Recognizing the Optimal Events by Matching their Pattern

(User Based Retrieval System)

[1]M.Manoranjitham, [2]Arthy Srinivasan
[1]Assistant Professor, [2]UG Student
[1]Department of Information Technology, Apollo Engineering College, Chennai, India

***Abstract*— This paper deals with huge generation of log files in the heterogeneous environment. Organizations uses n number of computers connected in the network in that case time taken to retrieve the event data might be high. And the generated events are the log files that can be classified as application logs, system logs, security logs and the setup logs. In order to reduce the time and efficiently retrieve the log files pattern matching technique is been used to match the unique pattern and identify the log files. A generic pattern based matching framework is proposed, which is compatible with the existing structure based techniques. To improve the matching efficiency, we devise several bounds of matching scores for pruning. Recognizing the NP-hardness of the optimal event matching problem with patterns, we propose efficient heuristic. Finally, extensive experiments demonstrate the effectiveness of our pattern based matching compared with approaches adapted from existing techniques, and the efficiency improved by the bounding, pruning and heuristic methods. In this paper it is stated that the application efficiently handles and retrieves the log files of the systems connected to network.**

***Index Terms*— log analysis, user based retrieval system, log tracker.**

I. INTRODUCTION

This User Based Retrieval System is to store log files (like system logs, application logs, security logs, setup logs) in cloud. User can retrieve data based on the events. Information systems of different divisions or branches in large corporations keep on generating heterogeneous event logs. It is strongly desired to integrate the event data, e.g., for finding steps leading to a same data in multiple sectors, identifying similar complex procedures in different branches, or obtaining a global picture of business processes in various divisions. Without exploring the correspondence among heterogeneous events, query and analysis on the event data may not yield any meaningful result. Unfortunately, directly applying existing schema matching techniques may fail to obtain the right retrieval of heterogeneous events. To solve the matching problem with these names, graph based matching approaches exploit the structural information among attributes. The more similar the dependency relationship is, the more likely the corresponding events can be retrieved with each other. The matching problem is to find a best mapping that can maximize the similarity of dependency relationships between two datasets. So this application can be used to solve existing data retrieval problem and retrieve correct data effectively based on the events.

In this application Admin can login (with user id and password) to the application. Once entered admin can view report that is registered user's details and then upload all the log file into the database. User can register with application (using name, gender, email, password (user choice), address, phone number, designation) to access the log file data. Once user gets registered the user id is automatically generated and displayed to the user. Registered users can login the application using user id and password (which is created on registration). Once user login to the application it will show home page that contains About us, Guide book and Log files. In guide, it consists of guidelines for accessing log files. It will be used for easily accessing the log files.

The Log files can contain System logs, Application logs, Security logs and Setup logs. In that System logs can contain system related backup files and information. User can use event id or level or source or date and time or task category as input in the search box to retrieve their data that contain all the information of that particular input. When click on the event id that will show description of the event. In the same way user can also retrieve other logs (Application, Security, Setup). Application logs can contain application related backup files and their information. Security logs can contain security related backup files and their information and Setup logs can contains setup and installation backup files and their information. These backup files are stored in database and helps to improve efficiency in retrieving data. This application also contains graphs to show the data levels and overall information about log files.

In corporate companies there are many sensitive files which are handled by some authorized user. Those files may present in his/her laptop and there are possibilities that it can easily leak out to the third party which are difficult to monitor in the network. The number of leaked sensitive data records has grown 10 times in the last 4 years, and it reached a record high of 1.1 billion in 2015. A significant portion of the data leak incidents are due to human errors.

For example: A lost or stolen laptop containing unencrypted sensitive files or transmitting sensitive data through mail without using end-to-end encryption such as PGP. A recent Kaspersky Lab survey shows that accidental leak by staff is the

leading cause for internal data leaks in corporate. The data-leak risks posed by accidents exceed the risks posed by vulnerable software. In order to minimize the exposure of sensitive data and documents, an organization needs to prevent clear text sensitive data from appearing in the storage or communication.

II. Ease of Use

A. Screening Tool

A screening tool can be deployed to scan computer file systems, server storage, and inspect outbound network traffic. The tool searches for the occurrences of plaintext sensitive data in the content of files or network traffic. It alerts users and administrators of the identified data exposure vulnerabilities.

For example, an organization's mail server can inspect the content of outbound email messages searching for sensitive data appearing in unencrypted messages. Data leak detection differs from the anti-virus (AV) scanning (e.g., scanning file systems for malware signatures) or the network intrusion detection systems (NIDS) (e.g., scanning traffic payload for malicious patterns). AV and NIDS typically employ automata-based string matching, which match static or regular patterns.

.

III. Approximate Event Matching

One of the early work on approximate event matching is ATOPSS which defines an approximate matching model based on fuzzy membership functions that specify the degree that a value in an event matches a value in a subscription. A-TOPSS does not consider schema approximation. Another work is STOPSS which considers schema and value semantic matching. It proposes the use of agreed-upon on to logs and a system architecture that generates events other than the original ones by replacing concepts with taxonomic or ad-hoc related concepts.

S-TOPSS provides a generic architecture but no concrete discussion or empirical validation has been provided. Generating new events out of the original ones has the disadvantage of overwhelming the system with large amount of events. The matching model in S-TOPSS is Boolean and scoring as a result of matching was not considered. FO-Match proposes the use of fuzzy onto logs that all interacting parties agree upon. FO-Match is the closest one to the work presented in this paper but it does not remove explicit semantic coupling from the system and does not free the user from using pre-defined vocabularies. Properties and values are handled in distinguishably and relatedness of terms is limited to a measure of combination of edge weights in a taxonomic and synonymy ontology.

IV. Event Matching Problem

In this section we discussed about the basic definition and problems of event matching.

An event is defined as any action performed by an system. The problem in event matching is that current approaches in complex event processing and stream reasoning focus on temporal relationships between composite events. The log files generated will be stored using the extensions EVT and EVTX.

The major disadvantages of existing system is that it takes more time to retrieve and occupies more memory space.

A. Pattern Matching Algorithm

The simple algorithm for pattern matching is given below.

Algorithm for pattern matching

```
int  X = 0;
int  [ ] next =new int [M] ;
for (int j =1 ; j< M; j++ )
{
if  (P.charAt ( X ) = = P.CharAt ( j ) )
{     //match
Next [ j ] = next [ X ] ;
X= X+ 1 ;
}
else
{     //mismatch
Next [ j ] = X + 1;
X= next [X] ;
}
}
```

V. Approximation Algorithm For NP-hard Problem

. In this section, we discuss a different approach to handling difficult problems of combinational optimization, such as the traveling salesman problem and the knap sack problem. As we pointed out in previous section the

decision versions of these problems and NP-complete. The optimization version of such difficult combinatorial problems fall in the class of **NP-hard problems**—problems that are at least as hard as NP-complete problems. Hence there are no known polynomial time algorithm for these problems, and there are serious theoretical reasons to believe that such algorithms do not exist. What then are our options for handling such problems, many of which are of significant practical importance?

If an instance of the problem in questions is very small, we might be able to solve by an exhaustive search algorithm. Some such problems can be solved by the dynamic programming technique as demonstrated in previous section. But even when this approach works in principle, its practicality is limited by dependence on the instance parameters being relativity small. The discovery of the branch-and-bound technique has proved to be an important breakthrough, because this technique makes it possible to get solutions to many large instances of difficult problems of combinatorial optimization in an acceptable amount of time. However, such good performance cannot usually be guaranteed.

There is a radically different way of dealing with difficult optimization problems: solve them approximately by a fast algorithm. This approach is particularly appealing for applications where a good but not necessarily optional solution will suffice. Beside in real life application, we often have to operate with inaccurate data to begin with. Under such circumstances, going for an approximate solution can be a particularly sensible choice.

Although approximation algorithm run a gamut in level of sophistication most of them are based on some problem specific heuristic. A **heuristic** is a common sense rule drawn from experience rather than from a mathematically proved assertion. For example, going to the nearest un visited city in the traveling salesman problem is a good illustration of this notation. We discuss an algorithm based on this heuristic later in this section.

Of course, if we use an algorithm whose output is just an approximation of the actual optimal solution, we would like to know how accurate this approximation is. We can quantify the accuracy of an approximate solution Sa to a problem minimizing some function f by the size of the relative error of this approximation.

$$re(Sa) = f(Sa)- f(S^*)/ f(S^*)$$

Where S* is an exact solution to the problem. Alternatively, since re(Sa)= f(Sa)/f(S*)-1, we can simply use the **accuracy ratio**

$$r(Sa)= f(Sa)/f(S^*)$$

A polynomial time approximation algorithm is said to be a approximation algorithm, where c>=1, if the accuracy ratio of the approximation it produces does not exceed c for any instance of the problem in the question.

$$r(Sa)<=c.$$

the best value of c for which inequality holds for the instances of the problem is called the **performance ratio** of the algorithm and denoted R_A.

VI. RELATED WORK

A. Event Ranking

Ranking events are according to range predicates, preference, diversity and freshness, probability of occurrence, fuzzy membership of attribute values or focused on efficient ranking in sliding windows rather than the ranking functionality. All of these works do not use semantic relatedness of events as a factor for ranking. FO-Match considers scoring based on semantic matching and evaluation was conducted using thresholds, however a precision recall trade off was not investigated.

B. Schema Matching with Opaque Column and Data Values

The schema matching problem at the most basic level refers to the problem of mapping schema elements (for example, columns in a relational database schema) in one information repository to corresponding elements in a second repository. While schema matching has always been a problematic and interesting aspect of information integration, the problem is exacerbated as the number of information sources to be integrated, and hence the number of integration problems that must be solved, grows. Such schema matching problems arise both in "classical" scenarios such as company mergers, and in "new" scenarios such as the integration of diverse sets of query able information sources over the web. Purely manual solutions to the schema matching problem are too labour intensive to be scalable; as a result, there has been a great deal of research into automated techniques that can speed this process by either automatically discovering good mappings, or by proposing likely matches that are then verified by some human expert. In this paper we present such an automated technique that is designed to be of assistance in the particularly difficult cases in which the column names and data values are "opaque," and/or cases in which the column names are opaque and the data values in multiple columns are drawn from the same domain. Our approach works by computing the "mutual information" between pairs of columns within each schema, and then using this statistical characterization of pairs of columns in one schema to propose matching pairs of columns in the other schema

C. Detection and Resolving Unsound Workflow Views for Efficient Provenance Analysis

Technological advances have enabled the capture of massive amounts of data in different domains, taking us a step closer to solving complex problems such as global climate change and uncovering the secrets hidden in genes. Workflow management systems are therefore increasingly used for managing and analysing this data, allowing users to specify complex, multi-step, "in-silicon" experiments or analyses. To ensure reproducibility and verifiability of results, many workflow systems are now providing support for provenance.

The *provenance* of a data item is the sequence of steps used to produce the data, together with the intermediate data and parameters used as input to those steps. In general, it can be thought of as a graph which captures the causal dependencies between entities such as data and processes, and queries of provenance as calculating transitive closures of dependencies. As workflows become large and complex, the size of the provenance graph as well as the cost of answering transitive closure queries

becomes problematic, and a number of techniques have recently been proposed for reducing the size of the provenance graph and complexity of calculating provenance information.

D. Text Joins in an RDBMS Web Data Integration

The integration of information from heterogeneous web sources is of central interest for applications such as catalogue data integration and warehousing of web data (e.g., job advertisements and announcements). Such data is typically textual and can be obtained from disparate web sources in a variety of ways, including web site crawling and direct access to remote databases via web protocols. The integration of such web data exhibits many semantics and performance-related challenges. Consider a price-comparison web site, backed by a database, that combines product information from different vendor web sites and presents the results under a uniform interface to the user. In such a situation, one cannot assume the existence of global identifiers (i.e., unique keys) for products across the autonomous vendor web sites. This raises a fundamental problem: different vendors may use different names to describe the same product. For example, a vendor might list a hard disk as "Western Digital 120Gb 7200 rpm," while another might refer to the same disk as "Western Digital HDD 120Gb" (due to a spelling mistake) or even as "WD 120Gb 7200rpm" (using an abbreviation). A simple equality comparison on product names will not properly identify these descriptions as referring to the same entity. This could result in the same product entity from different vendors being treated as separate products, defeating the purpose of the price-comparison web site. To effectively address the integration problem, one needs to match multiple textual descriptions, accounting for: erroneous information (e.g., typing mistakes) abbreviated, incomplete or missing information differences in information "formatting" due to the lack of standard conventions (e.g., for addresses) or combinations.

E. Discovering Complex Event Patterns

The discovery of complex event pattern has been studied in complex event processing. And we studied the problem of discovering frequent event patterns, i.e., the frequency of a subsequence is higher than a support degree. The discovery algorithm is starting with simple sub patterns and incrementally build larger pattern candidates. We improve the efficiency of the discovery algorithm and has ability to discover more complex patterns. As mentioned, the discovery or design of patterns is not the focus of this study. Instead, we directly utilized the given/discovered patterns. Nevertheless, heuristics are discussed above on choosing discriminative patterns for matching.

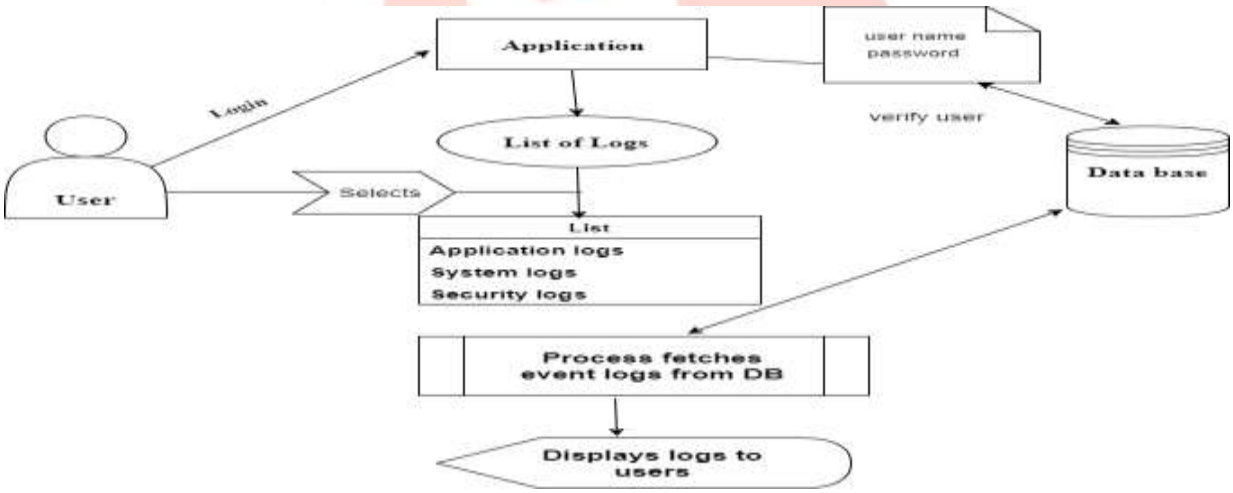

Fig 1: System Architecture Event Log Manager

VII. CONCLUSION

Event-based systems are coupled, via event subscriptions and patterns, to the semantics of the underlying event schema and values. Approximate semantic matching of heterogeneous events has been discussed in this paper in order to address event semantic coupling. Event semantic of types, properties and values has been considered as a dimension of decoupling required to scale event-based systems out to high heterogeneous environments such as the sensor web. A general model has been proposed with a hybrid instantiation based on both thesauri and distributional semantics-based semantic similarity and relatedness measures. Experiments have been conducted on real-world events
extracted from Wikipedia and Freebase. Results show that the proposed hybrid matcher outperforms matchers based on a single semantic similarity or relatedness measure.

REFERENCES

[1] . *Aumueller, D., Do, H.-H., Massmann, S., and Rahm, E.Schema and ontology matching with COMA++. Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, ACM (2005), 906-908.

[2] . *Belkin, N.J. and Croft, W.B. Information filtering andinformation retrieval: two sides of the same coin? Commun.ACM 35, 12 (1992), 29-38.*

[3] . *Drosou, M., Stefanidis, K., and Pitoura, E. Preference-aware publish/subscribe delivery with diversity. Proceedings of the Third ACM International Conference on Distributed Event- Based Systems, (2009), 6:1--6:12.*

[4] . *Wasserkrug, S., Gal, A., Etzion, O., and Turchin, Y. Complex event processing over uncertain data. Proceedings of the second international conference on Distributed eventbased systems, (2008), 253-264.*

[5] . *Liu, H. and Jacobsen, H.-A. A-TOPSS: a publish/subscribe system supporting approximate matching. Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment (2002), 1107-1110.*

[6] . *Erhard Rahm, Philip A. Bernstein: A survey of approaches to automatic schema matching. VLDB Journal 10(4) (2001).*

[7] . *Jayant Madhavan, Philip A. Bernstein, Erhard Rahm:Generic Schema Matching with Cupid. VLDB 2001: 49-58.*

[8] . *Silvana Castano, Valeria De Antonellis, Sabrina De Capitani di Vimercati: Global Viewing of Heterogeneous Data Sources. TKDE 13(2): 277-297 (2001)*

[9] O. Biton, S. C. Boulakia, S. B. Davidson, and C. S. Hara, "Querying and managing provenance through user views in scientific workflows," in Proc. 24th Int. Conf. Data Eng., 2008, pp. 1072–1081. [Online]. Available: http://dx.doi.org/10.1109/ ICDE.2008.4497516

[10] . C. Bettini, X. S. Wang, S. Jajodia, and J. Lin, "Discovering frequent event patterns with multiple granularities in time sequences," IEEE Trans. Knowl. Data Eng., vol. 10, no. 2, pp. 222–237, Mar./ Apr. 1998. [Online]. Available: http://dx.doi.org/10.1109/ 69.683754