Duplicate text detection based on LCS algorithm

JIANKUN YU[1*], MENGRONG LI[1], DENGYIN ZHANG[2, a]

[1]Nanjing University of Posts and Communications, No.66, Xin Mofan Road, Nanjing, China

[2]Jiangsu Nanyou IoT Science Park Co.Ltd , No.66, Xin Mofan Road, Nanjing, China
[a]zhangdy@njupt.edu.cn

Keywords: near-duplicate detection; duplicate detection; duplicate text filter.

Abstract. Broder's Shingling and MinHash are two of the state-of-the-art approaches in detecting near-duplicate documents. But both of these two methods did not take the relative position of elements into consideration. This paper proposes a method which combines Shingling and LCS algorithm called SWLR (Shingling with Location Relationship). And proposes a pre-filter method to speed up the execution speed of SWLR. Experiment results shows that SWLR performances better than Shingling in both recall and precision rate and better than MinHash in recall rate. By applying pre-filter method, SWLR could even be executed faster than MinHash and Shingling.

Introduction

As estimated about 35% web pages on the Internet are near-duplicate duo to the reproduction of the text content, such as news, quotations and so on. The main differences between these near-duplicate pages may only exists in the unimportant part of pages, such as titles, navigation bars, ads and the copyright notices on the bottom of the page. These near-duplicate web pages not only increase the space burthen of storage, but also degrade the user's query experience when searching on the Internet.

In 1997, Broder proposed Shingling which was based on Jacquard coefficient to measure similarity of different files [1]. And in 2000, Broder revised and extended Shingling by taking repeated occurrences of elements into consideration [2]. Though Shingling has a good performance in eliminating duplicated files, it will take a very long time to compare two large files. Then Broder proposed MinHash in 1997 which aimed to accelerate the comparison speed between large files by reducing the dimension of file's features [3]. MinHash was initially used in the AltaVista search engine to detect duplicate web pages. It has also been applied in large-scale clustering problems. SimHash was proposed by Charikar.M.S in 2002 and it has been widely used by Internet companies including Google [4]. It maps the features of a file to a long bit sequence. We can regard the two files as near-duplicated if the two bit sequences differ in only k bits. Generally, the length of the bit sequence is 64 and the number of different bits is 3.

Shingling and MinHash are based on Jacquard coefficient, which means the location relationships between features are regarded as unimportant. SimHash maps the features to a bit sequence. A single feature will have a great influence on the map result if the feature number is small. According to this characteristic, SimHash only works for detecting duplication among large files.

The longest common subsequences (LCS) is the problem of finding the longest subsequence common to all sequences in a set of sequences. This paper will propose a

near-duplicate detection method based on LCS algorithm which is called SWLR. As the time complexity of SWLR is *O(nlogn)* which means SWLR's execution speed will much slower than Shingling and MinHash. We will also propose a method to accelerate the detection speed of SWLR.

The rest of the paper is organized as follows. Section 2 briefly introduces Shingling and MinHash. Section 3 presents a new near-duplicate detection method called SWLR based on LCS and the way to speed up the detection speed. Experimental results for the comparison among SWLR, Shingling and MinHash will be presented in Section 4, followed by conclusions in Section 5.

Shingling and MinHash

A shingle is nothing but a word q-gram of a document [5]. For example, if a document has n words, a continuous subsequence of q words is a shingle. The document will have n-q+1 shingles.

Assuming S_A and S_B to be the set of all shingles of the documents A and B respectively. The shingling method uses Jacquard coefficient as similarity or resemblance measure of two documents A and B:

$$r(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} \tag{1}$$

We can let π be a uniformly at-random chosen permutation over the set of all permutations of document such as D, then Eq. 2 will be got.

$$\Pr(\min\{\pi(S_A)\} = \min\{\pi(S_B)\} = \frac{|S_A \cap S_B|}{|S_A \cup S_B|} = r(A,B) \tag{2}$$

Eq. 2 has been proved by author Henzinger M [6]. By choosing a set of t independent uniformly random permutations π_1, π_2 ,..., π_t, we can map the set of all shingles of document D to a t dimensional vector. Which represents as following vector.

$$\overline{S} = (\min\{\pi_1(S)\},...,\min\{\pi_t(S)\}) \tag{3}$$

For documents A and B, we can estimate the resemblance by calculating Jacquard coefficient of $\overline{S_A}$ and $\overline{S_B}$. Which will speed up the calculation due to using a smaller sketch.

MinHash also uses Jacquard coefficient to indicate the similarity of two sets. Let h be a hash function that maps the members of A and B to distinct integers, and for any set S define $h_{\min}(S')$ to be the minimal member of S with respect to h- that is, the number x of S' with the minimum value of $h(x)$. If we apply $h_{\min}$ to both A and B, we will get the same value exactly when the element of the union $A \cup B$ with minimum hash value lies in the intersection $A \cap B$. We can get Eq. 4.

$$\Pr(h_{\min}(A) = h_{\min}(B)) = r(A,B) \tag{4}$$

Usually, we choose t different hash functions mapping the document to a t dimensional vector. Then estimating the similarity of document A and B by calculating the Jacquard coefficient of vector A and vector B.

Near-Duplicate detection method based on LCS

Shingling with Location Relationship method. As was mentioned earlier in section 2, Shingling and MinHash are based on Jacquard coefficient. Thus, the comparison between two different sets does not take into account the effect of the relative position of the elements. Now we would like to introduce SWLR algorithm which combine LCS algorithm and Shingling. It will take the relative position of elements into consideration on the base of Shingling.

Assuming S_D is the set of elements of document D. We could use Eq. 5 to get the longest sub-elements of S_A and S_B.

$$\mathrm{SWLR}(S_A, S_B) = \begin{cases} \phi & if \quad i = 0 \quad or \; j = 0 \\ \mathrm{SWLR}(S_{Ai-1}, S_{Bi-1}) \cap S_{Ai} & if \; S_{Ai} = S_{Bi} \\ longest(\mathrm{SWLR}(S_{Ai}, S_{Bj-1}), \mathrm{SWLR}(S_{Ai-1}, S_j)) & if \; S_{Ai} \neq S_{Bi} \end{cases} \tag{5}$$

Then we can get the similarity score of document A and B by Eq. 6.

$$r(A, B) = \frac{|\, SWLR(S_A, S_B)\,|}{avg(|\, S_A\,| + |\, S_B\,|)} \tag{6}$$

SWLR with pre-filter method. The complexity of SWLR algorithm is *O(mn)*. Obviously it's not a good idea to use SWLR algorithm in detecting duplicate text among large documents. To minimize the time SWLR algorithm costs, we proposed a quick filter method based on bit comparison. Before detecting document A and B, we map the element sets of each document to a bit sequence with following steps:

1. Generate a bit sequence with length of 64 or 128. Initialize all the bits to 0.
2. Map the element into 64 or 128 with hash algorithm. Set the bit value under the corresponding index is 1.

When comparing document A and B. We first compare the hash value of their elements. If we regard document A and B are similarity in the case of $r(A, B) > q$, we can see that the bit difference between A and B is no more than $avg(Len(A) + Len(B))$. We can compare the hash value of document A and B first to filter out the majority of non similarity documents. Then use SWLR algorithm to judge weather document A and B are similar.

Experimental results

Dataset. Our experiments were performed on a collection of 864 restaurant records from the Fodor's and Zagat's restaurant guides that contains 112 duplicates [8].

Evaluation method. We will use recall and precision rate to present the performance of each method and we will compare the speed of each method on the same dataset. The definition of recall and precision rate are defined as Eq. 7:

$$recall = \frac{|D \cap \overline{D}|}{|D|} \qquad precision = \frac{|D \cap \overline{D}|}{|\overline{D}|} \tag{7}$$

D means all the duplicated elements in the dataset. $\overline{D}$ represents the elements which detected by the methods.

Except recall and precision rate, we will also compare the speed each method executes.

Results. We have carried out several experiments for each method on the dataset and picked the best result of each method according Eq. 8. The recall and precision rate are shown in Fig. 1.

$$score = 2 * recall * precision / (recall + precision) \tag{8}$$

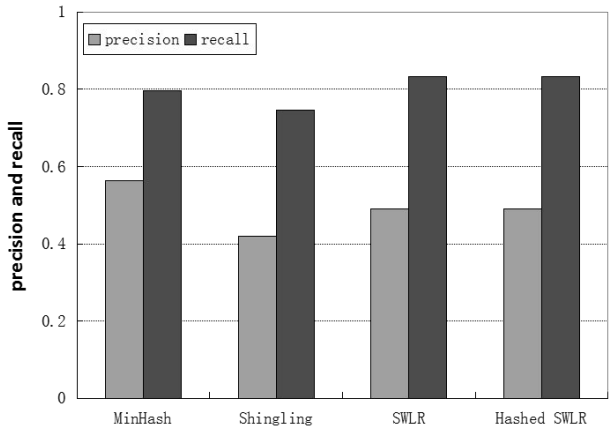

Fig. 1. Recall and precision rate comparison

As shown in Fig. 1. SWLR owns the best performance in recall rate. In both recall and precision rate, SWLR is better than Shingling. And the added pre-filter method do not take any negative impact to SWLR.

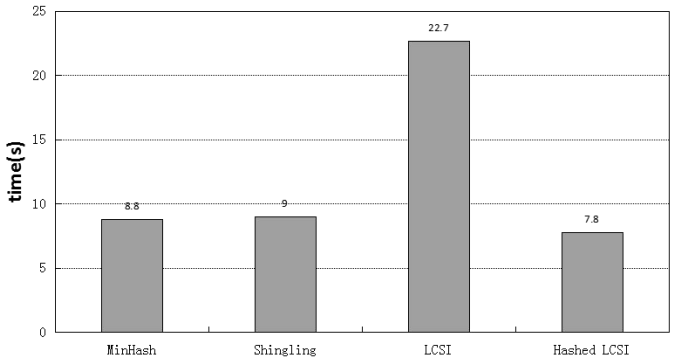

Fig. 2. Time cost comparison among different methods

From Fig. 2. We can conclude that when applying hash value to SWLR, the performance of SWLR method can be greatly improved. And even better than MinHash and Shingling.

Conclusions

A method was presented for detecting near-duplication text on the base of Shingling algorithm. Since the complexity of SWLR is too high to detecting on large dataset. We introduced a method based on hash indexing to exclude the non-similarity texts. Compared with MinHash and Shingling, SWLR takes the relative position of elements into consideration. Results show that SWLR performances better than Shingling and has a higher recall rate than MinHash. By applying the hash index method, SWLR could be executed 3 times faster than before and even faster than MinHash and Shingling. This paper offers an efficient solution to detect near-duplicate texts on large datasets.

Acknowledgement

The work was partially supported by Swedish Research Links [No.348-2008-6212], the National Natural Science Foundation of China [61571241], Industry-university-research Prospective joint project of Jiangsu Province [BY2014014] , and Major projects of Jiangsu Province university natural science research [15KJA510002].

References

[1] Broder A Z, Glassman S C, Manasse M S, et al. Syntactic clustering of the Web[J]. Computer Networks & Isdn Systems, 1997, 29(813):1157–1166.

[2] Broder A Z. Identifying and Filtering Near-Duplicate Documents.[J]. Lecture Notes in Computer Science, 2000:1-10.

[3] Broder A Z, Charikar M, Frieze A M, et al. Min-Wise Independent Permutations.[J]. Journal of Computer & System Sciences, 2000, 60(3):630-659.

[4] Charikar M S. Similarity estimation techniques from rounding algorithms[J]. Proc of Stoc, 2002:380-388.

[5] Gharghe Z E, Bidgoli B M. Weighted shingling: an adaptation of shingling for weighted shingles[C]// Innovations in Information Technology, 2009. IIT '09. International Conference on. IEEE, 2009:150-154.

[6] Henzinger M. Finding near-duplicate web pages: A large-scale evaluation of algorithms[C]// International Acm Sigir Conference on Research & Development in Information Retrieval. ACM Press, 2006:284 - 291.

[7] Chen L, Guo-Shi W U, Jing L I. Duplicate Detection for Chinese Texts Based on Semantic Fingerprint and LCS[J]. Computer Engineering & Software, 2014.

[8] http://www.cs.utexas.edu/users/ml/riddle/data.html