Efficient Algorithm for Removing Duplicate Documents

Suresh Subramanian, Sivaprakasam

Abstract*— Internet or Web world has a large amount of information, which may be html documents, word, pdf files, audio and video files, images etc. Huge challenges are being faced by the researches to provide the required and related documents to the users according to the user query. Additional overheads are available for researchers pertaining to identify the duplicate and near duplicate web documents. This paper addresses these issues through Genetic Algorithm and Duplicate Web Documents Identification Function is used to improve relevance of retrieved documents by removing the duplicate records from the dataset.

Keywords: Redundancy; Duplicate Web-pages; Inverted Index; Genetic Algorithm; Web Content Mining.

I. INTRODUCTION

As the voluminous of web documents increases on internet, it is a burden to search engines to provide the relevant information to the user query. In addition, more number of duplicates of documents also grows simultaneously on the web which increases the retrieval time and reduces the precision of the retrieved documents. Therefore to identify duplicate and near-duplicate web pages, researchers using the complexity algorithms rather using the classification algorithms.

In general, redundancies are identified by two ways. The first is exact duplication of web pages, when two web pages are having same content. The second, web pages which are similar called as near-duplicated web pages ie) the pages which are similar and must be more than the threshold value.

This paper focuses on detection on and removal of duplicate of web pages and nearly duplicated web pages from the dataset used for finding the fitness function applied in Genetic Algorithm using rank based objective function [1].

Duplicated web pages related work is being discussed in Section 2. A short description about Weighted Inverted Index (WWT) and GAHWM is discussed in Section 3, while Section 4 discuss the proposed algorithms for the duplicate removal function and the experiment results is discussed in Section 5. For the future improvement and conclusion is discussed in Section 6.

II. RELATED WORKS

Many researchers are working on the information retrieval, especially in the domain of web documents duplicates detection.

Manuscript received January 2014.

Suresh Subramanian, Suresh Subramanian, Research Scholar, Department of Computer Science, Karpagam University, Coimbatore, Tamilnadu, India.

Sivaprakasam,Sivaprakasam, Department of Computer Science, Sri Vasavi College, Erode, Tamilnadu, India.

In the modern era, duplicates and near duplicate documents detection is an interesting subject, which helps the user to get the related documents as per the given query.

Broder proposed the Digital Syntactic Clustering (DSC) to detect the near-duplicate web pages[2], it decides the duplicate pages by calculating the number of same shingles in text.

Min-yan Wang et al.[3] come up with an idea for the web page de-duplication method in which the information including original websites and original websites and web titles are extracted to eliminate duplicated web pages based on feature codes with the help of URL hashing. Charikar [4] suggested a method based on random projection of words in documents to detect near duplicate documents and improved the overall precision and recall. G Poonkuzhali et al. [5] proposed a mathematical approach based on signed and rectangular representation is developed to detect and remove the redundant web documents. Li Zhiyi et al. [6] summarizes the situation of duplicated web pages detection technology in China. Gaudence U et al. [7] proposed an algorithm for near duplicate documents detection, which uses the method of Word Positional Based Approach for Document Selection. Metzler D et al. [8] proposed the research method to calculate the sentence level similarity by comparing the word overlap. Junxiu An et al.[14] proposed an algorithm to detect the duplicate web pages based on edit distance.

III. GAHWM AND WEIGHTED INVERTED INDEX

In general, the main aim of the web searching is to find all documents that contains the key terms in user query. For that researchers use inverted index technique, which is the powerful data structure, and store the contents such as words or numbers, to its locations in the document, or a set of documents, or a database file. Inverted Index is a powerful technique used in IR (Information Retrieval) because of its efficiency in retrieving the relevant documents and it has been used for detecting the duplicate documents [9][10]. In [11] inverted index has been used in document detection based on a sentence level, the documents are compared sentence-by-sentence. In [7] inverted index has been used in document detection by word by word positional based approach.

In this paper, we have used the GAHWM method [1], which utilizes the WWT Weighted Web Tool Inverted Index [12], is a data structure which contains distinctive terms with a list of html documents containing frequency of the terms and weightage. The term weight is calculated based on the significance of the terms in identifying a document, that is, the location of the term within the html tags.

Document structure which includes

- Document Name
- Total Number of Words in that Document
- The frequency of Word in the Document
- Weightage of this word in this Document

Table 1 : HTML Tags and their weights

HTML Tag Name	Weight
Title	6
Head, H1, H2, H3	5
A: Anchor	4
B: Bold	3
Body	1

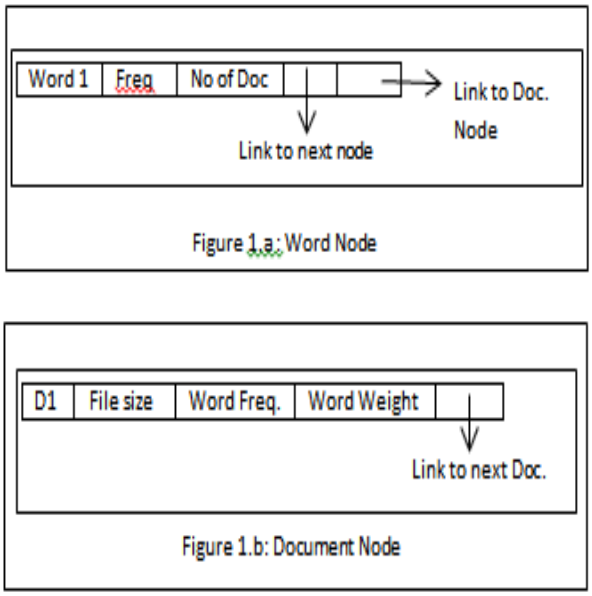

Figure 1: Word and Document structure

IV.WORD WEIGHTAGE BASED APPROACH FOR DOCUMENT SELECTION (WWBADS)

We propose a new approach; which removes the de-duplication documents from the dataset based on web title matching [3] and the number of anchored links available in the documents. If the documents are having the same web title and the number of anchored links are also same, then those documents are identified as de-duplication documents and which are removed from the dataset. In this process; the current dataset has been taken further to remove the nearly duplicate documents based on the weightage of the term and term frequency. Further, the documents are being ordered and listed according to the provided user query using Genetic Algorithm using Rank Based Objective Function [1].

Algorithm

Step 1: Load Wed Documents into memory
Step 2: Remove the Exact duplicate documents from the dataset
Step 3: Remove the Nearly Duplicate documents from the dataset
Step 4: Order the dataset according to the user query using Genetic Algorithm
Step 5: Display the rank based documents list

4.1 Remove The Exact Duplicate Documents

In [3], the documents are considered as duplicate documents if the documents are having the same web title. In [13], further if the documents are having the same number of anchored links inside it could be considered as duplicate documents. In our approach, if documents are having the same web title and same number of anchored links then they are considered as duplicate documents; those duplicate documents are not added into the list; hence the dataset will comprise a list of documents without the exact duplicates.

Algorithm 4.1

Step 1: Load Wed Documents into memory
Step 2: Initialize i=1; Initialize List
Step 3: While i is less than total number of documents : Do
Step 4: Assign j = i + 1
Step 5: If the web title D_i is similar to D_j
  If Yes Then
    Go to step 4
  Else
    Go to step 6
Step 6: If number of anchors in D_i is equal to Anchors in D_j
  Document Di and Dj are redundant
  Assign i=i+1
  Else
  If the document title is not there in the list
  Add the document title to the List
  Assign i=i+1
  End
Step 6: If the document title is not there in the list
  Add the document title to the List
  Assign i=i+1
  End

4.2 Remove Nearly Duplicate Documents

We proposed a method to identify the nearly duplicate documents by using the term weightage and frequency of terms. Using WWT tool, the keyword list which contains document identification, term, frequency and weightage. From the list, terms have been selected randomly and which has been checked whether the term's weightage is same and the frequency is same. Threshold value is assigned to identify the nearly duplicated documents, those nearly documents will be removed from the list.

Algorithm 4.2:

Step 1: Load the document list into memory
Step 2: Create the Word node list and Document Node list
Step 3: Calculate the total number of words in the list
Step 4: Initialize a TempList
Step 5: Assign the threshold value
Step 6: Randomly generate the number k;
  Where k between 1 and total number of words
Step 7: Pick the word, Wk from the list
  While number of Document nodes in the list : Do
  Identify duplicates by weightage and frequency
  Add W_k and document to TempList

Step 8: Continue Step 5 for number of times
Step 9: Read the TempList
Step 10: Check the document ids are repeating for each term
  If Yes then
    Assign pc = pc + 1
  Else
    Assign nc = nc + 1
  End
Step 11: Compare the threshold value is more than the positive count
  If Yes then
    Remove documents from the list
  End

4.3 Prepare Ranking List Using Genetic Algorithm:

In [1], Genetic Algorithm with ranking based objective function has been used to list the documents according to the user query. In our approach, we use the same fitness function to list the documents; however the provided list is free from the exact duplicate documents and nearly duplicate documents.

The fitness function used in genetic algorithm is what determines whether a given solution is optimal or not. In genetic algorithm, solutions are represented as chromosomes. These chromosomes are modified in such ways that in each generation, the fitness value of these chromosomes gets closer the optimal solution. The chromosomes presented in this research contain a list of randomly chosen files. Chromosomes with high fitness value tend to be closer to the optimal solutions, thus making the fitness value of a chromosome determines whether the file is relevant or not. The program uses the fitness function presented from the research in web mining, Genetic Algorithm for HTML Web Content Mining (GAHWM).

$$F(c) = \frac{1}{N} \times \sum_{j=1}^{L} \left(f(d_j) \times \sum_{i=j}^{N} \frac{1}{i} \right) \quad (1)$$

$$f(d_j) = \sum_{i=1}^{K} w_i \quad (2)$$

$$w_i = \frac{K_i}{K} \times \frac{F_i j}{F_j} \times \frac{1}{t_j} \times h_i j \times \log\left(\frac{T}{T_i}\right) \times \log \frac{N}{df_i} \quad (3)$$

V. EXPERIMENT AND RESULTS

The dataset used for the program is a collection of web pages from different universities taken from the World Wide Web Knowledge Base Project containing 8276 files. As a test data we have taken 100 web documents, out of that some documents were exactly duplicated and some documents were nearly duplicated by editing words inside the documents such as editing inside title tag, paragraph text and anchored links .

The program is developed in Windows 7 platform, and is executed in Eclipse SDK 3.3.1.1. Five chromosomes were used for every generation and the chromosomes were populated until generation 10.

The performance of algorithm is measured in terms of recall and precision. The recall is measured by the number of relevant retrieved documents in the collection of all relevant documents with respect to the user query. The precision is measured by the number of relevant retrieved documents in the collection of retrieved documents. Both are formulated as follows:

Recall = (Relevant Retrieved)/Relevant

Precision = (Relevant Retrieved)/Retrieved

A document is said to be relevant if it contains a number of terms greater than or equal to the terms in the user query.

Table 2: Results

GAHWM (for training dataset 100 files)		
	Without WWBADS	With WWBADS
Recall	0.9	0.64
Precision	0.68	0.88

Base from the table, we can observe that the precision score has been increased after the removal of exact and nearly duplicate documents from the training dataset. This incidence occurs since as the function gets more precise, the number of retrieved documents decreases, thus decreasing the chances of fairly relevant files to be included in the output list.

VI.CONCLUSION AND FUTURE WORK

This paper introduces the new approach for identifying the exact duplicate and nearly duplicate documents by combining the approaches of title matching, anchored count, WWT tool and finally GAHWM-WWBADS. The fitness function in GAHWM [1] only considers the weightage of terms in the document list, which reduces the precision of retrieved documents. The beauty of the new approach is simple, and the output list is free from redundancy documents and the output list is having the more promising results. However, future work could be concentrated on implementing the similarity score function in GAHWM-WWBADS and the training dataset could be increased to get the real time results.

REFERNCES

[1] Suresh S, Sivaprakasam, Genetic Algorithm with a ranking based objective function and inverse index representation for web data mining, International Journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 5, September – October (2013), pp. 84-90.

[2] Broder A Z,Glassman S C,Manasse M S,Syntactic clustering of the Web. The Sixth International Conference On World Wide Web 1997.

[3] Min-yan Wang, Dong-Sheng Liu(2009): the Research of web page De-duplication based on web pages Re-shipment Statement, First International Workshop on Database Technology and Applications, pp 271-274.

[4] Charika M S, Similarity estimation techniques from rounding algorithms, in proceedings of 34th Annual ACM symposium on Theory of Computing, (Montreal, Quebec, Canada, 2002) pp. 380-388.

[5] G. Poonkuzhali et al./ International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4026-4032.

[6] Li Zhiyi, Liyang Shijin, National Research on Deleting Duplicated Web Pages: Status and Summary, Library and Information Service,2011, 55(7): pp.118-121.

[7] Gaudence Uwamahoro, Zhang Zuping, Efficient Algorithm for Near Duplicate Documents Detection, International Journal Of Computer Science Issues, Vol 10, issue 2, March 2013.

[8] Metzler, D., Bernstein, Y., Croft, W.B., Moffat, A., Zobel,J., Similarity Measures for Tracking Information Flow, In: The 14th ACM Conference on Information and Knowledge Management (CIKM 2005), 2005, pp.517–524

[9] Zobel J, Alistair Moffat, Inverted files for Text Search Engines, ACM Computing Surveys, Vol . 38, No. 2, article 2006, pp. 1-55.

[10] Ajik Kumar Mahapatra, Sitanath Biswas, Inverted Index Techniques, International Journal of Computer Science Issues, Vol. 8, Issue 4, No. 1, 2011.

[11] Yerra, R., and Yiu Kai, NG., A sentence-Based Copy Detection Approach for Web Documents, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 3613, 2005, pp.557-570.

[12] Ammar A., Rasha S., Genetic Algorithm in Web Search Using Inverted Index Representation, 5th IEEE GCC Conference, 2009.

[13] T. H. Haveliwala, A. Gionis, D. Klein, and P. Indyk.Evaluating strategies for similarity search on the Web. In Proc. 11th International World Wide Web Conference, pages 432-442, May 2002.

[14] Junxi An, Pengsen Chen, The Chinese duplicate web pages detection algorithm based on Edit Distance.Journal of Software, Vol.8, No.7, July 2013

[15] Al-Dallal, A., & Shaker, R. (2009b). Genetic algorithm based mining for HTML document. Retrieved from

http://wwwis.win.tue.nl/bnaic2009/papers/junk/bnaic2009_submission_87.pdf

Mr. Suresh Subramanian is working as IS Analyst in Ahlia University, Kingdom of Bahrain. His research interest is in Web mining and Internet Technology.

Dr. Sivaprakasam is working as a Professor in Sri Vasavi College, Erode, Tamil Nadu, India. His research interests include Data mining, Internet Technology, Web & Caching Technology, Communication Networks and Protocols, Content Distributing Networks.