International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-3, February 2013

A Novel Approach to Perform Document Clustering Using Effectiveness and Efficiency of Simhash

Lavanya Pamulaparty, C.V. Guru Rao

Abstract – Similarity is the most important feature of document clustering as the amount of web documents and the need of integrating documents from the huge multiple repositories, one of the challenging issues is to perform clustering of similar documents efficiently. . A measure of the similarity between two patterns drawn from the same feature space is essential to most clustering procedures. From huge repositories, similar document identification for clustering is costly both in terms of space and time duration, and specially when finding near documents where documents could be added or deleted. In this paper, we try to find the effectiveness of Simhash based similarity measurement technique for detecting the similar documents which are used to perform clustering of documents using novel based K-means clustering method.

Keywords: document clustering, Simhash similarity measure, k-means clustering, near documents, fingerprints.

I. INTRODUCTION

As storage repositories become frequent everywhere it is increasingly difficult to organize and manage growing repository systems. The larger the repository, the more documents will involve which results in increased possibility of similar documents. Identical copies or older versions of documents often become separated and scattered across a repository structure. Consolidating or removing multiple versions of documents becomes desirable. Several studies suggest that approximately 30% of documents in the repositories are similar in their contents. The presence of similar documents may lead to unresolved bug and maintenance related problems by increasing the risk of indexing update/retrieval anomalies[6,7].Identifying all the similar documents benefits many applications as for example.

- For Web search engines, identifying near duplicate Web pages helps to perform focusedCrawling, increase the quality and diversity of query results, and identify spam's [Fetterlyet al. 2003; Conrad et al. 2003; Hen zinger 2006].
- Many Web mining applications rely on the ability to accurately and efficiently identify Near-duplicate objects. They include document clustering [Broder et al. 1997], finding replicated Web collections [Cho et al. 2000], detecting plagiarism [Hoad and Zobel 2003], community mining in a social network site [Spertus et al. 2005], collaborative filtering [Bayardo et al. 2007] and discovering large dense graphs [Gibson et al. 2005].

Quantitative techniques to be used for identifying similar documents could be useful for classification purposes as well as for clustering purposes.

Manual checking is possible if the scale of repository is small, e.g. hundreds or thousands of instances. When the amount of instances increases to millions and more, obviously, it becomes impossible for human beings to check them one by one, which is tedious, costly and prone to error. Resorting to computers for such kind of repeatable job is desired, of which the core is an algorithm to measure the difference between any pair of documents, including duplicated and near duplicated ones. Manku et al. [2] showed that Charikar's Simhash [4] is practically useful identifying near-duplicates in web documents. Simhash is a fingerprint technique enjoying the property that fingerprints of near-duplicates differ only in a small number of bit positions. A Simhash fingerprint is generated for each object. If the fingerprints of two objects are similar, then they are deemed to be near-duplicates. As for a Simhash fingerprint f , Manu et al. developed a technique for identifying whether an existing fingerprint f' differs from f in at most k bits. Their experiments show that for a repository of 8 billion pages, 64-bit SimHash fingerprints and $k = 3$ are reasonable. Another work by Pi et al. [2] confirmed the effect of SimHash and the work by Manku et al; besides, they proposed to do the detection among the results retrieved by a query, i.e. so-called query-based approach. It reduces the problem scale via divide-and-conquer, replacing global search with local search, and it is open to more settings possibly met in application, e.g. smaller k to remove fewer documents under some condition, and bigger k to delete more documents under other condition. In this paper, assumed sets show that SimHash is indeed effective and efficient in detecting both duplicate and near-duplicate (see the two typical examples in TABLE I and II.) among assumed documents repository.

Manuscript published on 28 February 2013.
***** Correspondence Author (s)
Lavanya Pamulaparty, Department of Computer Science and Engineering, Methodist College of Engineering and Technology, Hyderabad, India.
Dr. C. V. Guru Rao, Department of Computer Science and Engineering, SR Engineering College, Warangal, India.

1 274 35 156 647
1 35 156 647
3 213 30 3023 40 4207 44 645 733 542 562
3 213 30 3023 40 4207 44 645 733 542 562
1 35 156 647
2 23 3021 4203 648 732 76 542 561
3 213 30 3023 40 4207 44 645 733 542 562
2 23 3021 4203 648 732 542 561
1 35 156 647
2 23 3021 4203 648 732 542 561
3 213 30 3023 40 4207 44 645 733 77 542 562
2 23 3021 4203 648 732 76 542 561

Table 1: Typical (partial) output sets of document numbers that share common fingerprints.

1 1 274 35 156 647
3 1 35 156 647
3 3 213 30 3023 40 4207 44 645 733 542 562
2 2 23 3021 4203 648 732 76 542 561
2 2 23 3021 4203 648 732 542 561
1 3 213 30 3023 40 4207 44 645 733 77 542 562

Table 2: Lines of Document numbers that share common fingerprints after sorting.

Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity*. Clustering can also be used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases. Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce. For example, exceptional cases in credit card transactions, such as very expensive and frequent purchases, may be of interest as possible fraudulent activity. As a data mining function, cluster analysis can be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Alternatively, it may serve as a preprocessing step for other algorithms, such as characterization, attribute subset selection, and classification, which would then operate on the detected clusters and the selected attributes or features. Data clustering is under vigorous development. Contributing areas of research include data mining, statistics, machine learning, spatial database technology, biology, and marketing. Owing to the huge amounts of data collected in databases, cluster analysis has recently become a highly active topic in data mining research. As a branch of statistics, cluster analysis has been extensively studied for many years, focusing mainly on *distance-based cluster analysis*. Cluster analysis tools based on *k*-means, *k*-medoids, and several other methods have also been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS.[13].In this paper, The k-means clustering algorithm is used because it is efficient in clustering large data sets. This clustering algorithm was developed by MacQueen, and is one of the simplest and the best known unsupervised learning algorithms that solve the well-known clustering problem. The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into *k* clusters, where *k* is a predefined or user-defined constant. The main idea is to define *k* centroids, one for each cluster. The centroid of a cluster is formed in such a way that it is closely related (in terms of similarity function; similarity can be measured by using different methods such as cosine similarity, Euclidean distance, Extended Jaccard,Simhash) to all objects in that cluster.

Basic K-Means Algorithm

1. Choose k number of clusters to be determined
2. Choose k objects randomly as the initial cluster center
3. Repeat
   3.1. Assign each object to their closest cluster.
   3.2. Compute new clusters, i.e. Calculate mean points.
4. Until
   4.1. No changes on cluster centers (i.e. Centroids do not change location any more) OR
   4.2. No object changes its cluster (We may define stopping criteria as well)

Contributions: This paper presents a novel approach of document clustering based on fingerprinting of Web documents using a hash technique called *simhash. A novel* clustering scheme (K-means) is used to organize the processed set of documents on which the similarity technique is applied. This approach speeds up the potential clustering scheme and allows maintaining the clusters efficiently. This makes the approach capable of being used as incremental documental clustering tool or by a real-time document search engine. Our experimental analysis shows effectiveness of *simhash* in document similarity and how it enables to perform document clustering in repositories.

II. PROPOSED WORK

In this paper, K-Means algorithms have been applied to text clustering in a straightforward way. Typically it usages normalized, TF-IDF-weighted vectors and Simhash based similarity. Here, I have illustrated the k-means algorithm using a set of points in n-dimensional vector space for text clustering.

Steps involved in experimental study:

a. Document Representation

Each document is represented as a vector using the **vector space model**. The vector space model also called term vector model is an algebraic model for representing text document (or any object, in general) as vectors of identifiers. For example, TF-IDF weight. Here I have defined DocumentVector class whose instance holds the document and its corresponding representation on vector space.

This is the some of snap-shot of the code :

```
DocumentVector
{
   //Content represents the document(or any other object) to be clustered
   public string Content { get; set; }
   //represents the tf*idf of  each document
   public float[] VectorSpace { get; set; }
}
```

b. TF-IDF

TF-IDF stands for term frequency-inverse document frequency, is a numerical statistics which reflects how important a word is to a document in a collection or corpus, it is the most common weighting method used to describe documents in the Vector Space Model, particularly on IR problems.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The number of times a term occurs in a document is called its term frequency. We can calculate the term frequency for a word as the ratio of number of times the word occurs in the document to the total number of words in the document.
The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.
This is a snap-shot of the code :

```
//Calculates TF-IDF weight for each term t in document d
FindTFIDF(string document, string term)
{
  Use FindTermFrequency(document, term);
  Use  FindInverseDocumentFrequency(term);
  return (tf * idf values) ;
}
```

c. Finding Similarity Measure

I have used simhash based similarity to identify the similarity measure of a document. The method FindsimhashSimilarity takes two argument vecA and vecB as parameter which are vector representation of document A and B, and returns the similarity score which indicates that document A and B are completely exactly similar and near similar respectively.
This is the some of snap-shot of the code :

```
FindSimhashSimilarity(float[] vecA, float[] vecB)
 {
     //Finding out fingerprints for Document A and B
     //Compare(Fingerprints(A,B))
     // return similarity score of documents;
 }
```

d. K-Means Algorithm Implementation

To implement K-Means algorithm I have defined a class Centroid in which documents are assigned during the clustering process.
This is the some of snap-shot of the code :

```
class Centroid
{
 // Here documentcluster class will maintain a list of grouped document by using a global counter
 // Cluster center is initialized for the next iteration, here the count variable holds the value of user defined initial cluster center.
//closestcluster method returns the index of closest cluster center for each document, I have used Simhash based similarity measure to identify the closeness of document.
}
```

III. RELATED WORK

A variety of techniques have been proposed to identify academic plagiarism [11, 12, 13], Web page duplicates [2,3,8, 14] and duplicate database records [15,16]. However, it is noticed that there are very few works on the discussion of detecting near-duplicates among document repository until recently, including [1, 9]. Gong et al. [1] proposed the SimFinder which employ three techniques, namely, the ad hoc term weighting, the discriminate-term selection and the optimization techniques. It is a fingerprinting-based method as well, but takes some special processing while choosing features and their corresponding weights. Muthmann et al. [9] discussed the near-duplicate detection for Web forums which is another critical resource of user-generated content (UGC) on Internet. It is also built on the basis of fingerprinting technique. Though the theoretical basis may be similar, identification of near-duplicate short messages is believed much more difficult considering that: 1) it usually contains less than 200 characters, and there are few effective features to extract; 2) it tends to be informal and error prone; 3) the degree of duplicated and near duplicated is known as more severe than Web documents. All these can be explained by the fact that short messages are very popular and welcome by mobile users, and they are so short to be distributed easily. Outside the area of Similarity detection, *Google* is using *simhash* for finding near duplicate webpages [8]. Gong et al. [7] presents an approach for detecting near-duplicates within a huge repository of short messages. Similarly, SimFinder [18] is a fast algorithm proposed by Pi et al. to identify all near-duplicates in large-scale short text databases.

IV. CONCLUSION

Detection of similar documents for clustering provides several benefits in terms of cluster maintenance, understanding and its proper reuse. In this paper, a novel K-means clustering scheme is used and improved the time performance of the process by using the order of magnitude for document similarity measure using Simhash based technique, and demonstrated its feasibility for use in performing document clustering on repository. On the otherhand, Simhash has significant potential for the fast and large scale similarity detection but comes with increased possibility of its own limitations(multiple hashing). Although there is no special operation taken to process the features in our system, like those appearing in [1,9]. However, we also notice that there is space there for improvement. For example, in further to study the relationship of text document length, ratio of difference and suitable *k's* option. Besides, some advanced NLP (Natural Language Processing) techniques may be applied to improve the outcome. In this paper, it covered some of the fundamental schemes of general purpose similarity measure and clustering for document clustering, all finer modeling will be paid with more computing resource for further work

REFERENCES

[1] C. Gong., Y. Huang., X. Cheng. and S. Bai., "Detecting Near-Duplicates in Large-Scale Short Text Databases," Proc. of PAKDD 2008, LNAI, vol. 5012, pp. 877-883. Springer, Heidelberg
[2] G.S. Manku, A. Jain. and A.D. Sarma, "Detecting Near-Duplicates for Web Crawling," Proc. of 16th International World Wide Web Conference (WWW), 2007.
[3] Xiao, C., Wang, W., Lin, X., Yu, J. X., Wang, G. 2011. "Efficient Similarity Joins for Near Duplicate Detection". ACM Trans. Database System.
[4] M. Charikar, "Similarity Estimation Techniques from Rounding Algorithm," Proc. of 34th Annual Symposium on Theory of Computing (STOC), 2008, pp 380-388.
[5] M.R. Henzinger, "Finding near-duplicate web documents: a large-scale evaluation of algorithms," Proc. of ACM SIGIR, 2006, pp 284-291.
[6] S.Brin, J.Davis and H.Garcia-Molina, "Copy detection mechanisms for digital documents," Proc. of the ACM SIGMOD Annual Conference, San Francisco, CA, 1995.
[7] M.Zini, M.Fabbri and M.Mongelia. "Plagiarism detection through multilevel text comparison," Proc. of the 2nd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, Leeds, U.K., 2006.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

[8] Z.P.Tian, H.J.Lu and W.Y.Ji, "An n-gram-based approach for detecting approximately duplicate data records," International Journal on Digital Libraries, 5(3):325-331, 2001.
[9] K.Muthmann, W.M.Barczynski, F.Brauer and A.Loser,"Near-duplicate detection for web-forums,", 142- 152, International Database Engineering and Applications Symposium(IDEAS), 2009.
[10] J. Dean and S.Ghemawat, "MapReduce: Simplified data processing on large cluster," Proc. of 6th Symposium on Operating System Design and Implementation (OSDI), 2004.
[11] N.Shivakumar and H.Garcia-Molina, "SCAM : A copy detection mechanism for digital documents, " Proc. of 2nd International Conference in Theory and Practice of Digital Libraries, Austin, Texas, 1995.
[12] N.Shivakumar and H.Garnia-Molina, "Finding nearreplicas of documents on the web," Proc. of Workshop on Web Databases, Valencia, Spain, 1998.
[13] Jiawei Han and Micheline Kamber, "Data Mining – concepts and techniques", second edition, Morgon Kaufmann publishers
[14] Z.P.Tian, H.J.Lu and W.Y.Ji, "An n-gram-based approach for detecting approximately duplicate data records," International Journal on Digital Libraries, 5(3):325-331, 2001.

Lavanya Pamulaparty received her Bachelor's degree in computer science from Nagpur University of Kavikulguru Instititute of Science and Technology, Nagpur, India. the Masters degree in Software Engineering from School of Informatics from Jawaharlal Nehru Technological University of Hyderabad, India, and Pursuing the PhD degree in computer science and engineering from JNTUCEH, Hyderabad under the guidance of Dr. C V Guru Rao, Professor, KITS College, Warangal, India.. Presently, she is a professor in the Department of Computer Science and Engineering at Methodist College of Engineering and Technology, Osmania University, Hyderabad, where she has been a faculty member since 2001. Her research interests include information storage and retrieval, Web Mining, Clustering technology and computing, performance evaluation, and information security. She is a senior member of the ACM, Computer Society of India, and a member of the ACM.

Dr. Guru Rao C V received his Bachelor's Degree in Electronics & Communications Engineering from VR Siddhartha Engineering College, Vijayawada, India. He is a double post graduate, with specializations in Electronic Instrumentation and Information Science & Engineering. He received his M.Tech in Electronic Instrumentation from Regional Engineering College, Warangal, India and M.E in Information Science & Engineering from Motilal Nehru Regional Engineering College, Allahabad, India. He is a Doctorate holder in Computer Science & Engineering from Indian Institute of Technology, Kharagpur, India. With 24 years of teaching experience, currently he is the Professor, KITS Warangal, Andhra Pradesh, India. He has more than 25 publications to his credit. He is the Chairman, Board of Studies for Computer Science & Engineering and Information Technology, Kakatiya University, Warangal.Also, he is the Editorial Board member for International Journal of Computational Intelligence Research and Application journal. He is a life member of Indian Society for Technical Education, Instrumentation Society of India, and member of Institution of Engineers, Institution of Electronics & Telecommunications Engineers and Institution of Electrical & Electronics Engineers (USA).