Keyword Search on Structured and Semi-Structured Data

Yi Chen[1], Wei Wang[2], Ziyang Liu[1], Xuemin Lin[2]
Arizona State University[1], University of New South Wales and NICTA[2]
yi@asu.edu, weiw@cse.unsw.edu.au, ziyang.liu@asu.edu, lxue@cse.unsw.edu.au

ABSTRACT

Empowering users to access databases using simple keywords can relieve the users from the steep learning curve of mastering a structured query language and understanding complex and possibly fast evolving data schemas. In this tutorial, we give an overview of the state-of-the-art techniques for supporting keyword search on structured and semi-structured data, including query result definition, ranking functions, result generation and top-k query processing, snippet generation, result clustering, query cleaning, performance optimization, and search quality evaluation. Various data models will be discussed, including relational data, XML data, graph-structured data, data streams, and workflows. We also discuss applications that are built upon keyword search, such as keyword based database selection, query generation, and analytical processing. Finally we identify the challenges and opportunities of future research to advance the field.

Categories and Subject Descriptors

H.2.0 [**Database Management**]: General; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Design, Performance, Theory

Keywords

keyword search, databases, XML, top-k

1. INTRODUCTION

Searching for information is an indispensable component of our lives. Web search engines are widely used for searching textual documents, images, and videos. There are also vast collections of structured and semi-structured data both on the Web and in enterprises, such as relational databases, XML, data extracted from text documents, workflows, etc. Traditionally, to access these resources, users have to learn structured query languages, such as SQL and XQuery; they also need to access data schemas of each individual application domain, which are most likely complex, fast-evolving, or even unavailable in Web applications. A natural question to ask is whether we can empower users to effectively access structured data using keyword queries.

Ideally the result of a keyword search over structured data will automatically assemble relevant pieces of data that are in different locations but are inter-connected and collectively relevant to the query. There are several advantages of such an approach. First, it can relieve casual users from the steep learning curve of studying structured query languages and data schemas when accessing structured data. Second, it allows users to easily access heterogeneous databases. For instance, for websites with database back-ends, this approach provides a more flexible search method than the existing solution that uses a fixed set of pre-built template queries. Furthermore, this approach helps to reveal interesting or unexpected relationships among entities. Making database searchable will substantially increase the information volume that a user can access, have potential to provide search results with better quality compared with keyword search on textual documents, and thus increase the database usability and make significant impact to people's lives.

Due to substantial benefits of supporting keyword search on structured data, it becomes an emerging hot area in database research and development. Researchers from different disciplines (e.g., information retrieval and theoretical computer science) are joining the workforce to tackle various challenges in supporting keyword search on structured data. For example, there are *more than forty* papers on this topic in the last three years in SIGMOD/PODS, VLDB, ICDE and EDBT alone. Major database research laboratories, such as Microsoft and IBM, are working in this area [1, 51, 56]. The first workshop on *Keyword Search on Structured Data (KEYS)* [23] is held, collocated with ACM SIGMOD/PODS 2009.

The mission of supporting keyword search on structured data is also well aligned with recent keynotes in major database conferences [12, 21, 48, 55].

This is the first tutorial about keyword search on structured data in major database conferences. The most related (but orthogonal) tutorial is given in VLDB 2005 by Amer-Yahia and Shanmugasundaram, "XML Full-Text Search: Challenges and Opportunities" [2], focusing on the problem of structured query languages for XML data with full-text search functionalities that allow advanced users to precisely spec-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA.
Copyright 2009 ACM 978-1-60558-551-2/09/06 ...$5.00.

ify their needs. These two complementary tutorials together provide overviews of integrating database and information retrieval techniques.

The *objective* of this tutorial is to provide an overview of the state-of-the-art in supporting keyword search on structured data, outline the problem space in this area, introduce representative techniques that address different aspects of the problem, and discuss further challenges and promising directions for future work. The problem spectrum that we will present ranges from query result definition, ranking functions, query result generation and top-k query processing, result snippet generation, result clustering, query cleaning, performance optimization, to search quality evaluation. We will categorize and compare techniques to address the above problems in various data models, including XML data, relational data, graph-structured data, data streams, as well as workflows, and establish the connections between them. We will also discuss applications that are built upon keyword search, such as keyword based database selection, query generation, and analytical processing. We will identify and analyze challenges and opportunities of future research to advance the field. The tutorial will provide the researchers in databases a systematic and well-organized overview of the state-of-the art in supporting keyword search on structured data.

2. TUTORIAL OUTLINE

This three-hour tutorial categorizes existing work and covers the following topics.

2.1 Generating Search Results

Unlike traditional database applications where query results are fully specified by structured queries, the first task in keyword search is to define query results which *automatically* gather *relevant* information that is generally fragmented and scattered across multiple places (e.g., records in different relations in RDBMSs, different databases in Web/distributed databases, and elements/nodes in XML or graph-structured data).

Query Result Definition. When the data is modeled as a tree, *lowest common ancestor* (LCA) is a fundamental form to define the query results. That is, a result is a subtree rooted at the LCA of a set of nodes that collectively match query keywords [3, 5, 11, 25, 33, 37, 47, 57, 58]. A query result on a graph data model is commonly defined as a subtree of the data graph where no node or edge can be removed without losing connectivity or keyword matches. Since finding the smallest result, which is the group Steiner tree, is NP-hard, variations and relaxation of the definition have been proposed in order to attain reasonable efficiency [4, 10, 13, 22, 24, 29]. Furthermore, besides the data that match query keywords, studies have been performed on identifying data that do not match keywords, but are *implicitly* relevant [16, 26, 35, 38, 50].

Ranking Functions. Keyword searches are inherently ambiguous, and not all query results are equally relevant to a user. Various ranking schemes have been proposed to order the query results into a sorted list so that users can focus on the top ones, which are hopefully the most relevant ones. Various ranking schemes are used in existing work, which consider both the properties of data nodes (e.g., TF*IDF, node weight, and page-rank style ranking) and the properties of the whole query result (e.g., number of edges, weights on edges, size normalization, redundancy penalty) [3, 5, 6, 8, 10, 11, 13, 16, 24, 29, 28, 34, 39, 43, 44, 51, 53, 54, 56, 59].

Result Generation and Top-k Query Processing. We will introduce representative algorithms for query result generation and efficient top-k query processing. For keyword search on XML data, encoding and indexing schemes [5, 11, 33, 47, 57] as well as materialized views [36] have been exploited. For keyword search on relational databases, existing approaches are mainly based on candidate network (CN) generation, and differ on processing and optimization techniques to execute the CNs. We will distinguish the algorithms for monotonic ranking functions [17, 34] and non-monotonic ranking functions [39]. For keyword search on graph-structured data, there are exhaustive search based on dynamic programming [8], efficient generation of top-k answers [10], heuristics-based approaches [4, 22, 30, 31, 52], and approaches leveraging precomputing or indexing [6, 9, 13, 29, 41].

We will also compare and discuss the challenges of keyword search processing techniques when the data schema is available [1, 8, 15, 39, 46, 54, 56] versus when it is absent [8, 10, 13, 24].

2.2 Improving Search Quality

To improve search quality and users' search experience, various techniques have been proposed, such as result snippets, result clustering, query cleaning, etc, which have been successfully used in text search. However, they pose new challenges in the context of searching structured data.

Result Snippets. To compensate the inaccuracy of ranking functions, result snippets should be generated [18, 19]. The principle of result snippets is orthogonal to that of ranking functions: let users quickly judge the relevance of query results by providing a brief quotable passage of each query result, so that users can choose and explore relevant ones among many results.

Result Clustering. In face of query ambiguity, instead of displaying a mixture of query results of different semantics, it is more desirable to cluster query results based on their similarity, so that the user can quickly browse all possible interpretations of query semantics and choose the sets of results that are relevant [14, 27, 54].

Query Cleaning. Query cleaning involves semantic linkage and spelling corrections of database-relevant query keywords, followed by segmentation of nearby query keywords so that each segment corresponds to a high quality data term. Compared to query cleaning on textual documents, query cleaning for structured data brings great potentials with new challenges [42].

Evaluation. We will discuss evaluation framework for keyword search engines. One is based on empirical evaluation using benchmark data, such as INEX (INitiative for the Evaluation of XML Retrieval) [20], a benchmark for XML keyword search. The other is formal evaluation, which evaluates an approach based on a set of axioms that capture broad intuitions [37].

2.3 Applications of Keyword Search in Information Integration and Analysis

Supporting keyword search is not only helpful for users to access a single database, but also benefits information integration. As the number of potentially-related data sources continues to grow rapidly, the existing approach of using pre-defined forms and associated query templates can not adequately support diverse data sources and meet diverse user needs. Keyword search provides a light-weight mechanism to access multiple data sources without labor-intensive information integration upfront.

Database Selection. We will discuss techniques that summarize underlying databases by a keyword relationship graph, and select the most relevant data sources with respect to a user keyword search based on derived summaries [29, 43, 53, 59].

Query Generation. We will discuss techniques that allow a casual user to author new query templates and Web forms by posing keyword searches. The keyword searches are matched against source relations and their attributes to create multiple ranked queries linking the keyword matches. The set of queries is attached to a Web query form, which can be reused by anyone with related information needs [49].

Analytical Processing. Online Analytical Processing (OLAP) tools provide elaborate query languages that allow users to group and aggregate data in various ways, and to explore interesting trends and patterns in the data. However, the complexity of issuing such analytic queries is overwhelming. It is highly desirable, yet very challenging, to combine intuitive keyword-based search with the power of OLAP, to allow users to easily analyze complex data [51, 56, 61].

2.4 Open Challenges

We will discuss open problems and possible directions for future research, including:

Diverse Data Models. There are many types of structured data, whose structures can be exploited to provide high-quality search results compared with keyword search on textual documents. Existing work focuses on searching relational databases and data-centric XML data. There are many opportunities for supporting keyword search on other types of structured data, including data extracted from text documents (e.g. parse tree databases), data warehouses [51, 56], spatial and multimedia databases [7, 60], workflows [45], and probabilistic databases. Furthermore, techniques that enable users to seamlessly access vast collections of heterogeneous data sources are in great demand.

Query Forms: Complexity versus Expressive Power. Traditional database query languages, such as SQL and XQuery, are highly expressive but hard to learn; keyword queries are easy to use but lack the expressive power. A natural question to ask is where we could strike a good balance between the two [21]. Existing attempts include explicit or implicit restriction on the occurrences of keywords [34, 40], labeled keyword search [5, 34, 40], analytical keyword queries [51, 56], and a natural language query interface [32]. Studies on both the application needs and theoretical analysis of the trade-offs are imperative.

Search Quality Improvement. Few existing work addresses the quality of search results with respect to user needs. On one hand, there are much we can learn from the Information Retrieval field. In particular, having user involvement in search engine design will be helpful to provide personalized search experience, such as analysis of query log and user click-through streams. On the other hand, keyword search on structured data poses unique challenges on analyzing user preferences.

Evaluation. With the growing popularity of supporting keyword search on structured data, there is an increasing need to provide an evaluation framework to assess and guide the system design. Initiatives on developing empirical benchmarks - INEX [20] and an axiomatic framework [37] have been made for evaluating keyword search strategies on XML data. Contributions from the community are highly demanded for developing comprehensive frameworks for evaluating the retrieval and ranking strategies of keyword search on various structured data models.

3. ABOUT THE PRESENTERS

Yi Chen is an Assistant Professor in the Department of Computer Science and Engineering at Arizona State University, USA. She received Ph.D. degree in Computer Science from the University of Pennsylvania in 2005. She is a recipient of the NSF CAREER award. Her current research interests focus on empowering non-expert users to easily access diverse structured data, in particular, searching and optimization in the context of databases, information integration, workflows, and social network (`http://www.public.asu.edu/~ychen127/`).

Wei Wang is a Senior Lecturer in the School of Computer Science and Engineering at the University of New South Wales, Australia. He received his Ph.D. degree in Computer Science from Hong Kong University of Science and Technology in 2004. His recent research interests are integration of database and information retrieval technologies, similarity search, and spatial-temporal databases (`http://www.cse.unsw.edu.au/~weiw/`).

Ziyang Liu is a Ph.D. candidate and an SFAz (Science Foundation Arizona) Graduate Fellowship recipient in the Department of Computer Science and Engineering at Arizona State University. He joined Arizona State University in August 2006 and received M.S. degree in Computer Science in May 2008. His current research focuses on keyword search on structured and semi-structured data and workflow management (`http://www.public.asu.edu/~zliu41/`).

Xuemin Lin is a Professor of Computer Science and Engineering and the head of database research group at the University of New South Wales. Xuemin got his PhD in Computer Science from the University of Queensland (Australia) in 1992 and his BSc in Applied Math from Fudan University (China) in 1984. His current research interests lie in data streams, graph databases, keyword search, probabilistic queries, spatial and temporal databases, and web information systems. Currently, he is an associate editor of

ACM Transactions on Database Systems (`http://www.cse.unsw.edu.au/~lxue/`).

4. ACKNOWLEDGEMENT

Yi Chen is supported by NSF CAREER award IIS-0845647 and NSF grant IIS-0740129. Wei Wang is supported by ARC Discovery Grants DP0987273 and DP0881779. Xuemin Lin is supported by Google Research Award and ARC Discovery Grants DP0987557, DP0881035 and DP0666428.

5. REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *ICDE*, 2002.

[2] S. Amer-Yahia and J. Shanmugasundaram. XML full-text search: Challenges and opportunities. In *VLDB*, 2005.

[3] Z. Bao, T. W. Ling, B. Chen, and J. Lu. Effective XML Keyword Search with Relevance Oriented Ranking. In *ICDE*, 2009.

[4] G. Bhalotia, C. Nakhe, A. Hulgeri, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In *ICDE*, 2002.

[5] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSEarch: A semantic search engine for XML. In *VLDB*, 2003.

[6] B. B. Dalvi, M. Kshirsagar, and S. Sudarshan. Keyword search on external memory data graphs. *PVLDB*, 1(1), 2008.

[7] I. De Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, 2008.

[8] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in databases. In *ICDE*, 2007.

[9] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. Garcia-Molina. Proximity search in databases. In *VLDB*, 1998.

[10] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *SIGMOD*, 2008.

[11] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: Ranked keyword search over XML documents. In *SIGMOD*, 2003.

[12] A. Y. Halevy, M. J. Franklin, and D. Maier. Principles of dataspace systems. In *PODS*, 2006.

[13] H. He, H. Wang, J. Yang, and P. S. Yu. Blinks: ranked keyword searches on graphs. In *SIGMOD*, 2007.

[14] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastava. Keyword proximity search in XML trees. *IEEE Transactions on Knowledge and Data Engineering*, 18(4), 2006.

[15] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *VLDB*, 2002.

[16] V. Hristidis, Y. Papakonstantinou, and A. Balmin. Keyword proximity search on xml graphs. In *ICDE*, 2003.

[17] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-Style Keyword Search over Relational Databases. In *VLDB*, 2003.

[18] Y. Huang, Z. Liu, and Y. Chen. eXtract: a Snippet Generation System for XML Search. *PVLDB*, 1(2), 2008.

[19] Y. Huang, Z. Liu, and Y. Chen. Query biased snippet generation in XML search. In *SIGMOD*, 2008.

[20] INEX. Initiative for the evaluation of xml retrieval. `http://inex.is.informatik.uni-duisburg.de/`.

[21] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu. Making database systems usable. In *SIGMOD*, 2007.

[22] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, 2005.

[23] KEYS 2009. The first international workshop on keyword search on structured data, 2009.

[24] B. Kimelfeld and Y. Sagiv. Finding and approximating top-k answers in keyword proximity search. In *PODS*, 2006.

[25] L. Kong, R. Gilleron, and A. Lema. Retrieving Meaningful Relaxed Tightest Fragments for XML Keyword Search. In *EDBT*, 2009.

[26] G. Koutrika, A. Simitsis, and Y. E. Ioannidis. Précis: The essence of a query answer. In *ICDE*, 2006.

[27] G. Koutrika, Z. M. Zadeh, and H. Garcia-Molina. DataClouds: Summarizing Keyword Search Results over Structured Data. In *EDBT*, 2009.

[28] G. Li, J. Feng, J. Wang, and L. Zhou. An effective and versatile keyword search engine on heterogenous data sources. *PVLDB*, 1(2), 2008.

[29] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. EASE: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *SIGMOD*, 2008.

[30] G. Li, X. Zhou, J. Feng, and J. Wang. Progressive Top-k Keyword Search in Relational Database. In *ICDE*, 2009.

[31] W.-S. Li, K. S. Candan, Q. Vu, and D. Agrawal. Retrieving and organizing web pages by "information unit". In *WWW*, 2001.

[32] Y. Li, I. Chaudhuri, H. Yang, S. Singh, and H. V. Jagadish. Danalix: a domain-adaptive natural language interface for querying xml. In *SIGMOD*, 2007.

[33] Y. Li, C. Yu, and H. V. Jagadish. Schema-free XQuery. In *VLDB*, 2004.

[34] F. Liu, C. Yu, W. Meng, and A. Chowdhury. Effective keyword search in relational databases. In *SIGMOD*, 2006.

[35] Z. Liu and Y. Chen. Identifying meaningful return information for xml keyword search. In *SIGMOD*, 2007.

[36] Z. Liu and Y. Chen. Answering keyword queries on XML using materialized views. In *ICDE*, 2008.

[37] Z. Liu and Y. Chen. Reasoning and identifying relevant matches for xml keyword search. *PVLDB*, 1(1), 2008.

[38] Z. Liu, J. Walker, and Y. Chen. XSeek: A semantic XML search engine using keywords. In *VLDB*, 2007.

[39] Y. Luo, X. Lin, W. Wang, and X. Zhou. SPARK: Top-k keyword query in relational databases. In *SIGMOD*, 2007.

[40] Y. Luo, W. Wang, and X. Lin. Spark: A keyword

search engine on relational databases. In *ICDE*, 2008.

[41] A. Markowetz, Y. Yang, and D. Papadias. Reachability Indexes for Relational Keyword Search. In *ICDE*, 2009.

[42] K. Q. Pu and X. Yu. Keyword query cleaning. *PVLDB*, 1(1), 2008.

[43] M. Sayyadian, H. LeKhac, A. Doan, and L. Gravano. Efficient keyword search across heterogeneous relational databases. In *ICDE*, 2007.

[44] F. Shao, L. Guo, and C. Botev. Efficient Keyword Search over Virtual XML Views. In *VLDB*, 2007.

[45] Q. Shao, P. Sun, and Y. Chen. WISE: a workflow information search engine. In *ICDE*, 2009.

[46] Q. Su and J. Widom. Indexing relational database content offline for efficient keyword-based search. In *IDEAS*, 2005.

[47] C. Sun, C.-Y. Chan, and A. Goenka. Multiway SLCA-based keyword search in XML data. In *WWW*, 2007.

[48] Databases and IR: Perspectives of a SQL guy. NSF Information and Data Management PI Workshop, 2003.

[49] P. P. Talukdar, M. Jacob, M. S. Mehmood, K. Crammer, Z. G. Ives, F. Pereira, and S. Guha. Learning to create data-integrating queries. *PVLDB*, 1(1), 2008.

[50] Y. Tao and J. X. Yu. Finding Frequent Co-occurring Terms in Relational Keyword Search. In *EDBT*, 2009.

[51] S. Tata and G. M. Lohman. SQAK: doing more with keywords. In *SIGMOD*, 2008.

[52] T. Tran, S. Rudolph, P. Cimiano, and H. Wang. Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF) Data. In *ICDE*, 2009.

[53] Q. H. Vu, B. C. Ooi, D. Papadias, and A. K. H. Tung. A graph method for keyword-based selection of the top-k databases. In *SIGMOD*, 2008.

[54] S. Wang, Z. Peng, J. Zhang, L. Qin, S. Wang, J. X. Yu, and B. Ding. NUITS: A novel user interface for efficient keyword search over databases. In *VLDB*, 2006.

[55] G. Weikum. DB&IR: both sides now. In *SIGMOD*, 2007.

[56] P. Wu, Y. Sismanis, and B. Reinwald. Towards keyword-driven analytical processing. In *SIGMOD*, 2007.

[57] Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. In *SIGMOD*, 2005.

[58] Y. Xu and Y. Papakonstantinou. Efficient LCA based Keyword Search in XML Data. In *EDBT*, 2008.

[59] B. Yu, G. Li, K. R. Sollins, and A. K. H. Tung. Effective keyword-based selection of relational databases. In *SIGMOD*, 2007.

[60] D. Zhang, Y. M. Chee, A. Mondal, A. Tung, and M. Kitsuregawa. Keyword Search in Spatial Databases: Towards Searching by Document. In *ICDE*, 2009.

[61] B. Zhou and J. Pei. Answering Aggregate Keyword Queries on Relational Databases Using Minimal Group-bys. In *EDBT*, 2009.