A review on methods of Duplicate Detection

Ms. Laxmi R Adhav, Ms. Monali A. Gurule

Abstract— **Duplicate detection is major task in data processing and cleaning. In this paper we discussed about various methods of duplicate detection for a given dataset. Calculating edit distance is the most preferred approach for duplicate detection. Various methods like EdJoin , Winnowing are based on calculating edit distance.**

Strings could be divided into number of small substrings known as Grams. VGRAM algorithm uses this gram based approach. While calculating edit distance strings are divided into number of small strings called Chunks. VChunkJoin algorithm uses this chunking scheme. Comparison is made based on results for best duplicate detection of records.

Index Terms— **CDB, Chunk, Edit Distance, Gram, Virtual CDB.**

I. INTRODUCTION

Databases are of vital importance to IT industry without which any industry can't exist. To store huge amount of data Data-Warehouse is the only solution, but the processing of the data is not supported by data-warehouse. The Data Mining has capability to process on huge amount of data effectively. Data mining process these huge amount of data though ETL (Extraction, Transformation, and Loading) techniques. The industries using these databases require quality of information for their smooth functioning and business. Quality of data indicates clean and error free data. Clean data refers to duplicate records, null or empty records should be avoided. Hence it becomes important to remove duplicate data. Duplicate data detection is the process of detecting multiple records of single object. Duplicate data detection is supported by data preparation stage to store data in uniform manner. Duplicate data detection could use method of Field Matching Techniques which contains three distinct methods as

A. Character-based similarity metrics
B. Token-based similarity metrics
C. Phonetic similarity metrics

A. Character-based similarity metrics

In case of character- based similarity metrics characters are considered as major factor for content identification. It is also divided into various methods as

i. Edit distance,
ii. Affine gap distance,
iii. Smith-Waterman distance,
iv. Jaro distance metric, and
v. *Q*-gram distance

In Edit distance method, edit distance is the number of minimum edit operations required to transform one string to another. The edit operations could be

- Inserting a character
- Deleting a character or
- Replacing a character.

For each of these operations edit cost 1 is considered.

Edit distance has two major advantages over alternative distance measure:

(a) It focus on the ordering of tokens in string
(b) It permits non-trivial alignment of the chunks.

This method is useful for finding typing errors but ineffective for other kind of mismatches. In Q-gram distance method, Q-grams are considered as small substrings of original strings. This method is used for string approximation in which two strings are said to be equal if they share sufficient number of common Q- grams. Letters in q-grams including unigram, bi-gram, trigrams are used for text recognition and spelling checking.

B. Token-based similarity metrics

Character based methods work well for typing mistakes. However typing mistakes could lead to rearrangement of words, which Character based methods can't consider. In case of Token based method focus is given on Atomic Strings. Two strings are said to be *atomic matching* if they match exactly, or one string is prefix or suffix of other.

C. Phonetic similarity metrics

Character and token based similarity metrics focuses on string arrangement of the database records. However strings could be phonetically similar rather than characters or tokens. Soundex is a technique, which assigns identical code of digits to phonetically similar group of consonants. This technique is mainly used to match surnames in a dataset.

This paper mainly aims to conclude that the previous algorithms require large storage space and extra processing

Manuscript received May, 2017.

Ms. **Laxmi R. Adhav** received Masters and Bachelors of Computer Engineering degree from University of Pune. She is currently working as Lecturer Information Technology department at Sandip Foundation, Nashik. *Nashik, Maharastra, India*

Ms. Monali A. Gurule received Masters and Bachelors of Computer Engineering degree from University of Pune. She is currently working as Lecturer in Computer Department at Sandip Foundation, Nashik.
Nashik, Maharastra, India

overhead. They also use cost function to calculate edit distance between strings.

The rest of the paper is organized as section II is discussion about overview of various methods of Duplicate record detection. Section III discusses about VChunkJoin algorithm. Section IV compares previous methods with VChunkJoin and in section V paper is concluded followed by references.

II. History and Overview

There are various algorithms that use Q-gram technique to find similar strings for the purpose of finding Duplicate records or to find similar text in a given database. Ed-Join, Winnowing, VGRAM, etc. are the methods that work on Q-gram technique. VChunkJoin algorithm works with edit similarity join.

Edit based similarity measures considers difference between two objects as the number of edit operations required to convert one string object into another. Similarity and number of edit operations are reciprocal of each other i.e. similarity measure decreases as number of edit operations increases.

A similarity join finds pairs of objects from two data sets such that the pair's similarity value is no less than a given threshold [2]. *Edit similarity join* is a similarity join with considering Edit based similarity measures.

A. Ed-Join

In Ed-Join algorithm all the positional q-grams are extracted and ordered by decreasing order of their id values and increasing order of their locations. They called the sorted array, the 'q-gram' array of the string. This technique improves speed of similarity joins.

Three filtering approach were currently followed as Count filtering, Position filtering and Length filtering. In Ed-Join algorithm new filtering technique called Prefix filtering is introduces as

Prefix Filtering- Let x and y be two q-gram arrays and ed(str(x), str(y)) $\leq \tau$. Then the (q · τ +1)-prefix of x and the (q · τ + 1)-prefix of y must have at least one matching q-gram [1].

Location based mismatch filtering technique is used to detect errors that are within distance τ. In Content based mismatch filtering, they are selecting probing window and looks for the contents of the string. If the content difference in probing window is appropriate edit distance measure, similarity between two strings can be decided.

Ed-Join algorithm is improved version of the All-Pairs-Ed algorithm with two major modifications:

1. A minimum prefix of shorter length is used instead of the standard prefix length required by prefix filtering.

2. Introduced a new Verify algorithm that exploits mismatch based filtering. This filtering effectively reduces the final number of candidate pairs that is verified by the expensive edit distance function.

B. VGRAM

This algorithm mainly focuses on choosing high quality grams of variable length from collection of strings for providing support on queries.

This algorithm uses two variable q_{min} and q_{max} such that $q_{min}<q_{max}$. The grams are selected between q_{min} and q_{max}.

Gram Dictionary is set of sub-strings that are not too frequent. Using gram dictionary set of variable length grams is created. At each step grams are generated for longest substring for matching gram from the dictionary.

If no such gram exists in gram dictionary, then gram is generated of length q_{min}. Gram dictionary is stored as trie. The trie is a tree structure having each edge is labeled as character. According to this algorithm if two strings are within edit distance *k*, then their set of grams also has relevant similarity to *k*. Each, end of gram, is recognized by adding special symbol that does not belong to alphabet set. Path traversal from root to leaf node gives gram in Gram Dictionary. While constructing a gram Dictionary, gram frequencies are collected first then out of frequent grams high quality grams are selected.

C. Winnowing

Document fingerprinting is technique used to detect small partial copied content in a large set of documents.

A fingerprint also contains information about position to describe location from where fingerprint is arrived. In practice, the set of fingerprints is obtained from a small subset of all k-gram hashes.

DEFINITION- WINNOWING *In each window select the minimum hash value. If there is more than one hash with the minimum value, select the rightmost occurrence. Now save all selected hashes as the fingerprints of the document* [3].

In this algorithm 'k-gram' is considered as contiguous substring of length k. Document is divided into k-grams, the value of k is accepted from user.

If hash function is chosen such that probabilities of collision of these hash functions is very small. In such case if two documents share common fingerprint then they share a k-grams as well. Larger value of k indicates matches in the document are not coincidental. So the algorithm avoids matching of strings below the *noise threshold*. The upper bound on performance is expressed as trade-off between the numbers of fingerprints that must be selected and the shortest match that are guaranteed to detect. Substrings are considered as matching if they satisfy two properties as

1. If there is a substring match, at least, as long as the *guarantee threshold*, *t*, then this match is detected and

2. We do not detect any matches shorter than the *noise threshold k*.

The constants *t* and $k \leq t$ are chosen by the user [3]

This algorithm can't ensure copied contents detected completely. Note that there are almost as many *k*-grams as there are characters in the document, as every position in the

Document (except for the last $k - 1$ positions) marks the beginning of a k-gram. Now hash each k-gram and select some subset of these hashes to be the document's *fingerprints*. [3]

III. ABOUT VCHUNKJOIN

All the previous algorithms for similarity joins, compute edit distance to find similarity join which is very costly to compute. They all uses filter and verify techniques for eliminating non matching strings first, then performs verification of matching strings by calculating edit distance. The filter method used, is relatively inexpensive to compute.

Most widespread filtering approach is based on computing GRAMS. Computing Grams can be categorized as fixed length grams and variable length grams. Working with grams results in space and time overhead as computing grams results in large index size which can't entirely accommodated in memory and requires high query processing cost. This algorithm mainly focuses on edit similarity joins. VChunkJoin algorithm divides string into several small substrings, called as chunks. Only chunk's index is stored and processed hence very less space is required as compared to all previous duplicate detection algorithms. For a string of length *l*, we use only 4*l* /avg_chunk_len bytes to store the hashed representation of the chunks. This algorithm uses all previous filtering techniques to find out matching strings like length, count, prefix, location-based mismatch, and content based mismatch filtering. It also uses new filtering approach as rank, chunk number and virtual CDB filtering. A positional q-gram is a q-gram with its position represented in algorithm as of (qgram, pos). For matching q-grams they should have content and position within edit distance τ. This algorithm also uses a class of strong chunking schemes that applies tight lower bound on the number of chunks shared by similar strings. Good chunking scheme guarantees that avalanching effect is not generated. *Avalanching effect* is destruction of chunks while finding similarity distance in single edit distance.

A chunking scheme of this algorithm divides a string into different non-overlapping substrings, each called a chunk. A Chunk Boundary Dictionary (CBD) consists of a number of strings, each encoding a particular rule. In VChunkJoin, family of CBDs namely, tail-restricted CBD scheme is proposed. The chunking schemes take care of destroying at most two chunks per edit operation. A CBD is said to be conflict free only if chunking result is not affected by order of rules applied. Tail restricted CBD's are always conflict free as two substrings are disjoined. With a CBD, we can divide string into set of chunks. The starting position of a chunk in the string is called its *position*. The order of a chunk according to its position in the string is called its *rank*. A Vchunk is a chunk with its position and rank information, represented as (chunk, pos, rank).

A.The VChunkJoin Algorithm

The basic version of VChunkJoin algorithm is a chunk-based counterpart of basic All- Pairs-Ed algorithm with three major modifications as:

1. This algorithm replaces q-grams with Vchunks. The prefix length is shortened to 2τ+1, and significantly reduce inverted index size when q > 2.
2. Rank and chunk number filters are used in the algorithm
3. An improved verification algorithm VerifyVChunk is used.

Two Vchunks u and v are said to be matching (with respect to τ) if

- Their contents are the same, and
- Their positions are within τ, and
- Their ranks are within τ [2].

Chunk Number filtering is the unique filtering technique used for effective filtering. A previous filtering technique like location based mismatch filtering is also used. Content based mismatch filtering could not be directly applied. A new technique called virtual CBD filtering is applied. In case of virtual CBD, only resulting chunk numbers are stored for additional CDBs instead of storing and indexing the resulting chunks. Hence additional virtual CBD storage overhead per string is less.

B. CBD SELECTION

The basic requirement of the CBD is to be able to segment all strings into at least 2τ+ 1 chunk.

Out of many CBDs, the CBD giving best query performance is selected. The cost function of optimization is hard to determine, hence selection of optimal CDB is also hard. The algorithm uses greedy approach to select a good CBD for a given string automatically.

IV. COMPARISON

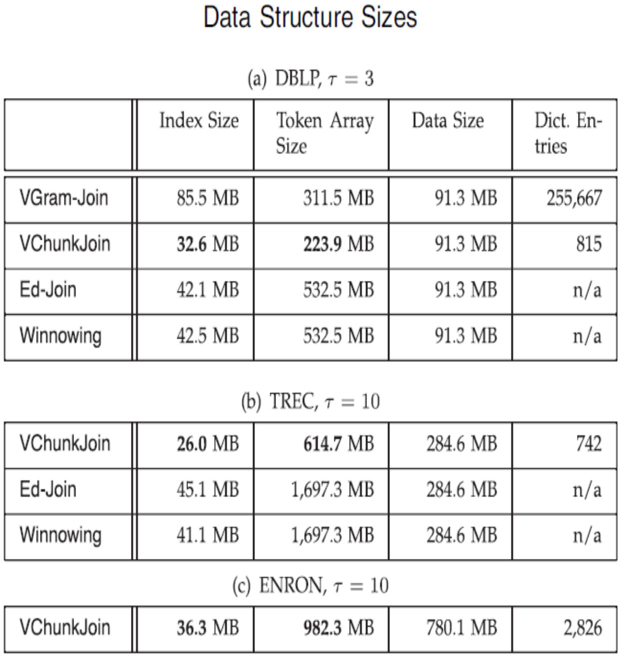
For experimental analysis this algorithm uses several publically real datasets like

- IMDB is a collection of actor names downloaded from IMDB website.
- DBLP is a snapshot of the bibliography records from the DBLP website.
- TREC is from the TREC-9 Filtering Track Collections.
- UNIREF is the UniRef90 protein sequence data from the UniProt project.
- ENRON This data set is from the Enron email collection [2].

For comparison following parameters are considered as

- The average length of the prefixes;
- The total size of the indexes;
- The number of the candidate pairs formed after probing the inverted index (denoted as CAND-1).
- The number of candidate pairs before the final edit distance verification (denoted as CAND-2);
- The running time.

Table 1: Data Structure Sizes



Data Structure Sizes

(a) DBLP, $\tau = 3$

	Index Size	Token Array Size	Data Size	Dict. Entries
VGram-Join	85.5 MB	311.5 MB	91.3 MB	255,667
VChunkJoin	**32.6** MB	**223.9** MB	91.3 MB	815
Ed-Join	42.1 MB	532.5 MB	91.3 MB	n/a
Winnowing	42.5 MB	532.5 MB	91.3 MB	n/a

(b) TREC, $\tau = 10$

VChunkJoin	**26.0** MB	**614.7** MB	284.6 MB	742
Ed-Join	45.1 MB	1,697.3 MB	284.6 MB	n/a
Winnowing	41.1 MB	1,697.3 MB	284.6 MB	n/a

(c) ENRON, $\tau = 10$

VChunkJoin	**36.3** MB	**982.3** MB	780.1 MB	2,826

Table 1 shows comparison of data structure sizes for DBLP, TREC and ENRON datasets. It is observed that VChunkJoin algorithm has smallest index size and data array size. The experiment on ENRON dataset using of Ed-Join and Winnowing cannot be performed because of large index size.

In case of q-gram based edit similarity join algorithms like Ed-Join and Winnowing, the total size of the q-grams is about six times larger than the size of the text strings.

It is usually seen that the prefix length increases linearly with the minimum edit distance value. It is experimentally stated that the Ed-Join and Winnowing algorithms shows similar prefix lengths on the datasets, whereas VChunkJoin shows the shortest prefix length in the experimental datasets.

The index size of VchunkJoin is slightly greater than Ed-Join as it contains one additional rank field.

Winnowing algorithm shows greatest CAND-1size because some common qgrams are also indexed. Content filtering used by all algorithms shows similar CAND-2 value, but when edit distance measure value is sufficiently large then VchunkJoin algorithm performs well.

V. CONCLUSION

In this paper, we discussed about various methods of finding duplicates in a given dataset. The gram based methods like Ed-Join, Winnowing, VGRAM gives results in terms of duplicate detection. Out of these old methods VGRAM method is more efficient. VChunkJoin is a novel approach for duplicate detection for edit similarity joins. With the help of experimental analysis conclusion can be made that VChunkJoin algorithm requires less index storage space yet gives better result. A new class of chunking scheme is devised based on Chunk Boundary Dictionary.

ACKNOWLEDGMENT

I owe greatest thanks to all people who helped and supported me during writing this paper. I want to convey my deepest thanks to the Guide of this paper for guiding and correcting me at each stage. I would also like to thank my Institution, faculty members and Principal without whom this paper would has been a distant reality. I also extend my grate thanks to my family members and well-wishers.

IV. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

REFERENCES

REFERENCES

[1] C. Xiao, W. Wang, and X. Lin, “Ed-Join: An Efficient Algorithm for Similarity Joins with Edit Distance Constraints*,” Proc. VLDB Endowment, vol. 1, no. 1, pp. 933-944, 2008.*

[2] Wei Wang, Jianbin Qin, Chuan Xiao, Xuemin Lin, and Heng Tao Shen, Senior Member, IEEE, “VChunkJoin: An Efficient Algorithm for Edit Similarity Joins.” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 8, AUGUST 2013.*

[3] S. Schleimer, D.S. Wilkerson, and A. Aiken, “Winnowing: Local Algorithms for Document Fingerprinting,” *Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 76-85, 2003.*

[4] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, “Duplicate record detection: A survey*,” TKDE, vol. 19, no. 1, pp. 1–16, 2007.*

[5] C. Li, B. Wang, and X. Yang, “VGRAM: Improving Performance of Approximate Queries on String Collections using Variable-Length Grams,” *Proc. 33rd Int’l Conf. Very Large Databases (VLDB), 2007.*

[6] C. Xiao, W. Wang, X. Lin, and J. X. Yu, “Efficient similarity joins for near duplicate detection,” *in WWW, 2008.*

[7] A. Mazeika, M.H. Bo¨hlen, N. Koudas, and D. Srivastava, “Estimating the Selectivity of Approximate String Queries,” *ACM Trans. Database Systems, vol. 32, no. 2, article 12, 2007.*

[8] A. Arasu, V. Ganti, and R. Kaushik. Efficient exact set-similarity joins in *VLDB, 2006.*

[9] S. Melnik and H. Garcia-Molina “Adaptive algorithms for set containment joins”, *ACM Trans. Database Syst., 28:56–99, 2003.*

[10] Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, and Cristian-Augustin Saita “Declarative data cleaning: Language, model, and algorithms” *in Proceedings of the 27th International Conference on Very Large Databases (VLDB 2001), pages 371.380, 2001.*

Laxmi R. Adhav received Masters and Bachelors of Computer Engineering degree from University of Pune. She is currently working as Lecturer in Information Technology department at Sandip Foundation, Nashik.

Monali A. Gurule received Masters and Bachelors of Computer Engineering degree from University of Pune. She is currently working as Lecturer in Computer Department at Sandip Foundation, Nashik.