detection in large web collections. In: Proceedings of ACM, SIGIR (2008)