

A Review on Data Normalization Techniques

Dr. M.Ramasubramanian, Professor, Department of Computer Science & Engineering, Krupajal Engineering College, Bhubaneswar
Rashmi Manjari Jayasingh, Assistant Professor, Department of Computer Science & Engineering, NM Institute Of Engineering & Tech, Bhubaneswar
Sourav Ghorai, UG Student, Department of Computer Science & Engineering, Krupajal Engineering College, Bhubaneswar
Rajendra Prasad Behera, UG Student, Department of Computer Science & Engineering, Krupajal Engineering College, Bhubaneswar

Abstract: A bulk data is generated from various sources. The sources may provide duplicate data with some representative changes. To mine such big data and create representative data is a challenging task. The data importance increases when it is linked with similar resources and similar data is fused in one source. Lot of research work has been done to provide a single representative data of all real world entities by removing the duplicate records. This task is called record normalization. This paper aims to study various existing data normalization techniques along with its advantages and limitations. Based on the existing system study a new technique is proposed.

Index Terms - Record normalization, data clustering, data fusion, data linkage, data integration.

1. INTRODUCTION

The bulk data is generated in the world wide web. Based on the user search parameter the data is collected from various sources. The structured data contents are stored in web warehouses containing web databases and web tables. The relevant data collection is done from various warehouses like Google, Bing Shopping. Google Scholar is an important mining domain. It is known as web data integration. In web data integration, the structured data should be matched automatically coming from various web warehouses. A data containing similar records, records that point to the same entity should be grouped together as a standard record set.

The result set generated after searching a query on the search engine generates the redundant results, showing multiple entries of the same record coming from various sources. This record representation contains duplicate and unnecessary entries. Such a result set is inconvenient to the end user for analysis.

Record normalization is important in a variety of domains. For example, in the case of research publication domain Citeseer or Google Scholar are important integrator websites that collect data from various sources from automatic data collection techniques. The data is displayed to the user based on the user query. The data should be clear and in normalized form. The search result should be:

1. Best match search
2. Data should be de-duplicated

If ad-hoc approaches for data matching is followed or all the matched records are displayed to the end user then it will be very frustrating for the end user to sort and extract useful information from the generated result set. Ad-hoc extraction of records may lead to records with missing value or incorrect data representation.

Record normalization is a challenging problem because various resources provide the same data in various formats. There is conflict in data which is collected from various sources due to erroneous data, incomplete data, different data representation or missing some attribute values.

Consider an example: User fire a search query as: "Data integration: the teenage years", based on the title matching various records are fetched like:

TABLE I. PUBLICATION RECORDS

Sr. No.	Author	Title	Venue	Date	Pages
1.	Halevy, A.; Rajaraman A.; Ordille, J.	Data integration: the teenage years	in proc 32nd int conf on Very large data bases	2006	
2.	A. Halevy, A. Rajaraman, J. Ordille	Data integration: the teenage years	in VLDB	2006	9-16
3.	A. Halevy, A. Rajaraman, J. Ordille	Data integration: the teenage years	in proc 32nd conf on Very large data bases	2006	pp.9-16
4.	A. Halevy, A. Rajaraman, J. Ordille	Data integration: the teenage years		2006	9-16

In the above table, the same author name representation is in the various forms. Venue and pages contain some missing value or variation in representation of the same data. By analyzing all the records the normal record should be generated as:

TABLE II. NORMALIZED RECORDS

Sr. No	Author	Title	Venue	Date	Pages
1	A. Halevy, A. Rajaraman, J. Ordille	Data integration: the teenage years	in proc 32nd int conf on Very large data bases	2006	pp.9-16

For normalized record generation record level duplication should be removed. With the record level comparison, field level comparison should be done. In the above example author, title, venue data and pages are various fields in a

record. For more precision the values in a field should be normalized. In the following section literature survey is discussed followed by problem formulation. Based on the analyzed problem a new system is proposed in section IV followed by the conclusion.

2. LITERATURE SURVEY

Culotta et al. proposes a record normalization at the very first time. The normalization technique is also called Canonicalization. This is a process of converting the data in one standard canonical form by analyzing various parameters. In this paper the author proposes a technique for the record normalization in databases. For normalization 3 types of solutions are provided. The solution is in terms of field values. These solutions are enlisted as follows:

1. String edit distance to find most relevant central record
2. Optimize the edit distance parameter
3. Feature-based solution to improve performance of Canonicalization.

This paper does not consider the value component level normalization and hence the normalized record database contains many instances of repetitive data and unnecessary normalized records [2].

Swoosh treats the data duplication problem as an entity relationship problem. The problem is like a black box function. This back box matches and merges the records. The ER algorithm is defined to invoke these functions. The system generates de-duplicate records but does not generate the normalized records. It increases the complexity of record matching problems [3].

Wick et al. proposes a technique for data integration using schema matching. It also focuses on coreference resolution, record canonicalization. For implementation it uses a discriminatively-trained model. Due to combined objectives, the system complexity increases. The paper only deal with field level record matching and not at the value level and hence the system do no generate the complete normalization records.[4]

Tejada et al. proposes a technique for database record normalization called object normalization. The system collects the data from various web sources and saves collectively in a database. At the time of search these database objects are normalized with duplication removal. The system uses attribute ranking as well as string ranking in attribute, based on the user's confidence score. [5]

Wang et al. works on shopping dataset. The dataset is normalized in terms of records. It works on data integration and data cleaning. It works on record marching and replacing the missing values with the most relevant values. It also corrects the data which is best suitable to the record by comparing the other dataset record entries. It do not work on value level and working globally on field level normalization.[6]

Chaturvedi et al. works on pattern discovery in the records. This technique does not focus on data normalization and removal of duplicate records but it extracts patterns from duplicate records and finds the most important and prevalent patterns in the dataset. This approach can be applicable for data normalization.[7]

Dragut et al. works on automatic labeling called as Label normalization. The label normalization is used for record normalization and assigning meaningful labels to the elements of an integrated query interface. It works on field level labeling and assigns labels to each attribute within the global interface. [8]

S. Raunich et. al. proposes an ATOM system. The Atom system works on Ontology merging which is nothing but a record normalization. But in the merging phase user involvement is required. The approach should be automated with less involvement of the end user [9].

Yongquan Dong et. al. works on automatic record normalization. The normalization is performed at three levels: record level, field level and value level. The normalization accuracy increases at each level of data pruning. The duplicate records are removed. A single entry is created by analyzing the duplicate entries. The related entries are not clubbed together. A single representation of the record is created. For more informative data representation data should be normalized and linked together. The data is processed with string operation functions and no natural language processing(NLP) techniques are used. NLP techniques may create more accurate results with less processing.[1].

3. PROBLEM FORMULATION

Let $E1$ be the real world entity. R_e is a set of records collected from various sources representing the same entity $E1$. $R_e = \{R1, R2, \dots, R_p\}$. This record is the collection of various fields. In each field various string values are present. Let FS be the set of fields $FS = \{f1, f2, \dots, f_q\}$ and $r_i[f_i]$ is the value in the field f_i . There is a need to define the problem as record normalization and linking problems. From the set of R_e , generate a new customized record that represents the entity $E1$ more accurately in a very descriptive manner using natural language processing techniques.

The records from other entities like $E1$ should be linked together by matching the field and value level components.

4. PROPOSED METHODOLOGY

A. Preliminaries:

1. Frequency Ranker:

The frequency ranker ranks the mostly occurred unit u in the list of distinct units.

$$FR(U) = [u_1, u_2, \dots, u_p]$$

Where, $FR(U)$ is a sorted list in the descending order of units based on the occurrence frequency.

2. Length Ranker:

The length ranker ranks the length of unit u in the list

of distinct units.

$$LR(U) = [u_1, u_2, \dots, u_p]$$

Where, LR(U) is a sorted list in the descending order of units based on the number of characters present in the unit.

3. Centroid Ranker:

This gives the ordered list of distinct units. It initially calculates the similarity score among units and finds the centroid. The centroid is calculated as:

$$UCS(u) = \frac{1}{|U|^2} \sum_{u \in U^F} AuAvSM(u, v)$$

Where,

U = bag of units

U' = distinct units in U

Au and Av: occurrence frequency of u and v.

4. Edit-distance based Similarity measure:

The number of edits required to transform one string to another. Edit distance based similarity between two string a and b is given as:

$$Sim - ed(a, b) = \frac{ed(a, b)}{(|a|, |b|)}$$

|a| and |b| is lengths of a and b respectively.

5. bigram similarity measure:

This distance is based on 2- character substring present in string. The similarity measure between string a and b is given as:

$$Sim\text{-bigram}(a, b) = \frac{2 * (|bigram(a) \cap bigram(b)|)}{(|bigram(a)| + |bigram(b)|)}$$

Bigram(a) and bigram(b) are 2-grams of a and b respectively.

6. Feature-based rankers:

Feature based rankers are divided in 2 sections:

a. Strategy feature:

This is a binary indicator that indicates the unit is a representative unit ranked by some ranking criteria.

b. Text Feature:

This feature examines the property of string. It checks if the string is acronyms or abbreviations of a certain representative string or not. For example: conf is an abbreviation of conference whereas VLDB is an acronym for Very Large Databases.

7. Collocation:

Collocation is a sequence of consecutive terms with the inverse term document frequency (idf) value less than the given threshold. N-collocation defines the consecutive n terms.

8. Sub-collocation

Is the substring of n-collocation string with k consecutive terms. For example "in the conference" is the sub-collocation of "in the conference of VLDB".

9. Template collocation:

An n- collocation term is called a template collocation if its inverse term document frequency (idf) is greater than the given threshold.

10. Twin template collocation:

The terms tc1 and tc2 are twin collocation if it satisfies the following conditions:

$$P(tc1, tc2) > p(tc1, tc), \text{ for all } tc \in TC \text{ and } tc1 \triangleleft tc2$$

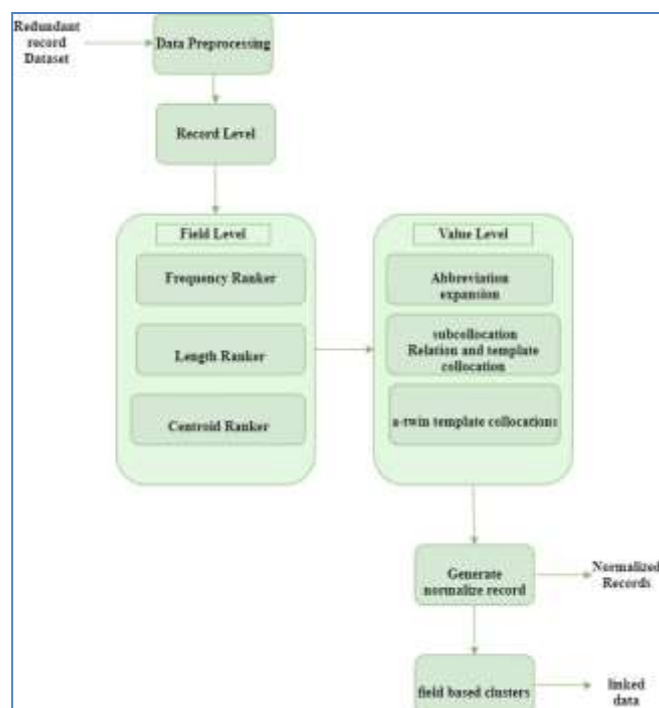
$$(p(tc1, tc2)) / (p(tc2)) > \text{threshold}$$

B. System Architecture

Redundant record Set is input to the system. After processing, the system generates Non-redundant normalized record set along with the data linking. The data processing is mainly categorized in 5 sections:

5. Data preprocessing
6. Record Level Normalization
7. Field Level Normalization and
8. Value Level Normalization.
9. Field Based Clusters

Following figure shows the architecture of the system.



C. System Description:

1. Pre-processing step:

Initially from the given data each record is separated and from each record various fields are extracted.

For Example: Consider the following citation:

A. Halevy, A. Rajaraman, J. Ordille, "Data integration: the teenage years", in proc 32nd int conf on Very large data bases, 2006, pp.9-16

In this citation the following fields can be separated as:

Author: A. Halevy, A. Rajaraman, J. Ordille

Title: Data integration: the teenage years

Venue: in proc 32nd int conf on Very large data bases

Date: 2006

Pages: pp.9-16

All the comma separated values are extracted and added in the respective fields.

1. Record selection:

The record is generated with the combination of various fields. There should be all values present in each field so that a complete informatory citation can be generated as a representative of all redundant data. This is a selection criterion for record level data filtering. The selected records are further processed using field and value level.

2. Field Selection:

The normalized record is generated by combining the most descriptive features of all fields. From all the records each field data is normalized and then a new record is generated. For record normalization frequency ranker, length ranker, centroid rankers and feature based ranker are used.

3. Value Selection:

The values of each field are extracted. The abbreviation and acronyms are replaced by Mining Abbreviation-Definition Pairs algorithm. Afterwards its collocation, sub collocation and twin-collocation is identified using the Mining TemplateCollocation-SubCollocation Pairs (MTS) algorithm. A normalized record is generated at the value level. For mining abbreviations NLP technique is used that helps to find n-gram nouns and its respective aberrations.

4. Field based Clusters:

Based on the normalized value extracted for each field in the record, relevant records are linked as per the field value details.

V. CONCLUSION

In this paper, a study of various existing systems is proposed. The record normalization is carried out in a variety of ways like duplication removal, record level normalization, field level normalization and value level normalization. Some techniques require user involvement in record normalization working for accuracy improvement. These techniques use string processing functions to find record level, field level and value level duplications and alternate text used in the same context. None of the existing techniques uses NLP for value level normalization. The proposed system generates Normalized records by removing duplicate entries that point to the same entity. Data normalization processing is applied at tree levels: record level, Field level and value level. The precision of de-duplication increases from record level to value level. The values are normalized using NLP n-gram technique. Along with the duplication removal similar entities are grouped together using field and value level data comparison. The grouped data is linked together to generate more representative data.

REFERENCES

- [1] Yongquan Dong, Eduard C. Dragut and Weiyi Meng, "Normalization of Duplicate Records from Multiple Sources", in IEEE Transactions on Knowledge and Data Engineering, Vol. 31, Issue 4, April 2019, pp. 769 – 782
- [2] A. Culotta, M. Wick, R. Hall, M. Marzilli, and A. McCallum, "Canonicalization of database records using adaptive similarity measures," in SIGKDD, 2007, pp. 201–209.
- [3] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom, "Swoosh: A generic approach to entity resolution," VLDBJ, vol. 18, no. 1, pp. 255–276, 2009.
- [4] M. L. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum, "A unified approach for schema matching, coreference and canonicalization," in SIGKDD, 2008, pp. 722–730.
- [5] S. Tejada, C. A. Knoblock, and S. Minton, "Learning object identification rules for information integration," Inf. Sys., vol. 26, no. 8, pp. 607–633, 2001.
- [6] L. Wang, R. Zhang, C. Sha, X. He, and A. Zhou, "A hybrid framework for product normalization in online shopping," in DASFAA, vol. 7826, 2013, pp. 370–384.
- [7] S. Chaturvedi and et al., "Automating pattern discovery for rule based data standardization systems," in ICDE, 2013, pp. 1231-1241.
- [8] E. C. Dragut, C. Yu, and W. Meng, "Meaningful labeling of integrated query interfaces," in VLDB, 2006, pp. 679-690.
- [9] S. Raunich and E. Rahm, "Atom: Automatic target-driven ontology merging," in ICDE, 2011, pp. 1276-1279.