Classification of the Approaches of the Near Duplicate Document Detection and Elimination

Kavita Garg
School of Computer and Information Sciences, MVN University, Palwal, India

&

Dr. Saba Hilal
http://sabahilal.blogspot.in/

&

Dr. Jay Shankar Prasad,
School of Computer and Information Sciences, MVN University, Palwal

Abstract: The area of identification and removal of near duplicate document is important to research as near duplicate pages increase overhead on web. It increases storage space and indexing cost, crawlers produces same type of results and make searching process ineffective. Identical pages on web exists because it contains huge volume of records. Many researches have been done in this area to solve the problem but the problem still exists. The researchers have studied the problem from different perspectives and tried to formulate solutions, however the problem is intensifying as new pages are added to the web. This paper studies previous research work and classify those algorithms and approaches with the intention of structuring the area of duplicate document finding.

Keywords: Near duplicate document; URL normalization; Shingling.

1. Introduction
1.1 Identification and removal of near duplicate document
The area of identification and removal of near duplicate documents defines many approaches and algorithms which are described below.

1.1.1 Hash Index and page size:
To identify near duplicate based on hash index and page is described in [1]. This approach accept input from user and calculate hash index and page size and the calculate values are stored in hash table with pointer to first data source. One drawback of this approach is that if formatting changes or word order changes then the results are not considered duplicate even they contains same content [1].

1.1.2 Instance level constrained clustering
Fingerprint approach and word comparison are not sufficient for finding near duplicate document. In [2], it states a new approach based on instanced level clustering. Instance level clustering is a semi supervised process. It form near duplicate clusters using information such as document attributes and content structure in the clustering process. It incorporates constraints on document attributes and content structures. The instance level constraint is based on document attributes and is the main component of this process.

1.1.3 Fusion of Algorithm
  o Fusion of state of art algorithms
  o Efficient approach using clustering, sentence Feature and fingerprinting.

  o Fusion of “state of art” algorithms
In [3], three algorithms are merged to give better result in the field of identification of near duplicate documents. It fuses shingling algorithm, I-Match, and simhash to give better outcome. First, it takes sequence of words (shingles) from document, after that it imports fingerprints of the document into shingling based simhash algorithm.
Efficient approach using clustering, sentence Feature and fingerprinting

To identify near duplicate pages approach based on sentence level feature and fingerprinting is fused in [4]. The approach is applied in the database of digital documents. First it parses the document and removes HTML tags and Stop words. After parsing sentence level feature and simhash is applied to identify and remove near duplicate document. Finally it uses K-mode clustering to fasten the result.

1.1.4 Semantic approach

- Fuzziness based
- Semantic graphs

- Fuzziness based
In [5] Salha.et.al developed the approach for plagiarism detection using fuzziness semantic based string similarity approach. This algorithm works in four stages:
  First it does preprocessing in which it performs tokenization and stemming and it removes stop words. After preprocessing it finds out suspicious document using shingling and Jaccard coefficient. After that it compares suspicious document with candidate document sentence wise and calculate fuzzy degree of similarity that lies between 0 and 1. If two documents cross the threshold they are considered plagiarized. The last step is post processing in which sentence are combined to form a single paragraph.
- Semantic graphs
In [6] Krishnamurthy Koduvayur Viswanathan et al, suggest approach that recognizes two web documents or graphs. It recognizes pair of documents which are identical in semantic web document. To compute similarity it consider three characteristics and it works in three phases:
  First convert all the documents into a uniform n-triple serialization. After that it generates reduced form to compute similarity measure. Canonical forms are break into further four reduced forms. In last it applies similarity measure to compute the similarity.

1.1.5 Keyword Prioritization

There are many approaches have been discussed above for finding and eliminating near duplicate documents. But the problem still exists. A new approach to find near duplicate documents based on position of word in the document and keywords of document are discussed in [7,8].

In [7] V.A.Narayana et al. discussed approach for detecting and eliminating near duplicate document. First keyword of documents are extracted and stored in a table. Then the similarity score of current document and stored web document is calculated from the keyword of pages. If it crosses the predefined threshold value then the documents are considered similar.

In [8] Dhiraj Khurana et al discussed approach that uses suffix tree to retrieve keywords form the documents. When a new page is introduced it uses suffix tree parser to retrieve the keywords and suffix tree of page is stored in a repository. After generation of suffix tree it is compared with previously stored suffix tree and duplicacy ratio is calculated.

1.1.6 Plagiarism detection

Plagiarism [9] is the most complex problem on web. As huge amount of digital data is available it is difficult to detect plagiarism as it is not exact copy of the data, but it also includes reframing of words, paraphrasing etc. which makes the problem difficult to handle. It is based on natural language to compare documents. In [9] SCAM algorithm is implemented for identification of plagiarized documents. SCAM is standard copy analysis Mechanism. The SCAM algorithm works in four stages:

In first step indexing is performed on data set.
After that text document is preprocessed and stemming, tokenization, stop word removal is applied. After preprocessing index will be searched to find out match between test document and dataset documents. After searching similarity score is calculated between test document and dataset document.
It not simply matches string but it gives result it also indicates that registered document is a superset or subset of test document.

1.1.7 Priority based approach
In [10], Ling et al. proposed new approach based on DOM tree and priority of web page. This algorithm classifies web page into two types: Index type and theme type web page. Index type web page is based on sequence of hyperlinks and theme type web page mainly contains text information and less of linking information. This algorithm works for second type of web pages i.e. theme type web pages. Text information of web pages is extracted with the help of DOM tree, whereas priority of pages were used to find out near duplicate web page. The algorithm works in divided into three modules: First module is, Page information pretreatment module which creates DOM tree to removes the navigation information, advertisement copyright etc. and holds only the body part of the source code. priority information generation module was used to generate priority of text information in descending order. In last, priorities of different pages are compared for similarity and pages that are highly similar are merged.

1.1.8 Pos and Sequence Alignment algorithm approach
Syntactical and sequential alignment algorithms to find out duplicate and near duplicate web pages are proposed in [11]. This algorithm is based on Part of Speech (POS) and LCS (Longest common sequence). POS convert text into POS tags and LCS convert text into a sequence of strings. The algorithm works in three phases. Firstly it does, syntactical processing, without following stemming and removal of stop words. It applies Tree tagger to convert text into a syntactical POS tags based on documents content. It only uses text content and remove HTML information. After this, In second phase it does string optimization. The result of. After applying tagging and tag optimization, a set of strings representing the ordered tags corresponding to each word in the original documents is received. In last phase i.e. matching & ranking phase, String produced in second phase are tested to see the effects of using POS tags using LCS algorithm.

1.1.9 Word Position
In [12], approaches are discussed to detect near duplicate document detection based on word position. In[12] Gaudence Uwamahoro etal proposed an approach based on word position and uses recursion to find near duplicate process. It save storage process and fasten the searching process. After preprocessing it stores each word with its relative position in document and creates an inverted index. It not only searches the word of a query but also considers the sequence of those words to generate the best results.

1.1.10 Detection base on document Title
In [13] another approach is suggested by M. Kiruthika etal to detect duplicate documents based on title of documents. This is different from the other approach and reduces network overhead and optimize the performance. It creates the four modules to find the near duplicate documents. First module i.e. database module is a repository of URLs. Second module is search module which searches user query in database of URL. In third module i.e. extraction module it extracts html coding and title of the web page. In last module it compares the keyword of query with title of document. If match found then the duplicate exist otherwise not.

1.1.11 Supervised and Unsupervised Learning
  - Technique of supervised and unsupervised learning.
  - XNDDF

  - Technique of supervised and unsupervised learning

A new technique based on sentence level feature and supervised learning by Yung-Shen Lin et al is discussed in [14].It works on a large collection of documents. The approach proceeds in three phases: feature selection, similarity measure, and discriminant derivation. To find out the near duplicate document each sentence of the document is retrieved and removal of stop words, stemming is done, the
weight of each term is calculated, and the term with the highest weight is the feature of the sentence. A similarity measure is used to calculate similarity between input document and stored document. It uses support vector machine (SVM) to find a discriminant function which is applied to determine whether a document is a near-duplicate to the input. The sentence-level features reveal the feature of a document.

- **XNDDF**
  
  **XNDDF** is eXtensible Near Duplicate Detection Framework [15]. It proposed an algorithm which uses weighted terms of a document to extract local feature. It uses classifier that computes discriminative function that is best used by the near-duplicate detection algorithm. It provides a framework XNDDF which combines clustering framework, near duplicate detection algorithm and classifier.

### 1.1.12 Machine learning Techniques

In [16], a framework name CEDAR is provided to identify duplicate news stories. On web there are multiple algorithms that work for detecting duplicate news stories in plain text but complexity increases when it is difficult to find identical news stories when the page includes metadata or tags etc. CEDAR stands for Content Extraction and Duplicate Analysis and Recognition. The approach works in two phases.

In first phase it extracts news story web pages that are not website specific. It then uses extracted content which to identify documents that contains same news story content. After this it applies shingling to calculate similarity score of documents.

### 1.1.13 Sporogenous programming approach

The approach used for detecting duplicate in digital library is suggested in [17]. It defines three operations Selection, Crossover and mutation. All the operation are performed in database. Execution of operations applies the de-duplication function. After removing duplicates it applies the sporogeneous functions. These functions take less computational resources in implementation.

### 1.1.14 Comparative study

**A large scale evaluation of algorithms**

In [18], Duplicate web pages detection is done by combining two algorithms shingling and random projection. Random projection based approach gives better precision overall but both of the algorithm works well for finding near duplicate. Both algorithms convert HTML page into sequence of tokens by replacing HTML markup with spaces and it ignores formatting instructions. Shingling based approach works as follows: this algorithm fingerprinted sequence of tokens using 64-bit Rabin finger print which gives a sequence of \( n - k + 1 \) fingerprint called shingles. Let \( s(d) \) be the set of shingles on page \( d \). This algorithm assumes that percentage of unique shingles on which two pages \( d \) and \( d_i \) agree. This algorithm takes the order into account as shingles are based on order of the tokens but ignores the frequency of shingles. Random projection based algorithm works as follows: It ignores the order of tokens i.e. two pages with same token set have same bit string but it considers the frequency of term into account. In this algorithm every token is projected into \( b \)-dimensional space by randomly choosing \( b \) entries from \( \{-1, 1\} \). Thus for this algorithm similarity of two pages is the number of bits their projection agree on. Both algorithms do not return perfect result and returns wrong pairs as near duplicates. To return better results both the algorithm are combined and it works as follows: It computes \( B \)-similar (shingling) pairs and then it filters out those pairs whose \( c \)-similarity (random projection based algorithm) falls below a certain threshold. It chooses threshold by plotting the precision of the combined algorithm for different threshold value.

### 1.1.15 Algorithm used by AltaVista search Engine

The algorithm used by AltaVista search engine has been discussed in [19]. It is based on mathematical concept in which Similarity between two documents can be calculated using fixed size sketch of each page. Similarity between pages is expressed in the form of intersection and size of intersection is calculated by random sampling.
1.1.16 Approach for near duplicate detection in clustering of document

In [20], approach for duplicate document in clustering of documents is proposed. The presence of near duplicate documents unnecessarily increases the information and degrades the performance. The method works in three steps. At first the hash values of each keyword in webpage has been calculated. After calculating hash values, they are stored in temporary database. After storing hash values in database similarity measure is done on the basis of hash values stored in database. If threshold is above 80%, then it is near duplicate web page. When new web page comes, its similarity is compared with already existing web pages in database using equation 1. Sim score is the similarity score.

\[
\text{sim score} = \frac{\text{Number of hash value in } w_1 w_2 ... w_n}{\text{Total } w_1, w_2 ... w_n}.
\]

2. Classification of the approaches of near duplicate document detection.

The area of classification defined by M. Kiruthika et al. [13] is very narrow, classified approaches on based on shingling, similarity metrics and signature based approach. In[14], Suresh Subramanian also does related work in the field of near duplicate approaches on the basis of digital syntactic clustering, random projection based, mathematical approach, word position based, sentence level similarity, edit distance etc. These are the very confined area of approaches.

J Prasanna Kumar [21] classified approach for near duplicate detection based on degree similarity of two documents called shingling by broder. It also studied approach based on (GHV) representation by Bernstein, similarity metrics for the detection of duplicated pages in Web sites by Di Lucca. Approach based on I-Match signatures is also studied. Other approaches consider by this author are based on conceptual tree-similarity measure, Charikar's fingerprinting technique.

In [22], Bassma S. Alsulami et al. classified techniques of near duplicate in 3 areas as given below:

![Figure 1: Classification of the approaches of near duplicate given by Bassma S. Alsulami](image)

In [23], Rahul Mahajan et al. mainly classified approaches of near duplicate detection based on shingling. It studied approach of shingling suggested by Broder. Another approach based on super shingles is given by Fetterly. It studied the area of URL based approach through DUST algorithm. Also the algorithm given by Charikar's for fingerprinting is studied. SPEX algorithm for identifying shared chunks in a collection is also considered.

In [24], approaches related to different URL with same text are studied and work is done in this area only. It studied approach based on host pair with replicated content, copy detection mechanism for digital documents, syntactic clustering of web, and finding duplicated content across multiple databases. The area of near duplicate detection with the problem of different URL with similar text has studied by many authors. In [25], Amit Agarwal
Et al. classified it into two areas given by Bar-Yossef et al. who gave the DUST algorithm and another approach given by Das Gupta who extended the work of previous algorithm.

In [3] Jun fan, does fusion of three approaches based on shingling, I-Match, simhash. It broadly gives classification in the following areas:

a) Shingling  
b) Super shingling  
c) Mini-wise Independent permutation algorithm.  
d) I-Match  
e) Random projection  
f) Simhash  

The area of shingling and super shingling is defined by Broder. Random projection is defined by Charikar. Manku developed the algorithm.  

In [1] Oumair Naseer developed algorithm based on the hash index and page size. It classified area of near duplicate document detection as follows:  
a) Detection of near duplicate in distributed retrieval.  
b) Replicated web collections.  
c) Hash technique to remove duplicate documents.
3. Conclusion
When user search a query over search engine it gets number of results. All the results are not dissimilar and 2 to 3 links are almost similar. These links are duplicate links and many algorithms have been developed to reduce similarity. This paper has studied those approach and classified approaches in different way. The future work will concentrate on reducing similar links to achieve better results.

References


