ISSN No. 0976-5697

Volume 6, No. 5, May - June 2015

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Data Duplication and Near Data Duplication Methods: A Review

Radha Saini
M. Tech Scholar
DCTM College
Palwal (India)
sainiradha2010s@gmail.com

Dr. Seema Phogat
Assistant Professor (Computer Science)
Pt. J. l. N. Govt P. G College
Faridabad (India)
seema.phogat@gmail.com

Abstract: Search engines are the major component on the web for retrieving the information. But List of retrieved documents contains a high degree of duplicated and near document result. So in this case, there is a need to improve the performance of search results. For elimination of data duplicate and near duplicate document detection, search engine use data filtering algorithm to save user's effort and time. In this paper, a detailed comparison of existing methods is presented.

Keywords: duplicate document, near duplicate pages, detection approaches

INTRODUCTION

Information on the Web is very large in size. There is a need to efficiently satisfying the user's need. Search engines become the major component on the web for retrieving the information. Where, among users looking for information on the Web, 85% submit information requests to various Internet search engines. Search engines are critically important to help users find relevant information on the Web
Whenever Search engines gives the list of documents ranked according to best match to the user's request. These documents are presented to the user for examination and evaluation.
The efficient identification of near duplicates is an important issue especially when it has a large amount of data and the necessary to save data from different sources and needs to be addressed. Though near duplicate documents contain similarities, they are not bit wise similar [1].

NEAR DUPLICATE DOCUMENT

Two documents are regarded as duplicates if they have contained identical document content. Documents that find small dissimilarities and are not identified as being "exact duplicates" of each other but are identical to a remarkable extent are known as near duplicates [2].

NEAR DUPLICATE DOCUMENT DETECTION APPROACHES:

In general these approaches may be broadly classified into Syntactic, URL based and Semantic based approaches [2].
Web contains duplicate pages and gapered web pages in affluence. Standard check computing techniques can facilitate the easy apprehending of documents that are duplicates of each other.

Following are some of the examples of near duplicate documents [2] :

- Files with a few different words – outspread form of near-duplicates
- Files with the same content but different formatting – like bold, italics, and underline etc.
- Files with the same content but different file type – for instance, Microsoft Word and PDF versions of the same file.

PROBLEMS DUE TO NEAR DUPLICATE DATA:

- **INDEXING:**
  These pages enlarge the space required to stored the index that either decelerates or amplifies the cost of serving result.

- **CRAWLING:**
  Due to presence of duplicate data crawler waits its time to fetching pages which are not relevant to user.

- **CACHING:**
  Due to absence of knowledge about the replicated of stored space is not properly utilized.

NEAR DUPLICATE DOCUMENT DETECTION:

Detection of Near Duplicate Document (NDD) is the problem of detecting all documents promptly whose similarities are equal to or greater than a given threshold. Near Duplicate document detection became an interesting problem in late 1990s with the cultivation of Internet [2].
Most existing techniques for identifying near duplicates are divided into two categories

- Near duplicate prevention.
- Near duplicate detection.

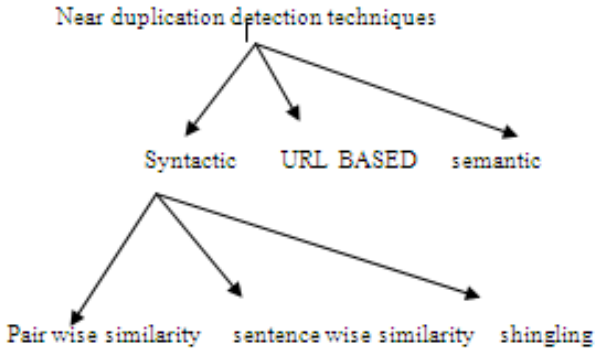

Fig 1: Syntactical Approaches

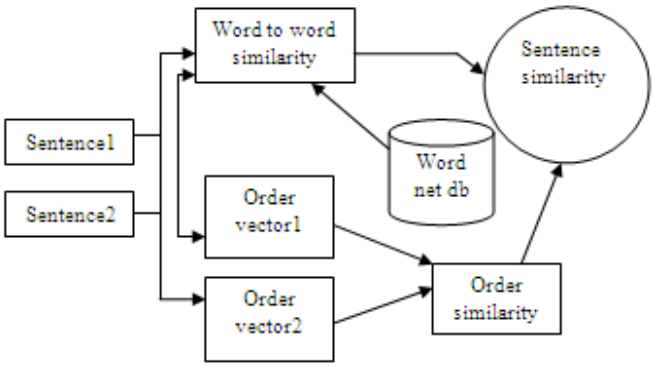

Fig 2: Sentence similarity method

PAIR WISE SIMILARITY

Pair-wise similarity method deals with finding pairs of objects in a large dataset that are similar according to some measure.

$$Sim(di, dj) = \sum wt,di \cdot wt,dj \quad t \in v$$

where
sim(di, dj) is the similarity between documents di and dj
V is the vocabulary set. In this type of similarity measure, a term will contribute to the similarity between two documents only if it has non-zero weights in both. Therefore, t 2 V can be replaced with t 2 di \ dj .

Algorithm 1 Compute Pairwise Similarity Matrix

1. v i, j : sim[i, j]
2. for all t € V do
3. $P_t \leftarrow$ postings(t)
4. for all di, dj 2 pt do
5. sim[i, j] ← (sim[i, j] + wt,di · wt,dj

SENTENCE WISE SIMILARITY

Sentences-wise similarity measure similarity comparing exterior tokens of inter-sentences, but relevance measure can be obtained only by comparing the interior meaning of the sentences.

Steps for computing semantic similarity between two sentences:

- First each sentence is partitioned into a list of tokens.
- Part-of-speech disembarkation.
- Emanate words.
- Find the most appropriate sense for every word in a sentence
- Finally, compute the similarity of the sentences based on the similarity of the pairs of words

- **SHINGLING**

Shingles are sequences of tokens of length w that appear in a document and the similarity of two documents can be calculated based on the number of shingles [3]. Since it is computationally impervious to calculate the similarity of the sets of all of the shingles for every document, a method based on the sketch of a document is used instead. To calculate the sketch of a document, each shingle in a document is hashed using h hash functions and a list is maintained of the minimum hash values found for each hash function. The sketch of a document is then its set of h minimum hash values and the similarity of two documents is estimated based on the overlap of their sketches [3].

In this study, we make use of hash functions in the form of:

$$h(x) = (Ax+B) \bmod p,$$

- where x is the shingle,
- p is a large prime, which we set to 232 − 1,
- A and B are random integers in the range [1, p].

To find near duplicates based on their sketches, each document is represented by pairs of the h minimum hash values - Mh - and the document ID in the form of <Mh, doc id> and a list of all the pairs for all documents is compiled. This list is then used to build a second list of documents that have a Mh in common in the form of <Mh, doc id1, doc id2>. This second list can then be scanned and the number of Mh that each pair of documents <doc id1, doc id2> have in common can be estimated and then divided by h to calculate the resemblance of the two documents.

URL Based Approaches:

A algorithm, DUST (Different URLs with Similar Text) was defined for similarity between different url have same text. Dust algorithm provides previous crawl entries instead of probing the page contents to mine the dust efficiently. Search engines can increase the efficiency of crawling, reduce indexing overhead, it help in increasing the crawling process and improve page ranking in future which are the benefits provided by the information about the DUST [2].

Reference [2] shows another approach where detecting process was divided into three steps:

- Removal according to URLs
- Remove scattered information in the pages and extract the texts
- Detect with DDW algorithm. Use the DDW algorithm to detect similar pages.

Semantic Approaches:

A method on duplication detection using fuzzy semantic-based string similarity approach was proposed.

The algorithm was developed through four main stages.

- First is prepossesing which includes tokenization, stemming and stop words removing.
- Second is retrieving a list of candidate documents for each wondering document using shingling and Jaccard coefficient similarity ..
- This stage evoke the computation of fuzzy degree of similarity that ranges between two edges: 0 for completely different sentences and 1 for exactly identical sentences.
- The last step is post-processing hereby consecutive sentences are joined to form single paragraphs/sections .

CONCLUSION

In this paper, we have presented a comprehensive survey of up-to-date researches of Duplicate/Near duplicate document detection. We study the main near duplicates document approaches from these techniques we have estimated data duplication detection techniques is cumbersome process and it takes so much time and memory. An effective and efficient deduplication algorithm, which requires minimum number of comparisons for records with less memory and time need to be developed in future and it improves search engine performance.

REFERENCES

[1] Rahul Mahajan, Dr. S.K. Gupta, Mr. Rajeev Bedi., Lin," A Survey of Duplicate And Near Duplicate Techniques", International Journal of Scientific & Engineering Research, Volume 5, Issue 2, February-2014 1531 ISSN 2229-5518

[2] Bassma S. Alsulami, Maysoon F. Abulkhair, Fathy E. Eassa." Near Duplicate Document Detection Survey", Bassma S Alsulami et al, International Journal of Computer Science & Communication Networks,Vol 2(2), 147-151 ISSN:2249-5789

[3] Kyle Williams‡, C. Lee Giles," Near Duplicate Detection in an Academic Digital Library", ‡Information Sciences and Technology, †Computer Science and Engineering