NORMALIZATION OF DUPLICATE RECORDS FROM MULTIPLE SOURCES

[1]M. sangeetha,[2]Dr. Gulab singh chauhan
[1]M.tech, Department of Computer Science, VAAGESWARI COLLEGE OF ENGINEERING Thimmapur, Karimnagar, Telangana, INDIA, H.no: 20S41D5815, ms.sangeetha0196@gmail.com
[2]Department of Computer Science, VAAGESWARI COLLEGE OF ENGINEERING Thimmapur, Karimnagar, Telangana, INDIA, gulsinchu@gmail.com

ABSTRACT: Data consolidation is a challenging issue in data integration. The usefulness of data increases when it is linked and fused with other data from numerous (Web) sources. The promise of Big Data hinges upon addressing several big data integration challenges, such as record linkage at scale, real-time data fusion, and integrating Deep Web. Although much work has been conducted on these problems, there is limited work on creating a uniform, standard record from a group of records corresponding to the same realworld entity. We refer to this task as record normalization. Such a record representation, coined normalized record, is important for both front-end and back-end applications. In this paper, we formalize the record normalization problem, present in-depth analysis of normalization granularity levels (e.g., record, field, and value-component) and of normalization forms (e.g., typical versus complete). We propose a comprehensive framework for computing the normalized record. The proposed framework includes a suit of record normalization methods, from naive ones, which use only the information gathered from records themselves, to complex strategies, which globally mine a group of duplicate records before selecting a value for an attribute of a normalized record. We conducted extensive empirical studies with all the proposed methods. We indicate the weaknesses and strengths of each of them and recommend the ones to be used in practice.

Keywords *– Record normalization, data quality, data fusion, web data integration, deep web.*

1. INTRODUCTION

THE Web has evolved into a data-rich repository containing a large amount of structured content spread across millions of sources. The usefulness of Web data increases exponentially (e.g., building knowledge bases, Web-scale data analytics) when it is linked across numerous sources. Structured data on the Web resides in Web databases [1] and Web tables [2]. Web data integration is an important component of many applications collecting data from Web databases, such as Web data warehousing (e.g., Google and Bing Shopping; Google Scholar), data aggregation (e.g., product and service reviews), and metasearching [3]. Integration systems at Web scale need to automatically match records from different sources that refer to the same real-world entity [4], [5], [6], find the true matching records among them and turn this set of records into a standard record for the consumption of users or other applications. There is a large body of work on the record matching problem [7] and the truth discovery problem [8]. The record matching problem is also referred to as duplicate record detection [9], record linkage [10], object identification [11], entity resolution [12], or deduplication [13] and the truth discovery problem is also called as truth finding [14] or fact finding [15] - a key problem in data fusion [16], [17]. In this paper, we assume that the tasks of record matching and truth discovery have been performed and that the groups of true matching records have thus been identified. Our goal is to generate a uniform, standard record for each group of true matching records for end-user consumption. We call the generated record the normalized record. We call the problem of computing the normalized record for a group of matching records the record normalization problem (RNP), and it is the focus of this work. RNP is another specific interesting problem in data fusion. Record normalization is important in many application domains.

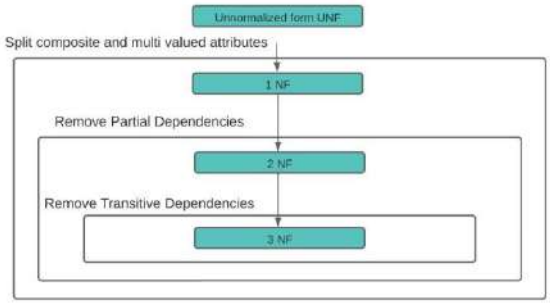

Fig.1: Example figure

For example, in the research publication domain, although the integrator website, such as Citeseer or Google Scholar, contains records gathered from a variety of sources using automated extraction techniques, it must display a normalized record to users. Otherwise, it is unclear what can be presented to users: (i) present the entire group of matching records or (ii) simply present some random record from the group, to just name a couple of ad-hoc approaches. Either of these choices can lead to a frustrating experience for a user, because in (i) the user needs to sort/browse through a potentially large number of duplicate records, and in (ii) we run the risk of presenting a record with missing or incorrect pieces of data. Record normalization is a challenging problem because different Web sources may represent the attribute values of an entity in different ways or even provide conflicting data. Conflicting data may occur because of incomplete data, different data representations, missing attribute values, and even erroneous data. For example, Table 1 contains four records corresponding to the same entity (publication). They are extracted from different websites. Record Rnorm is constructed by hand for illustration purposes. One notices that the same publication has different representations in different websites. For instance, the field author uses the format ”last-name, first-name-initial” in the record Ra, but the values of the same field in the records Rb, Rc, and Rd use the format ”first-name-initial. last-name”. One can also observe that the value of the field pages is absent in Ra. The field venue has incomplete values in three of the four records and has no value in Rd; it contains the abbreviations “proc”, “int”, “conf” to represent “proceedings”, “international” and “conference”, respectively, in the records Ra and Rc; it contains the acronym “VLDB” to represent “Very Large Data Bases” while missing “proceedings of the 32nd international conference on” in Rb. Some values of the attributes of Rnorm cannot be acquired directly from the given set of matching records, such as the first names of the authors. They could be obtained by mining external sources, such as a search engine. In this paper, we focus on the besteffort record normalization: we compute Rnorm from the set of matching records and do not explore external sources.

2. LITERATURE REVIEW

Reclaiming space from duplicate files in a serverless distributed file system

The Farsite distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes 1) convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and 2) SALAD, a SelfArranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant.

Dupless: Server aided encryption for deduplicated storage

Cloud storage service providers such as Dropbox, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded. Should clients conventionally encrypt their files, however, savings are

lost. Message-locked encryption (the most prominent manifestation of which is convergent encryption) resolves this tension. However it is inherently subject to brute-force attacks that can recover files falling into a known set. We propose an architecture that provides secure deduplicated storage resisting brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees. We show that encryption for deduplicated storage can achieve performance and space savings close to that of using the storage service with plaintext data.

Efficient dispersal of information for security, load balancing, and fault tolerance

An Information Dispersal Algorithm (IDA) is developed that breaks a file F of length L = (F (into n pieces F,, 1 5 i 5 n, each of length (F, 1 = L/m, so that every m pieces suffice for reconstructing F. Dispersal and reconstruction are computationally efficient. The sum of the lengths (F, 1 is (n/m). L. Since n/m can be chosen to be close to I, the IDA is space eflicient. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks, to fault-tolerant and efficient transmission of information in networks, and to communi-cations between processors in parallel computers. For the latter problem provably time-efftcient and highly fault-tolerant routing on the n-cube is achieved, using just constant size buffers. Categories and Subject Descriptors: E.4 [Coding and Information Theory]: nonsecret encoding schemes

Secure deduplication with efficient and reliable convergent key management

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage. Although convergent encryption has been extensively adopted for secure deduplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, we propose Dekey , a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments.

Proofs of ownership in remote storage systems

Cloud storage systems are becoming increasingly popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of repeating data. Client-side deduplication attempts to identify deduplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file. (In parallel to our work, a subset of these attacks were recently introduced in the wild with respect to the Dropbox file synchronization service.) To overcome such attacks, we introduce the notion of proofs-ofownership (PoWs), which lets a client efficiently prove to a server that that the client holds a file, rather than just some short information about it. We formalize the concept of proof-of-ownership, under rigorous security definitions, and rigorous efficiency requirements of Petabyte scale storage systems. We then present solutions based on Merkle trees and specific encodings, and analyze their security. We implemented one variant of the scheme. Our performance measurements indicate that the scheme incurs only a small overhead compared to naive client-side deduplication.

3. METHODOLOGY

Integration systems at Web scale need to automatically match records from different sources that refer to the same real-world entity find the true matching records among them and turn this set of records into a standard record for the consumption of users or other applications. There is a large body of work on the record matching problem and the truth discovery problem. The record matching problem is also referred to as duplicate record detection, record linkage, object identification, entity resolution, or deduplication and the truth discovery problem is also called as truth finding or fact finding - a key problem in data fusion.

Disadvantages:

- Truth Discovery Problem.
- Record Matching Problem.

We propose three levels of granularities for record normalization along with methods to construct normalized records according to them.

- We propose a comprehensive framework for systematic construction of normalized records. Our framework is flexible and allows new strategies to be added with ease. To our knowledge, this is the first piece of work to propose such a detailed framework.
- We propose and compare a range of normalization strategies, from frequency, length, centroid and feature-based to more complex ones that utilize result merging models from information retrieval, such as (weighted) Borda.
- We introduce a number of heuristic rules to mine desirable value components from a field. We use them to construct the normalized value for the field.
- We perform empirical studies on publication records. The experimental results show that the proposed weighted-Borda-based approach significantly outperforms the baseline approaches.

Advantages:

- High Accuracy.
- Best Performance.
- We analyzed the record and field level normalization in the typical normalization.
- In the complete normalization, we focused on field values and proposed algorithms for acronym expansion and value component mining to produce much improved normalized field values.

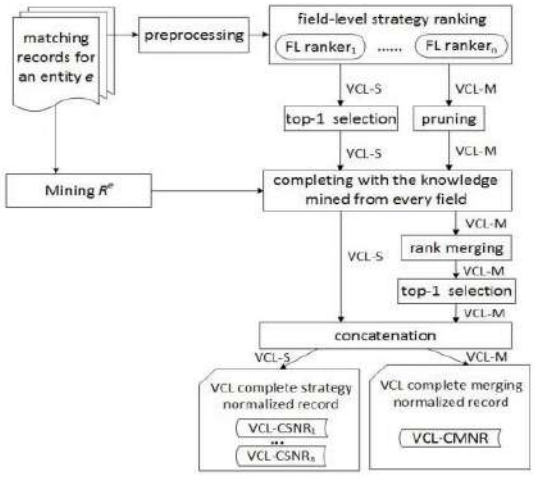

Fig.2: System architecture

Complete Normalization

Complete normalization seeks to produce the normalized record with the property that the value of each of its fieldsis both complete (not missing component) and self-explanatory. For example, there are several different representations of an author's name, such as full name versus first name initial and last name. One would consider

the former to be a better, less ambiguous representation of an author's name than the latter. Likewise, a fully spelled out conference name or journal name is better than its abbreviated counterpart. A record in this form of normalization is unique modulo certain set of transformations, such as permutation (e.g., "the 32nd international conference on Very large data bases, in proceedings of") or replacement with equally unambiguous (e.g., "in proceedings of the thirty second international conference on Very large data bases") of value components. This form of normalization is difficult to achieve in practice. Instead, we strive to produce a version of the normalized record as complete and selfexplanatory as possible given the data at hand. Only the value-component-level strategy can achieve this form of normalization. The reason is that normalization at the record-level and field-level are inherently confined to work with monolithic field values (not value components) from the matching records, which are often incomplete

4. IMPLEMENTATION

MODULES:

- System Model
- Data Deduplication
- File level Deduplication Systems
- Block level Deduplication systems

MODULES DESCRIPTION:

System Model

- In this first module, we develop two entities: User and Secure-Cloud Service Provide.
- User: The user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth. Furthermore, the fault tolerance is required by users in the system to provide higher reliability.
- S-CSP: The S-CSP is an entity that provides the outsourcing data storage service for the users. In the deduplication system, when users own and store the same content, the S-CSP will only store a single copy of these files and retain only unique data. A deduplication technique, on the other hand, can reduce the storage cost at the server side and save the upload bandwidth at the user side. For fault tolerance and confidentiality of data storage, we consider a quorum of S-CSPs, each being an independent entity. The user data is distributed across multiple S-CSPs.

Data Deduplication:

- Data Deduplication involves finding and removing of duplicate datas without considering its fidelity.
- Here the goal is to store more datas with less bandwidth.
- Files are uploaded to the CSP and only the Dataowners can view and download it.
- The Security requirements is also achieved by Secret Sharing Scheme.
- Secret Sharing Scheme uses two algorithms, share and recover.
- Datas are uploaded both file and block level and the finding duplication is also in the same process.
- This is made possible by finding duplicate chunks and maintaining a single copy of chunks.

File Level Deduplication Systems:

- To support efficient duplicate check, tags for each file will be computed and are sent to S-CSPs.
- To upload a file F , the user interacts with S-CSPs to perform the deduplication.
- More precisely, the user firstly computes and sends the file tag ϕF = TagGen(F) to S-CSPs for the file duplicate check.
- If a duplicate is found the user computesand sendsit to a server via a secure channel.
- Otherwise if no duplicate is found the process continues,i.e secret sharing scheme runs and the user will upload a file to CSP.

- To download a file the user will use the secret shares and download it from the SCSP's .
- This approach provides fault tolerance and allows the user to remain accessible even if *any* limited subsets of storage servers fail.

Block Level Deduplication Systems:

- In this module we will show to achieve fine grained block-level distributeddeduplication systems.
- In a block-level deduplication system, the user also needs to firstly perform the file-level deduplication before uploading his file.
- If no duplicate is found, the user divides this file into blocks and performs block-level deduplication.
- The System setup is similar to the file level deduplication except the parameter changes.
- To download a block the user gets the secret shares and download the blocks from CSP.

5. EXPERIMENTAL RESULTS

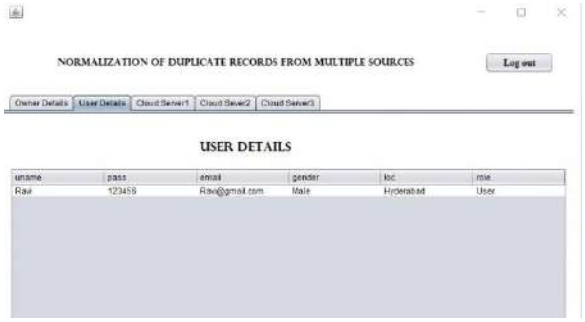

Fig.3: Home screen

Fig.4: registration screen

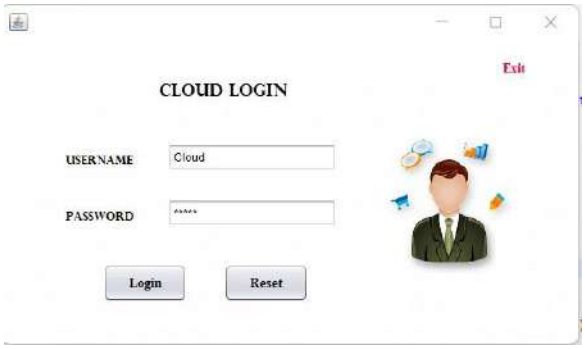

Fig.5: Cloud login

Fig.6: Owner details

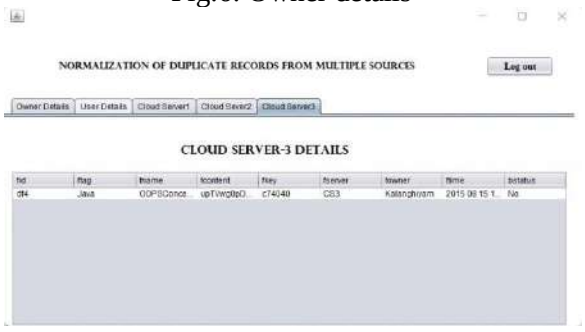

Fig.7: Cloud server-3 details

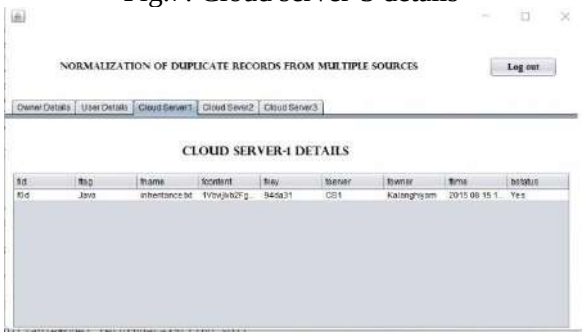

Fig.8: Cloud server-1 details

6. CONCLUSION

In this paper, we studied the problem of record normalization over a set of matching records that refer to the same real-world entity. We presented three levels of normalization granularities (record-level, field-level and value component level) and two forms of normalization (typical normalization and complete normalization). For each form of normalization, we proposed a computational framework that includes both single-strategy and multi-

strategy approaches. We proposed four single-strategy approaches: frequency, length, centroid, and feature-based to select the normalized record or the normalized field value. For multistrategy approach, we used result merging models inspired from metasearching to combine the results from a number of single strategies. We analyzed the record and field level normalization in the typical normalization. In the complete normalization, we focused on field values and proposed algorithms for acronym expansion and value component mining to produce much improved normalized field values. We implemented a prototype and tested it on a real-world dataset. The experimental results demonstrate the feasibility and effectiveness of our approach. Our method outperforms the state-of-the-art by a significant margin. In the future, we plan to extend our research as follows. First, conduct additional experiments using more diverse and larger datasets. The lack of appropriate datasets currently has made this difficult. Second, investigate how to add an effective human-in-the-loop component into the current solution as automated solutions alone will not be able to achieve perfect accuracy. Third, develop solutions that handle numeric or more complex values.

REFERENCES

[1] K. C.-C. Chang and J. Cho, "Accessing the web: From search to integration," in SIGMOD, 2006, pp. 804–805.
[2] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang, "Webtables: Exploring the power of tables on the web," PVLDB, vol. 1, no. 1, pp. 538–549, 2008.
[3] W. Meng and C. Yu, Advanced Metasearch Engine Technology. Morgan & Claypool Publishers, 2010.
[4] A. Gruenheid, X. L. Dong, and D. Srivastava, "Incremental record linkage," PVLDB, vol. 7, no. 9, pp. 697–708, May 2014.
[5] E. K. Rezig, E. C. Dragut, M. Ouzzani, and A. K. Elmagarmid, "Query-time record linkage and fusion over web databases," in ICDE, 2015, pp. 42–53.
[6] W. Su, J. Wang, and F. Lochovsky, "Record matching over query results from multiple web databases," TKDE, vol. 22, no. 4, 2010.
[7] H. Kopcke and E. Rahm, "Frameworks for entity matching: A ¨ comparison," DKE, vol. 69, no. 2, pp. 197–210, 2010.
[8] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," ICDE, 2008.
[9] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," TKDE, vol. 19, no. 1, pp. 1–16, 2007.
[10] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," TKDE, vol. 24, no. 9, 2012.
[11] 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.
[12] L. Shu, A. Chen, M. Xiong, and W. Meng, "Efficient spectral neighborhood blocking for entity resolution," in ICDE, 2011.
[13] Y. Jiang, C. Lin, W. Meng, C. Yu, A. M. Cohen, and N. R. Smalheiser, "Rule-based deduplication of article records from bibliographic databases," Database, vol. 2014, 2014.
[14] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: Is the problem solved?" in PVLDB, vol. 6, no. 2, 2012, pp. 97–108.
[15] J. Pasternack and D. Roth, "Making better informed trust decisions with generalized fact-finding," in IJCAI, 2011, pp. 2324–2329.

