Volume 5, Issue 1, January 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Evaluating the Effectiveness of Relational Keyword Search with Experimental Results

R. Karunyam*, P. Rajkumar
Dept. of CSE & Mount Zion College of Engg & Tech
Anna University, Tamil Nadu, India

Abstract— In the present data warehousing environment schemes face lots and lots of issues to fetch out the resources with the help of referential or relational keyword terminologies. So the system requires some kind of advanced data manipulating schemes to extending the keyword search paradigm to relational data has been an active area of research within the database and information retrieval (IR) community. Many approaches have been proposed and implemented to fix the standardizations of relational keyword search procedures, but the result gets down while large amount of data interrupts. This lack of standardization has resulted in contradictory results from different evaluations, and the numerous difference confusion what advantages are offer by different approaches. This System provides a thorough experimental research and evaluation of relational keyword search systems. This system proves that the effectiveness of performance in retrieval tasks and data maintaining procedures. In particular, memory consumption precludes many search techniques from scaling beyond small datasets with tens of thousands of vertices. And also it explores the relationship between execution time and factors varied in previous evaluations; this approach indicates that these factors have relatively little impact on performance. In summary, the outcome confirms previous claims regarding the unacceptable performance of these systems and underscores the need for standardization as exemplified by the IR community when evaluating these retrieval systems.

Keywords— Data warehousing, Information Retrieval, Data interrupts, Relational keyword search system.

I. INTRODUCTION

Everywhere the search text box has transformed the way people interact with information. Nearly half of all Internet users use a search engine daily, performing in excess of 4 billion searches. The success of keyword search stems from what it does not require namely, a specialized query language or knowledge of the underlying structure of the data. Internet users increasingly demand keyword search interfaces for accessing information, and it is natural to extend this paradigm to relational data. This extension has been an active area of research throughout the past decade. Despite a significant number of research papers being published in this area, no research prototypes have transitioned from proof-of-concept implementations into deployed systems.

The lack of technology transfer coupled with discrepancies among existing evaluations indicates a need for a thorough, independent empirical evaluation of proposed search techniques. As part of previous work in this area, we create the first benchmark to evaluate relational keyword search techniques. This benchmark satisfies calls from the research community to standardize the evaluation of these search techniques, and the evaluation of search effectiveness revealed that many search techniques perform comparably despite contrary claims in the literature. During the evaluation of search effectiveness, thus surprised by the difficulty that had in searching the data sets. In particular, straightforward implementations of many search techniques could not scale to databases with hundreds of thousands of tuples, which forced to write "lazy" versions of core algorithms and reduce the memory footprint. Even though the surprised by an excessive runtime of many search techniques.

A. *Structured and Relational Data*

Other researchers have recently reported similar experiences. The shared experience with existing search techniques suggests that the ad hoc evaluations that appear in the literature are inadequate. This sentiment is supported by the survey of existing evaluations and by others who are familiar with the practices established by the IR community for the evaluation of retrieval systems. Keyword search on semi structured data (e.g., XML) and relational data differs considerably from traditional IR. A discrepancy exists between the data's physical storage and a logical view of the information. Relational databases are normalized to eliminate redundancy, and foreign keys identify related information. Search queries frequently cross these relationships (e.g., a subset of search terms is present in one tuple and the remaining terms are found in related tuples), which forces relational keyword search techniques to recover a logical view of the information. The implicit assumption of keyword searches that is, the search terms are related complicates the search process because typically there are many possible relationships between search terms. It is frequently possible to include another occurrence of a search term by adding tuples to an existing result. This realization leads to tension between the compactness (and consequently performance) and coverage of search results. Composing coherent search results from discrete tuples is the primary reason that searching relational data is significantly more complex than searching unstructured text. Unstructured text allows indexing information at the same granularity as the desired results (e.g., by documents or sections within documents). This task is impractical for relational data because an index over logical (or materialized) views is often an order of magnitude larger than the original data. Such an approach will not scale to large databases such as those underlying electronic medical records (EMRs) or social networking sites. In this paper, many relational keyword search techniques approximate solutions to intractable problems. Although worst case performance bounds for many of these algorithms have been established, they perform much better in practice than their algorithmic analysis might suggest. Researchers consequently use empirical evaluation to ascertain the benefits of proposed search techniques. Another motivation for this work is the discrepancies among existing evaluations that litter the literature.

B. Relative Runtime Performance of Relational Keyword Search

First the difference in the relative runtime performance of each search technique is startling. Thus, do not expect the most recent evaluation to downgrade the orders of magnitude performance improvements to performance degradations, which is certainly the case on the DBLP data set. Second, the absolute execution times for the search techniques vary widely across different evaluations. The original evaluation of each approach claims to provide “interactive” response times (on the order of a few seconds). Hence, there remains considerable uncertainty regarding both the relative and absolute performance of existing search techniques. Both of these concerns warrant independent evaluation to establish a performance baseline for realistic retrieval tasks.

C. Contributions of Proposed Approach

The major contributions of this paper are as follows:

- This Project conducts an independent, empirical evaluation of the runtime performance of seven relational keyword search techniques. The evaluation is the most extensive and thorough one to appear to date in the literature.
- The results do not substantiate previous claims regarding the scalability and performance of relational keyword search techniques. Existing search techniques perform poorly on databases exceeding tens of thousands of tuples or require an inordinate amount of memory.
- This paper shows that many parameters varied in existing evaluations are at best loosely correlated with runtime performance. The lack of a meaningful relationship gives merit to previous claims of unpredictable performance for existing search techniques.

Main work is first to combine performance and search effectiveness in the evaluation of such a large number of search techniques. Considering these two issues in conjunction provides better understanding of these two critical tradeoffs among competing approaches.

II. RELATED WORK

The results indicate that many existing search techniques do not provide acceptable performance for realistic retrieval tasks. In particular, memory consumption precludes many search techniques from scaling beyond small datasets with tens of thousands of vertices. Thus, also exploring the relationship between execution time and factors varied in previous evaluations; the analysis indicates that these factors have relatively little impact on performance. In summary, the work confirms previous claims regarding the unacceptable performance of these systems and underscores the need for standardization as exemplified by the IR community when evaluating these retrieval systems.

- Execution Time consumption is less.
- File length and Execution time can be seen.
- Easy to maintain data
- Resolving the Security Issues by means of maintain their identity as well as crypto document privacy
- Less Time Consuming Process
- Low Cost is enough to manipulate the data, because the space required for maintaining the data is less.
- Simple to extract from the server
- Maintenance of Data Owner and user gives greater benefit to analyse them if any case of misuse.

A. Dynamic Programming Algorithm

Dynamic programming algorithms are used for optimization (for example, finding the shortest path between two points, or the fastest way to multiply many matrices). A dynamic programming algorithm will examine all possible ways to solve the problem and will pick the best solution. Therefore, it can roughly think of dynamic programming as an intelligent, brute-force method that enables us to go through all possible solutions to pick the best one. If the scope of the problem is such that going through all possible solutions is possible and fast enough, dynamic programming guarantees finding the optimal solution. The alternatives are many, such as using a greedy algorithm, which picks the best possible choice "at any possible branch in the road". While a greedy algorithm does not guarantee the optimal solution, it is faster. Fortunately, some greedy algorithms (such as minimum spanning trees) are proven to lead to the optimal solution.

B. Pseudo Polynomial-Time Algorithm

A pseudo-polynomial-time algorithm is used to display the exponential behaviour only when confronted with instances containing exponentially large numbers of clusters, which might be rare for the application, are interested in. If so, this type of algorithm might serve the purposes almost as well as a polynomial time algorithm. This algorithm helps to improve the time taken for searching the data from large set of cluster based on the respective keyword and produce the results quickly within a fraction of seconds with the help of Steiner Tree Problem. The Steiner tree problem is superficially similar to the minimum spanning tree problem: which gives set of clusters and searching the required resource from that clusters and produce the results within a minimum of time.

C. Bidirectional Search Algorithm

Bidirectional search algorithm is a searching algorithm that finds a shortest path from an initial highest point to a goal highest point in a directed way. It runs two simultaneous searches: one forward from the initial state and one backward from the goal, stopping when the two meet in the middle. The reason for this approach is that in many cases it is faster: for instance, in a simplified model of search problem complexity in which both searches expand a tree with branching factor b, and the distance from start to goal is d, each of the two searches has complexity O(bd/2) (in Big O notation), and the sum of these two search times is much less than the O(bd) complexity that would result from a single search from the beginning to the goal.

D. Sparse Algorithm

The Sparse algorithm discovers the files by its keyword those are presented into the content of the file and executes it in a fraction of second for the user.

E. Skyline Sweep Algorithm

The Skyline Sweep Algorithm is used to minimize the total number of database probes during a search. Searching keywords in databases is complex task than search in files. Information Retrieval (IR) process search keywords from text files and it is very important that queering keyword to the relational databases. Generally to retrieve data from relational database Structure Query Language (SQL) can be used to find relevant records from the database. There is natural demand for relation database to support effective and efficient IR Style Keyword queries. This algorithm clearly supporting effective and efficient top-k keyword search in relational databases also describe the frame word which takes keywords and K as inputs and generates top-k relevant records. The results of implemented system with Skyline Sweeping Algorithm show that it is one effective and efficient style of keyword search.

F. Breadth-First Algorithm

The BFS begins search at a root node and inspects all the neighboring nodes. Then for each of those neighbor nodes in turn, it inspects their neighbor nodes which were unvisited, and so on. In the approach it results the content wise usage of data or results the searching count based on the searched content.

III. SYSTEM DESIGN

A. System Architecture

In proposed system, empirical performance evaluation of relational keyword search systems. Our results indicate that many existing search techniques do not provide acceptable performance for realistic retrieval tasks. In particular, memory consumption precludes many search techniques from scaling beyond small datasets with tens of thousands of vertices. We also explore the relationship between execution time and factors varied in previous evaluations; our analysis indicates that these factors have relatively little impact on performance. In summary, our work confirms previous claims regarding the unacceptable performance of these systems and underscores the need for standardization as exemplified by the IR community when evaluating these retrieval systems.

- File length and Execution time can be seen.
- Easy to maintain data
- Resolving the Security Issues by means of maintain their identity as well as crypto document privacy
- Less Time Consuming Process
- Low Cost is enough to manipulate the data, because the space required for maintaining the data is less.
- Simple to extract from the server
- Maintenance of Data Owner and user gives greater benefit to analyze them if any case of misuse.

Our results should serve as a challenge to this community because little previous work has acknowledged these challenges. Moving forward, we must address several issues. First, we must design algorithms, data structures, and implementations that recognize that main memory is limited. Search techniques must manage their memory utilization efficiently, swapping data to and from disk as necessary. Such implementations are unlikely to have performance characteristics that are similar to existing approaches but must be used if relational keyword search systems are to scale to large data sets (e.g., hundreds of millions of tuples). Second, evaluations should reuse data sets and query workloads to provide greater consistency of results, for even our results vary widely depending on which data set is considered. Fortunately, our evaluation benchmark is beginning to gain traction in this area as evidenced by others' adoption of it for their evaluations. Third, the practice of researchers implementing search techniques may account for some evaluation discrepancies. Making the original source code (or a binary distribution that accepts a database URL and query as input) available to other researchers would greatly reduce the likelihood that observed differences are implementation artifacts.

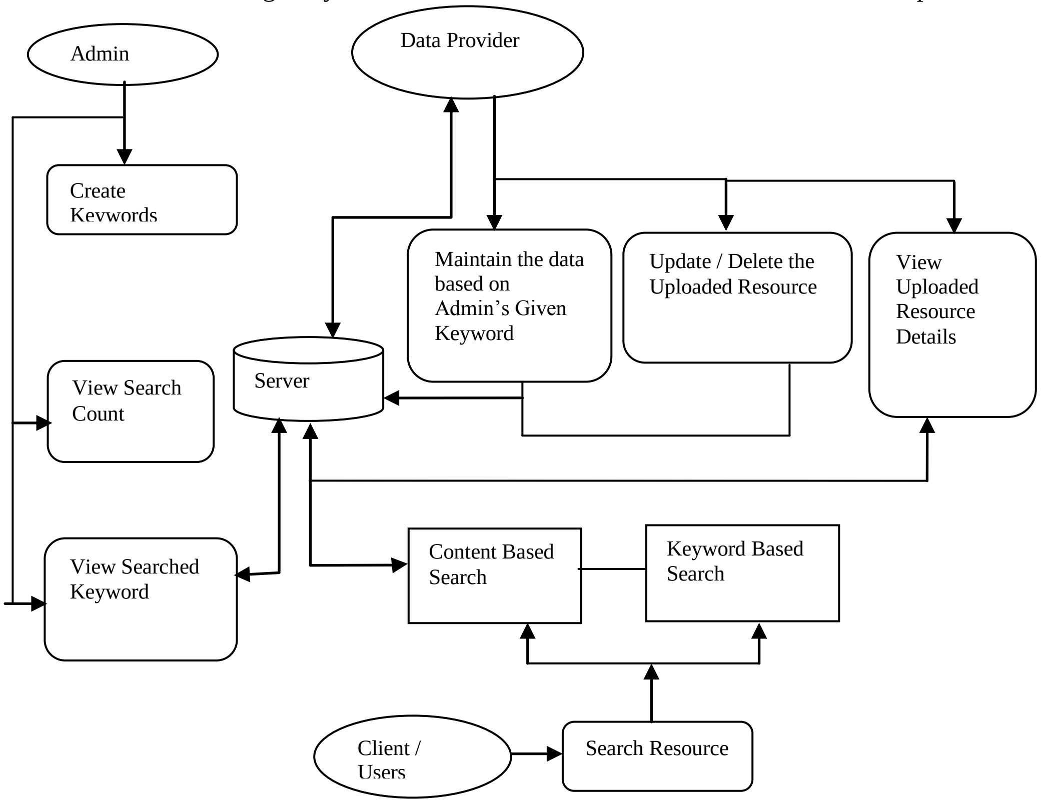


Fig 1: System Architecture

B. Modules

1. Owner Resource Regulation Portal
2. Resource Manipulation
3. Client / Owner Authentication
4. Header Keyword Requisition

1) Owner Resource Regulation Portal: Owner resource regulation portal is the mode of generating resources. The resources are collected and stored into the database after the verification from the admin.

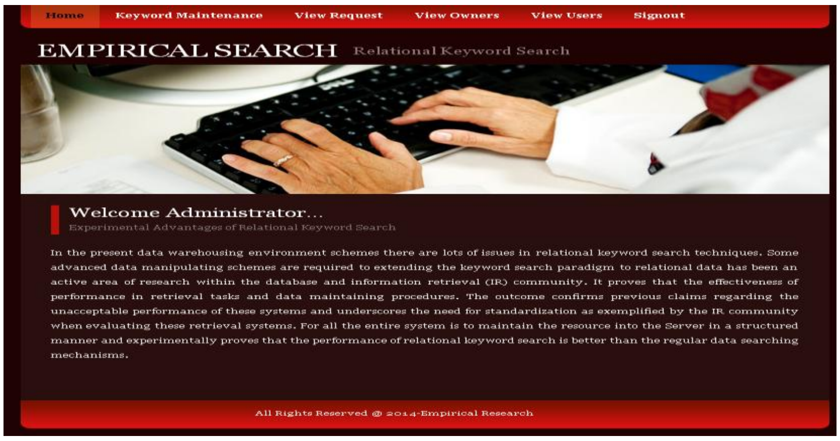
2) Resource Manipulation: Resources are entered into the database through the resource regulation port and they are manipulated based on the particular keyword generated for that resource.

3) Client/Owner Authentication: Authentication is the process of assuring that the particular resource is generated by the customer. It helps in reducing false data into database.

4) Header Keyword Requisition: When you search, browse, create non-catalog request, or perform any action that initiates the creation of a requisition, your user preference is validated. In case of error, the preferences window is opened and you can click on the View Errors link to see the errors. You need to fix your preferences before you can proceed with creating a requisition.

C. Screen Shots

1) Admin



EMPIRICAL SEARCH Relational Keyword Search

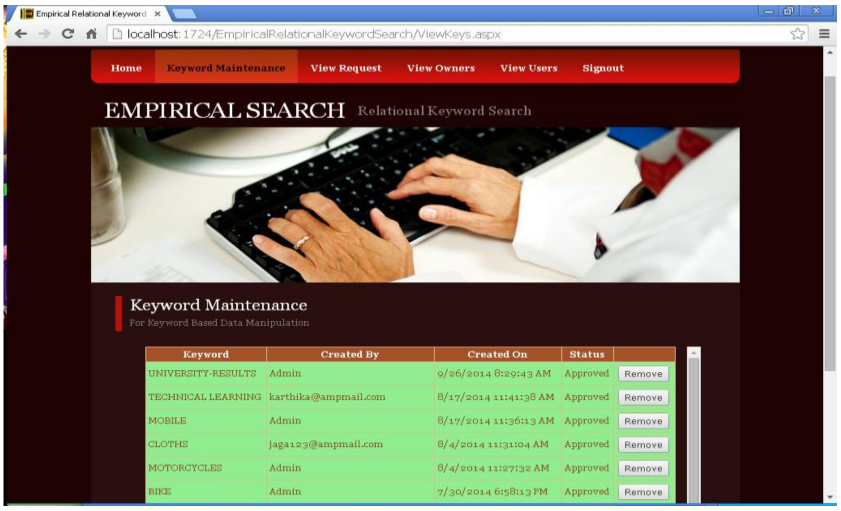
Welcome Administrator...

Experimental Advantages of Relational Keyword Search

In the present data warehousing environment schemes there are lots of issues in relational keyword search techniques. Some advanced data manipulating schemes are required to extending the keyword search paradigm to relational data has been an active area of research within the database and information retrieval (IR) community. It proves that the effectiveness of performance in retrieval tasks and data maintaining procedures. The outcome confirms previous claims regarding the unacceptable performance of these systems and underscores the need for standardization as exemplified by the IR community when evaluating these retrieval systems. For all the entire system is to maintain the resource into the Server in a structured manner and experimentally proves that the performance of relational keyword search is better than the regular data searching mechanisms.

All Rights Reserved @ 2014-Empirical Research

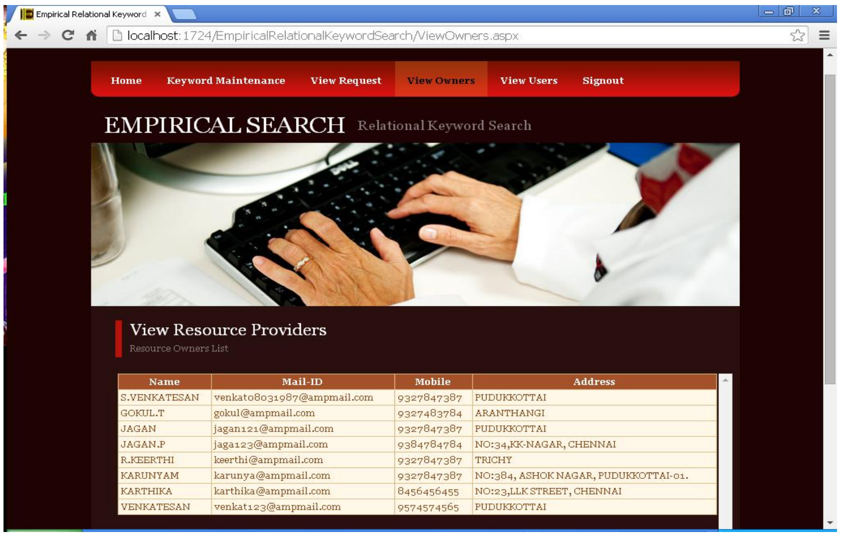
2) Keyword Maintenance


Empirical Relational Keyword
localhost:1724/EmpiricalRelationalKeywordSearch/ViewKeys.aspx
Home
Keyword Maintenance
View Request
View Owners
View Users
Signout
EMPIRICAL SEARCH
Relational Keyword Search

Keyword Maintenance
For Keyword Based Data Manipulation

Keyword	Created By	Created On	Status	
UNIVERSITY-RESULTS	Admin	9/26/2014 8:29:43 AM	Approved	Remove
TECHNICAL LEARNING	karthika@ampmail.com	8/17/2014 11:41:38 AM	Approved	Remove
MOBILE	Admin	8/17/2014 11:36:13 AM	Approved	Remove
CLOTHS	jaga123@ampmail.com	8/4/2014 11:31:04 AM	Approved	Remove
MOTORCYCLES	Admin	8/4/2014 11:27:32 AM	Approved	Remove
BIKE	Admin	7/30/2014 6:58:13 PM	Approved	Remove

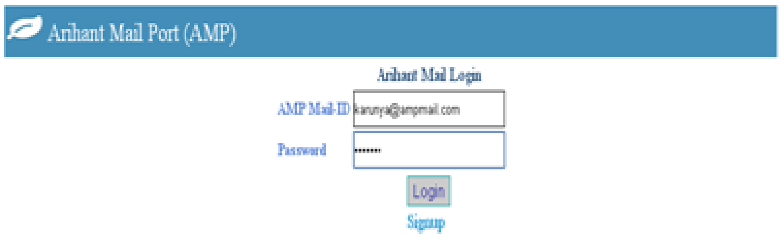
3) Resource Providers


Empirical Relational Keyword
localhost:1724/EmpiricalRelationalKeywordSearch/ViewOwners.aspx
Home
Keyword Maintenance
View Request
View Owners
View Users
Signout
EMPIRICAL SEARCH
Relational Keyword Search

View Resource Providers
Resource Owners List

Name	Mail-ID	Mobile	Address
S.VENKATESAN	venkat08031987@ampmail.com	9327847387	PUDUKKOTTAI
GOKUL.T	gokul@ampmail.com	9327483784	ARANTHANGI
JAGAN	jagan121@ampmail.com	9327847387	PUDUKKOTTAI
JAGAN.P	jaga123@ampmail.com	9384784784	NO:34,KK-NAGAR, CHENNAI
R.KEERTHI	keerthi@ampmail.com	9327847387	TRICHY
KARUNYAM	karunya@ampmail.com	9327847387	NO:384, ASHOK NAGAR, PUDUKKOTTAI-01.
KARTHIKA	karthika@ampmail.com	8456456455	NO:23,LLK STREET, CHENNAI
VENKATESAN	venkat123@ampmail.com	9574574565	PUDUKKOTTAI

4) Owner Regulation Portal



AMP Mail-ID: karunya@ampmail.com
Password: •••••••

IV. CONCLUSIONS

The current system produces the different search results in the literature; it investigates the overall, end-to-end performance of relational keyword search techniques, so it is favour for the user to a realistic query workload instead of a larger workload with queries that are unlikely to be representative (e.g., queries created by randomly selecting terms from the data set). The proposed system does not replicate well on existing relational keyword search techniques. Runtime performance is unacceptable for most search techniques. Memory consumption is also excessive for many search techniques. This system outcome queries the scalability and improvements claimed by preceding evaluations. These conclusions are reliable with preceding assessment that demonstrates the poor runtime performance of existing search techniques as an introduction to a newly planned approach.

Acknowledgment

Our completion of this paper could not have been accomplished without the support of staff members those who are working with us. We are very much thankful to them. For the reference, we refer many articles and research papers of many authors and institutions those are available in online and offline. We offer our sincere appreciation for the learning opportunities provided by those authors and institutions. At last but not least, our heartfelt thanks to all our family members for caring us and for the huge support.

References

[1] J. Coffman And A.C. Weaver, "A Framework For Evaluating Database Keyword Search Strategies," Proc. 19th Acm Int'l Conf. Information And Knowledge Management (Cikm '10), Pp. 729-738, Oct. 2010.

[2] W. Webber, "Evaluating The Effectiveness Of Keyword Search," IEEE Data Eng. Bull., Vol. 33, No. 1, Pp. 54-59, Mar. 2010.

[3] Q. Su and J. Widom, "Indexing Relational Database Content Offline for Efficient Keyword-Based Search," Proc. Ninth Int'l Database Eng. And Application Symp.(Ideas '05), Pp. 297-306, July 2005

[4] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional Expansion For Keyword Search on Graph Databases," Proc. 31st Int'l Conf. Very Large Data Bases (VLDB '05), pp. 505-516, Aug. 2005.

[5] H. He, H. Wang, J. Yang, and P.S. Yu, "BLINKS: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '07), pp. 305-316, June 2007.

[6] A. Baid, I. Rae, J. Li, A. Doan, and J. Naughton, "Toward Scalable Keyword Search Over Relational Data," Proc. Vldb Endowment, Vol. 3, No. 1, Pp. 140-149, 2010.

[7] Y. Chen, W. Wang, Z. Liu, and X. Lin, "Keyword Search on Structured and Semi-Structured Data," Proc. Acm Sigmod Int'l Conf. Management of Data (Sigmod'09), Pp. 1005-1010, June 2009.

[8] W. May, "Information Extraction and Integration with Florid: The Mondial Case Study," Technical Report 131, University Freiburg, Institute¨ Information, 1999.