DEDUPLICATION IN DATABASES USING PATTERN MATCHING

[1]Sadhik M S, [2]Eldo P Elias, [3]Surekha Mariam Varghese
[1]M Tech Student, [2]Professor, [3]Head of the Department
[1,2,3] Department of Computer Science,
[1,2,3] Mar Athanasius college of Engineering, Kothamangalam, Kerala

Abstract: ***Semantic duplicates in databases represent today an important data quality challenge which leads to bad decisions. In large databases, sometimes find ourselves with tens of thousands of duplicates, which necessitates an automatic deduplication. Deduplication is a capacity optimization technology that is being used to dramatically improve storage efficiency. For this, it is necessary to detect duplicates, with a fairly reliable method to find as many duplicates as possible and powerful enough to run in a reasonable time. In proposed system, introduce an effective duplicate detection method for automatic deduplication of text files and repeated strings. This will be working with the dataset from WordNet. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. In the proposed system check the strings or text files that are semantically similar. If they are semantically similar, then remove the string and maintain only one copy of the data. To achieve this, KMP algorithm and Levenshtein Distance method used. These algorithms give better results than those of known methods, with a lesser complexity.***

IndexTerms - Deduplication, KMP algorithm, Levenshtein Distance, WordNet.

I. INTRODUCTION

The fast growth of data volumes stored in the cloud storage has led to an increased demand for techniques for saving disk space and network bandwidth. Recently the focus of text analysis has been shifting toward short texts such as microblogs, search queries, search results, ads, and news feeds. Semantic similarity measurement between short texts is a fundamental task and can be used for various applications including text clustering and text classification. The challenge in measuring the similarity between short texts lies in the sparsity, i.e., there are likely to be no term co-occurrence between two texts. To reduce resource consumption, many storage services employ a deduplication technique. Semantic duplicates in databases represent today an important data quality challenge which leads to bad decisions [1]. In large databases, sometimes find ourselves with tens of thousands of duplicates, which necessitates an automatic deduplication. Here mainly focus on data deduplication. It is often called as intelligent compression or single-instance storage. It is a process that eliminates redundant copies of data and reduces storage overhead. Data deduplication techniques ensure that only one unique instance of data is retained on storage media.

When two lines in a database have different identifiers while they represent the same physical reality, we call them semantic duplicates. Deduplication is the complete process from detection to removal of duplicates records. The duplicates treatment in a database is very important and necessary regardless of the action to be undertaken on the data. Duplicates are on average 4% of the data in the databases [1]. When the size of the database becomes larger, their retrieval becomes more expensive and difficult. Duplicates are the cause of many problems that have a significant impact. For example, in an organization, if an employee is represented several times in the database of payroll, obviously he will have as many salaries as represented in the database, representing a loss for the organization.

In proposed system, the main objective is to perform deduplication on text files and repeated strings. The main goal of deduplication is to increase file storage efficiency by eliminating redundant data from files located on the file system. Semantic duplicates in databases represent today an important data quality challenge which leads to bad decisions. In large databases, sometimes find tens of thousands of duplicates, which necessitate an automatic deduplication. So that identifies the repeated words or text files from the database and remove such files and keeps only one copy. Also check the semantically similar words and remove such words. Only one unique copy of each file is stored if the file data is represented more than once in the database. The file data is also compressed to further improve storage efficiency.

II. RELATED WORK

The automatic elimination of duplicate data in a storage system, commonly known as deduplication, is increasingly accepted as an effective technique to reduce storage costs. Thus, it has been applied to different storage types, including archives and backups, primary storage, within solid-state drives, and even to random access memory. Although the general approach to deduplication is shared by all storage types, each poses specific challenges and leads to different trade-offs and solutions. This diversity is often misunderstood, thus underestimating the relevance of new research and development.

In 2003, Peter Christen [2] proposes Standard Blocking (SB) method clusters records into blocks where they share the identical blocking key [8]. A blocking key is defined to be composed from the record attributes in each data set. An example of a blocking key is the first four characters of a surname attributes. A blocking key can also be composed of more than one attribute, for example, a postcode attribute could be combined with an age category attribute.

There is a cost-benefit trade-o_ to be considered in choosing the blocking keys [10]. If the resulting blocks contain a large number of records, then more record pairs than necessary will be generated, leading to an inefficiently large number of comparisons. For example, using a gender attribute as blocking key puts all the available records into two very large blocks. On the other hand, if the blocks of records are too small, then true record pairs may be missed, therefore reducing linkage accuracy (sensitivity).

Sorted Neighbourhood (SN) method [6] sorts the records based on a sorting key and then moves a window of fixed size w sequentially over the sorted records. Records within the window are then paired with each other and included in the candidate record pair list. The use of the window limits the number of possible record pair comparisons for each record to 2w1. The resulting total number of record pair comparisons (assuming two data sets with n records each) of the sorted neighbourhood method is O (wn). Similar to standard blocking, it is

advantageous to do several passes (iterations) with different sorting keys and a smaller window size than one pass only with a large window size [5]. Canopy Clustering with TFIDF (Term Frequency/Inverse Document Frequency) forms blocks of records based on those records placed in the same canopy cluster. A canopy cluster is formed by choosing a record at random from a candidate set of records (initially, all records) and then putting in its cluster all the records within a certain loose threshold distance of it. The record chosen at random and any records within a certain tight threshold distance of it are then removed from the candidate set of records. The number of record pair comparisons resulting from canopy clustering is O(fn2/c) [11] where n is the number of records in each of the two data sets, c is the number of canopies and f is the average number of canopies a record belongs to.

In 2005, Andrew McCallum [5] provides empirical evidence that using canopies for clustering can increase computational efficiency by an order of magnitude without losing any clustering accuracy. The basic clustering approach use here is Greedy Agglomerative Clustering. In order to perform clustering in this domain, we must provide a distance metric for the space of bibliographic citations. A powerful choice for measuring the distance between strings is string edit distance, as calculated by dynamic programming using different costs associated with various transformation rules. There are different transformation costs for (1) deleting a character, (2) deleting a character where the last operation was also a deletion, (3) deleting a period, (4) deleting a character in one string when the other string is currently at a period, (5) substituting one character for another character, (6) substituting a non-alphabetic character for another non-alphabetic character, and (7) deleting a non-alphabetic character.

III. PROPOSED SYSTEM

Here propose a novel technique for the automated deduplication of data and text files in the database is proposed. The method is based on Knuth-Morris-Pratt algorithm and Levenshtein distance method. The algorithms that are mainly used for string comparison. In proposed system, mainly focus on analysis of similar strings. That is repeated strings or text files that are removed from the databases. Also in proposed system, checking the semantically similar words or text files. If any of the strings or text files that are semantically similar to any other words, then that string is also removed from the database and only store one copy of data. It is also applicable to text files.

The proposed system has five stages. They are Data Acquisition, Preprocessing, Blocking method, Matching and Deduplication process. The proposed architecture is shown in Figure 1. It shows how each of the phases is related with its predecessor phases.

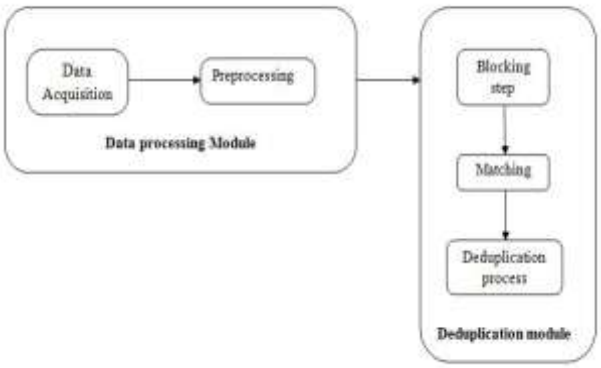

Figure 1: Proposed System Architecture

Data acquisition is the process of sampling data based on the dataset. Synonyms are words that have similar meanings. Here collect the dataset from WordNet. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members.

In preprocessing step, visual analysis of the sample is performed. This analysis is performed by scanning through all the sample data in order to identify the operations of standardization and normalization to be made on the data to be normalized (date formats standardization for example), and secondly to identify \stop words". These operations are necessary for the consideration of the syntactic differences important for the real semantic duplicates. “Stop words” are words that will not be taken into account in the calculation of the blocks during the blocking phase, as it is believed that they are not meaningful for this operation.

The blocking step consists of dividing the data set in small blocks of similar data that could be duplicates, in order to reduce the pair comparison number. In practice, several approaches can be used for blocking. In this work, only interested in the approaches based on blocking functions which take only the used record as input to determine its block. It is called the blocking functions by hashing. The computing time of a block does not depend on the size of the dataset. Therefore, they can be efficiently implemented in management software, at the transaction level, to automatically detect the potential duplicates when creating records. In blocking step use Knuth-Morris-Pratt string searching algorithm (KMP algorithm) for text comparison. The KMP algorithm searches for occurrences of a word “W” within a main text string “S” by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing reexamination of previously matched characters The basic idea behind KMP's algorithm is: whenever detect a mismatch (after some matches), already know some of the characters in the text of next window. Then take advantage of this information to avoid matching the characters that it is anyway match.

The Match step consists of comparing pairs of data to say whether they form a duplicate or not. This step, in general, uses some metrics of similarity distance calculation between the pairs of records. A similarity distance metric is a function which takes as input two records and returns a value considered as their similarity distance. A similarity distance metric can be defined as a positive function which takes two records as input and returns a number which is the said distance of similarity between them. Here use the Levenshtein distance algorithm to check the similarity of strings. Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t).

Deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data. In deduplication process, the repeated strings or records that are identified using blocking and matching step. Here also identify the semantically similar words based on the preprocessing method. So the duplicated words are removed from the database and improve the storage capacity of the database.

IV. CONCLUSION

This work had for objective to propose duplicate detection techniques. The algorithms used in this work improve the efficiency of detection. Through this work, the deduplication performed on text _les and strings. Here it removes only repeated strings or text files from the databases. This is done by using KMP algorithm and Levenshtein distance algorithm. In the first phase use the above algorithms to detect the repeated words and text files in the database. The effectiveness of the proposed system is validating using accuracy obtained by checking the similarity of obtained words.

V. REFERENCES

[1] Ibrahim M N, Amolo-Makama Oph_eli. Fast Semantic Duplicate Detection Techniques in Databases, Journal of Software Engineering and Applications, 2017, 10, 529-545

[2] Bhagyashri, Kelkar, A. and Manwade, K.B. (2012) Identifying Nearly Duplicate Records in Relational Database, International Journal of Computer Science and Information Technology Security , 2, 514-517.

[3] Baxter, R., Christen, P. and Epidemiology, C.F. A Comparison of Fast Blocking Methods for Record Linkage. Proceedings of Workshop Data Cleaning, Record Linkage, and Object Consolidation, Washington DC, August 24-27 2003, 25-27.

[4] Aizawa and Oyama, K. A Fast Linkage Detection Scheme for Multi Source Information Integration. Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration, Tokyo, 8-9 April 2005, 30-39.

[5] Yan, S., Lee, D.W., Kan, M.-Y. and Lee, C.G. Adaptive Sorted Neighborhood Methods for Efficient Record Linkage. Proceedings of the 7thACM/IEEE-CS Joint Conference on Digital Libraries , Vancouver, 18-23 June 2007, 185-194.

[6] McCallum, Nigam, K. and Ungar, L.H. Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, Boston, 20-23 August 2000, 169-178.

[7] Ramos, J. (2003) Using Tf-idf to Determine Word Relevance in Document Queries.

[8] Christen, P. (2007) Performance and Scalability of Fast Blocking Techniques for Deduplication and Data Linkage. Proceedings of the VLDB Endowment, Vienna, 23-28 September, 1, 1253{1264.

[9] Metha, Kadhum, Alnoory, Musbah and Aqel, M. (2011) Performance Evaluation of Similarity Functions for Duplicate Record Detection. Master's Thesis, Middle East University, Beirut.

[10] Bianco, G.D., Galante, R. and Heuser, C.A. A Fast Approach for Parallel Deduplication on Multicore Processors. 26th Symposium on Applied Computing SAC'11, TaiChung, 21-25 March 2011, 1027-1032.

[11] Raghavan, H. and Allan, J. Using Soundex Codes for Indexing Names in ASR Documents. Proceeding SpeechIR `04 Proceedings of the Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval at HLT- NAACL 2004, Boston, 6 May 2004, 22-27.

[12] Christen, P. A Comparison of Personal Name Matching: Techniques and Practical Issues. 6th IEEE International Conference on Data Mining Workshops , Hong Kong, 18-22 December 2006, 290-294.

[13] W. K. Ng, W. Wen, and H. Zhu, \Private data deduplication protocols in cloud storage," Proc. ACM SAC'12, 2012.